



# **MobiLink™ Serveradministration**

**Version 16.0**

**Februar 2013**

Version 16.0  
Februar 2013

© 2013 SAP AG oder ein SAP-Konzernunternehmen. Alle Rechte vorbehalten.

Sie können diese Dokumentation (ganz oder teilweise) unter folgenden Bedingungen benutzen, reproduzieren und verteilen: 1) Sie müssen diese und alle anderen Urheberrechtsvermerke auf allen Kopien oder Auszügen der Dokumentation wiedergeben. 2) Sie dürfen die Dokumentation nicht verändern. 3) Sie dürfen nichts tun, aus dem abgeleitet werden könnte, dass Sie oder jemand anderer als SAP Verfasser oder Quelle der Dokumentation ist. Die hier enthaltenen Informationen können jederzeit ohne vorherigen Hinweis geändert werden.

Einige Softwareprodukte, die von der SAP AG oder einem ihrer Vertriebspartner vermarktet werden, enthalten Softwarekomponenten anderer Softwareanbieter. Die nationalen Produktspezifikationen können unterschiedlich sein.

Diese Dokumentationen werden von der SAP AG und ihren Tochtergesellschaften ("SAP Group") lediglich zu Informationszwecken bereitgestellt, ohne dass eine Gewährleistung oder eine Garantie irgendeiner Art gegeben wird. Die SAP Group übernimmt keine Verantwortung im Hinblick auf Fehler oder Auslassungen in den Dokumentationen. Die einzigen Garantien für Produkte und Dienstleistungen der SAP Group sind diejenigen, die in den mit den Produkten und Dienstleistungen eventuell gelieferten ausdrücklichen Garantieerklärungen enthalten sind. Keine der hier enthaltenen Informationen kann als Gewährung einer weitergehenden Garantie betrachtet werden.

SAP und weitere erwähnte SAP-Produkte und -Dienstleistungen sowie die entsprechenden Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und anderen Ländern. Weitere Hinweise finden Sie unter <http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark>.

---

---

# Inhalt

<b>Über diese Dokumentation .....</b>	<b>v</b>
<b>MobiLink-Servertechnologie .....</b>	<b>1</b>
MobiLink - konsolidierte Datenbanken .....	1
MobiLink-Server .....	30
MobiLink-Serveroptionen .....	44
Synchronisationsmethoden .....	107
MobiLink-Performance .....	143
Zentrale Administration von entfernten Datenbanken .....	153
MobiLink-Profiler .....	192
Der Relay Server .....	213
MobiLink - dateibasierte Downloads .....	213
<b>MobiLink-Ereignisse .....</b>	<b>231</b>
Schreiben von Synchronisationsskripten .....	231
Synchronisationsereignisse .....	272
<b>MobiLink-Server-APIs .....</b>	<b>463</b>
Schreiben eines Synchronisationsskripts in Java .....	463
Referenz der MobiLink-Server-API für Java .....	477
Synchronisationsskripten in .NET .....	536
Referenz zur MobiLink-Server-API für .NET .....	552
Direkte Zeilenbehandlung .....	627
<b>MobiLink-Referenz .....</b>	<b>639</b>
MobiLink Replay-C++-Callback-Funktionen .....	639
Systemprozeduren des MobiLink-Servers .....	648
MobiLink-Dienstprogramme .....	697
MobiLink-Datenzuordnungen zwischen entfernten und konsolidierten Datenbanken .....	714

**Hinweise zum Zeichensatz ..... 765**

**ODBC-Treiber für MobiLink ..... 767**

**Deployment von MobiLink-Anwendungen ..... 771**

**Index ..... 787**

---

# Über diese Dokumentation

Diese Dokumentation beschreibt, wie Sie MobiLink-Server, konsolidierte Datenbanken und MobiLink-Anwendungen einrichten und verwalten. Sie beschreibt außerdem den SQL Anywhere-Monitor für MobiLink, ein Webbrowser-basiertes Administrationstool, das Informationen über den Zustand und die Verfügbarkeit von MobiLink-Servern bereitstellt, sowie den Relay Server, der die sichere Kommunikation zwischen mobilen Geräten und MobiLink-Servern über einen Webserver ermöglicht.



---

# MobiLink-Servertechnologie

Dieser Abschnitt stellt die MobiLink-Technologie vor und beschreibt, wie sie bei der Synchronisation von Daten zwischen zwei oder mehr Datenbanken eingesetzt werden kann.

## MobiLink - konsolidierte Datenbanken

Ihre konsolidierte Datenbank enthält Systemobjekte, die für MobiLink erforderlich sind. Normalerweise sind dort auch Ihre Anwendungsdaten enthalten, aber Sie können alle bzw. Teile Ihrer Anwendungsdaten auch an anderen Stellen aufbewahren.

MobiLink unterstützt konsolidierte Datenbanken für Windows und Linux in 32-Bit- und 64-Bit-Umgebungen. Ihre konsolidierte Datenbank kann eins der folgenden ODBC-kompatiblen RDBMSs sein:

- Adaptive Server Enterprise (keine Linux-Unterstützung in 64-Bit-Umgebungen)
- IBM DB2 LUW
- Microsoft SQL Server
- MySQL
- Oracle
- SAP HANA
- SAP Sybase IQ
- SQL Anywhere

Hinweise zur Versionsunterstützung finden Sie unter <http://www.sybase.com/detail?id=1002288>.

Weitere Hinweise zu empfohlenen ODBC-Treibern für MobiLink finden Sie unter <http://www.sybase.com/detail?id=1011880>

Ihre SQL Anywhere-Installation enthält ein MobiLink-Setupkript für jeden RDBMS-Typ. Führen Sie, je nachdem, welchen RDBMS-Typ Sie mit MobiLink verwenden möchten, das entsprechende Setupkript aus. Das Setupkript fügt Tabellen hinzu und speichert Prozeduren, die für MobiLink erforderlich sind.

Weitere Hinweise zum Einrichten der verschiedenen RDBMS-Typen als konsolidierte Datenbank finden Sie unter „[Einrichten einer konsolidierten Datenbank](#)“ auf Seite 3.

Weitere Hinweise zur Erstellung von Synchronisationsskripten für bestimmte konsolidierte Datenbanken finden Sie unter „[RDBMS-abhängige Synchronisationsskripten](#)“ auf Seite 5.

### Andere Datenquellen synchronisieren

Ihre MobiLink-Umgebung muss über eine Datenbank verfügen, die als konsolidierte Datenbank eingerichtet wurde. Sie können jedoch auch andere Datenquellen als die konsolidierte Datenbank synchronisieren. Die anderen Datenquellen können praktisch alles sein: eine Textdatei, ein Webdienst, eine nicht-relationale Datenbank, eine Tabellenkalkulation usw. Sie können Folgendes durchführen:

- Nur eine konsolidierte Datenbank synchronisieren.
- Nur eine andere Datenquelle synchronisieren.

- Eine hybride Anwendung erstellen, in der Sie sowohl eine konsolidierte Datenbank als auch eine andere Datenquelle synchronisieren.

Siehe „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

### **Einschränkungen für die Änderung der konsolidierten Datenbank**

Einige Benutzer können nur eingeschränkt das Schema ihrer konsolidierten Datenbank ändern. Für diese Fälle stellt MobiLink Lösungen bereit, bei denen Änderungen an der konsolidierten Datenbank, soweit möglich, auf ein Minimum beschränkt werden. MobiLink stellt z.B. eine Reihe von Lösungen für die Verwaltung von eindeutigen Primärschlüsseln im gesamten Synchronisationssystem bereit, von denen einige nur geringfügige Auswirkungen auf das Schema der konsolidierten Datenbank haben.

Außerdem können Sie nahezu alle Auswirkungen auf die konsolidierte Datenbank vermeiden, indem Sie Ihre MobiLink-Systemobjekte in einer separaten Datenbank speichern. Siehe „[MobiLink-Systemdatenbank](#)“ auf Seite 4.

## **Beziehung zwischen entfernten und konsolidierten Tabellen**

Das Synchronisationsdesign legt die Zuordnungen zwischen Tabellen und Spalten in den entfernten Datenbanken zu den Tabellen und Spalten in der konsolidierten Datenbank fest. Normalerweise entsprechen Tabellen und Spalten in entfernten Datenbanken genau den Tabellen und Spalten in der konsolidierten Datenbank bzw. sind Teilmengen davon.

### **Willkürliche Beziehungen zulässig**

Die Tabellen in einer entfernten Datenbank brauchen nicht mit den Tabellen in der konsolidierten Datenbank übereinzustimmen. Synchronisierte Daten in einer entfernten Anwendungstabelle können auf Spalten in verschiedenen Tabellen verteilt werden. Diese Beziehungen werden mithilfe der Synchronisationsskripten angegeben.

### **Einfache direkte Beziehungen**

Der einfachste und gängigste Entwurf verwendet eine Tabellenstruktur in der entfernten Datenbank, die eine Teilmenge der Tabellenstruktur in der konsolidierten Datenbank ist. Auf diese Weise gibt es für jede Tabelle der entfernten Datenbank eine genaue Entsprechung in der konsolidierten Datenbank. Entsprechende Tabellen haben dieselbe Struktur und dieselben Fremdschlüsselbeziehungen wie in der konsolidierten Datenbank.

Die konsolidierte Datenbank enthält häufig Spalten und Tabellen, die nicht konsolidiert sind. Einige dieser Spalten oder Tabellen können für die Synchronisation benutzt werden. So kann z.B. eine TIMESTAMP-Spalte neue oder aktualisierte Zeilen in der konsolidierten Datenbank identifizieren oder eine Schattentabelle zum Verfolgen von Löschvorgängen benutzt werden. Nicht synchronisierte Spalten oder Tabellen in der konsolidierten Datenbank können Informationen enthalten, die an den entfernten Standorten nicht erforderlich sind.

Entfernte Datenbanken enthalten häufig auch Tabellen oder Spalten, die nicht synchronisiert werden.

**Siehe auch**

- „MobiLink-Datenzuordnungen zwischen entfernten und konsolidierten Datenbanken“ auf Seite 714

## Einrichten einer konsolidierten Datenbank

### Setupskripten

Um eine Datenbank so einzurichten, dass sie als konsolidierte MobiLink-Datenbank verwendet werden kann, müssen Sie ein Setupskript ausführen. Ihre SQL Anywhere-Installation enthält ein Setupskript für jedes unterstützte RDBMS. Diese Skripten befinden sich in %SQLANY16%\MobiLink\setup. Weitere Hinweise zur MobiLink-Systemeinrichtung finden Sie unter „MobiLink-Systemkonfiguration“ [[MobiLink - Erste Orientierung](#)].

Das MobiLink-Setupskript fügt Ihrer Datenbank MobiLink-Systemtabellen, gespeicherte Prozeduren, Trigger und Ansichten hinzu. Diese Tabellen und Prozeduren sind für die MobiLink-Synchronisation erforderlich.

Weitere Hinweise zu den installierten gespeicherten Prozeduren finden Sie unter „Systemprozeduren des MobiLink-Servers“ auf Seite 648.

**Hinweis**

Der gleiche Datenbankbenutzer, der die Setupskripten ausführt, wird als Benutzer erwartet, der die MobiLink-Systemtabellen während der Synchronisation aktualisiert. Dieser Benutzer muss verwendet werden, um den MobiLink-Server zu starten und MobiLink zu konfigurieren. Siehe [Erforderliche Privilegien](#) auf Seite 31.

Weitere Hinweise dazu, wie Sie diese Setupskripten ausführen, finden Sie im Abschnitt zu Ihrem RDBMS-System.

- „Konsolidierte Adaptive Server Enterprise-Datenbank“
- „Konsolidierte IBM DB2 LUW-Datenbank“
- „Konsolidierte Microsoft SQL Server-Datenbank“
- „Konsolidierte MySQL-Datenbank“
- „Konsolidierte Oracle-Datenbank“
- „Konsolidierte SAP HANA-Datenbank“ auf Seite 25
- „Konsolidierte SAP Sybase IQ-Datenbank“
- „Konsolidierte SQL Anywhere-Datenbank“

### ODBC-Verbindung

Der MobiLink-Server benötigt eine ODBC-Verbindung zur konsolidierten Datenbank. Sie müssen den geeigneten ODBC-Treiber für Ihr RDBMS konfigurieren und eine ODBC-Datenquelle für die Datenbank auf dem Computer erstellen, auf dem der MobiLink-Server ausgeführt wird.

Weitere Hinweise zu MobiLink-ODBC-Treibern finden Sie unter „ODBC-Treiber für MobiLink“ auf Seite 767.

Aktualisierte Informationen und vollständige Funktionsspezifikationen der ODBC-Treiber, die Sie mit MobiLink verwenden können, finden Sie unter <http://www.sybase.com/detail?id=1011880>.

## MobiLink-Systemdatenbank

In gewissen Fällen kann es sinnvoll sein, die konsolidierte Datenbank in zwei Datenbanken aufzuteilen: eine Datenbank für Daten und eine weitere Datenbank für die MobiLink-Systeminformationen. Hierbei müssen Sie der konsolidierten Datenbank keine MobiLink-Systemobjekte hinzufügen. Alle MobiLink-Systemobjekte können in einer separaten Datenbank gespeichert werden, die als MobiLink-Systemdatenbank bezeichnet wird.

### Hinweis

Dieses Setup erfordert einen Koordinator für verteilte Transaktionen. Derzeit wird von MobiLink nur einer unterstützt, nämlich Microsoft Distributed Transaction Coordinator (MS DTC), der nur unter Windows läuft.

Als MobiLink-Systemdatenbank kann jede Datenbank verwendet werden, die als konsolidierte Datenbank unterstützt wird. Es muss sich dabei nicht um dasselbe RDBMS wie bei Ihrer konsolidierten Datenbank handeln.

Die Einrichtung einer MobiLink-Systemdatenbank ist einfach. Wenden Sie einfach MobiLink-Setupskripten auf eine Datenbank an, bei der es sich nicht um die konsolidierte Datenbank handelt. Wenn Sie den MobiLink-Server starten, verbinden Sie sich mit beiden Datenbanken unter Verwendung von -c für die konsolidierte Datenbank und -cs für die Systemdatenbank.

### Hinweise

- Wenn Sie eine separate Systemdatenbank verwenden, können Sie den MobiLink-Server nur unter Windows ausführen unter Verwendung von Microsoft Distributed Transaction Coordinator.
- Sie können MobiLink-Systemdatenbanken nicht mit dem **MobiLink-Assistenten für das Deployment eines Synchronisationsmodells** verwenden.
- Das Speichern von MobiLink-Systemobjekten in einer separaten Datenbank führt zu einer schlechteren Performance.

## Systemtabellen des MobiLink-Servers

In den MobiLink-Systemtabellen werden Angaben zu MobiLink-Benutzern, Subskriptionen, Tabellen, Skripten, Skriptversionen und andere Angaben gespeichert. Sie sind für die MobiLink-Synchronisation erforderlich. Sie können zwar die MobiLink Systemtabellen ändern, aber normalerweise ist das nicht erforderlich.

MobiLink-Systemtabellen werden erstellt, wenn Sie das MobiLink-Setupskript für Ihre konsolidierte Datenbank ausführen. Sie müssen in Ihrer konsolidierten Datenbank gespeichert sein und erhalten immer das Präfix **ml\_**. Der Datenbankbenutzer, der das Skript ausführt, ist der Eigentümer der vom Skript erstellten MobiLink-Systemtabellen.

Siehe „[Einrichten einer konsolidierten Datenbank](#)“ auf Seite 3.

## Hinweise

- In diesem Abschnitt werden die Datentypen für die MobiLink-Systemtabellen in konsolidierten SQL Anywhere-Datenbanken beschrieben. In verschiedenen RDBMS-Systemen unterscheiden sich die Datentypen ein wenig.

## RDBMS-abhängige Synchronisationsskripten

Mithilfe von Synchronisationsskripten definiert MobiLink die zur Datensynchronisation verwendeten Regeln. Die Synchronisationsskripten definieren Folgendes:

- Wie Daten, die aus der entfernten Datenbank heraufgeladen wurden, in die konsolidierte Datenbank übernommen werden.
- Welche Daten aus der konsolidierten Datenbank in die entfernte Datenbank heruntergeladen werden sollen.

Siehe [„Schreiben von Synchronisationsskripten“](#) auf Seite 231.

Eine vollständige Liste der Ereignisse, für die Sie Skripten schreiben können, finden Sie unter [„Synchronisationsereignisse“](#) auf Seite 272.

Konkrete Hinweise zu den verschiedenen Typen von konsolidierten Datenbanken finden Sie unter:

- [„Konsolidierte SQL Anywhere-Datenbank“](#)
- [„Konsolidierte Adaptive Server Enterprise-Datenbank“](#)
- [„Konsolidierte IBM DB2 LUW-Datenbank“](#)
- [„Konsolidierte Microsoft SQL Server-Datenbank“](#)
- [„Konsolidierte MySQL-Datenbank“](#)
- [„Konsolidierte Oracle-Datenbank“](#)
- [„Konsolidierte SAP HANA-Datenbank“](#) auf Seite 25
- [„Konsolidierte SAP Sybase IQ-Datenbank“](#)

## .NET- und Java-Synchronisationsskripten

Sie können Ihre Synchronisationslogik in die Version der von Ihrer Datenbank verwendeten SQL-Sprache schreiben. Außerdem können Sie besser portierbare und leistungsstärkere Skripten unter Verwendung von Java oder .NET schreiben. Java und .NET bieten - bei vollständiger SQL-Kompatibilität - mehr Flexibilität als jedes RDBMS per SQL. Wenn Sie die Java- oder .NET-Synchronisationslogik verwenden, können Sie Variable für eine ganze Sitzung beibehalten, Authentifizierung in externe Server integrieren usw.

Weitere Hinweise zur Java-Synchronisationslogik finden Sie unter [„JavaSynchronisationslogik“](#) auf Seite 465.

Weitere Hinweise zur .NET-Synchronisationslogik finden Sie unter [„Synchronisationsskripten in .NET“](#) auf Seite 536.

## Prozeduren von Skripten aufrufen

Für einige Datenbanken, wie etwa Microsoft SQL Server, müssen Prozeduraufrufe mit Parametern unter Verwendung der ODBC-Syntax geschrieben werden.

```
{ CALL procedure_name( {ml param1}, {ml param2}, ... ) }
```

Sie können Werte zurückgeben, indem Sie die Parameter in der Prozedurdefinition als OUT oder INOUT angeben.

## CHAR-Spalten

In vielen anderen RDBMS-Systemen haben CHAR-Datentypen eine feste Länge und werden bis zur vollen Länge der Zeichenfolge mit Leerzeichen aufgefüllt. In entfernten MobiLink-Datenbanken (SQL Anywhere oder UltraLite) entspricht CHAR dem Datentyp VARCHAR: Werte werden nicht bis zu einer bestimmten Länge mit Leerzeichen aufgefüllt. Wenn Sie SQL Anywhere nicht als konsolidierte Datenbank verwenden, sollten Sie unbedingt VARCHAR anstelle von CHAR in der konsolidierten Datenbank verwenden. Wenn Sie CHAR verwenden müssen, können Sie mit der mlsrv16-Befehlszeilenoption -b während der Synchronisation nachgestellte Leerzeichen aus Zeichenfolgen entfernen. Diese Option ist wichtig für Zeichenfolgenvergleiche, mit denen Konflikte erkannt werden.

Siehe „[mlsrv16-Option -b](#)“ auf Seite 51.

## Konvertierung von Daten

Weitere Hinweise zur Konvertierung von Daten für den Fall, dass ein MobiLink-Server mit einer konsolidierten Datenbank kommuniziert, die keine SQL Anywhere-Datenbank ist, finden Sie unter „[MobiLink-Datenzuordnungen zwischen entfernten und konsolidierten Datenbanken](#)“ auf Seite 714.

# Synchronisation von räumlichen Daten

Der MobiLink-Server unterstützt die Synchronisation von Tabellen, die Spalten mit räumlichen Datentypen enthalten. Die folgenden Typen von konsolidierten Datenbanken werden für die Synchronisation von räumlichen Daten unterstützt:

- SQL Anywhere
- Oracle
- Microsoft SQL Server
- IBM DB2 LUW
- MySQL

Der MobiLink-Server verwendet eine WKB-Darstellung (Well Known Binary) der räumlichen Daten zusammen mit der ID ihres räumlichen Bezugssystems (SRID), um Daten aus entfernten Datenbanken in eine konsolidierte Datenbank hochzuladen. Außerdem verwendet er WKB und SRID für den Download von räumlichen Daten aus der konsolidierten Datenbank in die entfernten Datenbanken. Aus diesem Grund müssen die Tabellenskripte für Upload und Download in der Lage sein, räumliche Daten im WKB-Format zu verarbeiten.

## Dimensionale Einschränkungen

Der MobiLink-Server ist in der Lage, zweidimensionale, dreidimensionale und vierdimensionale räumliche Daten zwischen der konsolidierten Datenbank und den entfernten Datenbanken zu synchronisieren, wenn die konsolidierte Datenbank auf einem SQL Anywhere-Server läuft. Wenn jedoch die konsolidierte Datenbank einen der anderen unterstützten RDBMS-Typen aufweist, kann der MobiLink-Server nur zweidimensionale Daten zwischen der konsolidierten Datenbank und den entfernten Datenbanken synchronisieren.

Wenn ein Upload-Datenstrom dreidimensionale bzw. vierdimensionale räumliche Daten enthält und die konsolidierte Datenbank auf einem Nicht-SQL Anywhere-Datenbankserver läuft, generiert der MobiLink-Server eine Warnmeldung und löscht anschließend den dritten bzw. vierten Dimensionswert, bevor er die räumlichen Daten an die konsolidierte Datenbank sendet.

## Anforderungen an die SRID

Der MobiLink-Server sucht nicht automatisch nach einer Zuordnung zwischen den SRIDs in der konsolidierten und in der entfernten Datenbank. Sie müssen Folgendes sicherstellen: Entweder müssen die in den entfernten Datenbanken verwendeten SRIDs mit den in der konsolidierten Datenbank definierten übereinstimmen oder benutzerdefinierte Upload- und Download-Tabellenskripten müssen in der Lage sein, SRIDs zwischen den in der konsolidierten Datenbank und den entfernten Datenbank definierten SRIDs hin und her zu konvertieren.

## Benannte Parameter

Upload-Tabellenskripten können unter Verwendung von benannten Parametern oder Fragezeichen (?) geschrieben werden. Die Verwendung von Fragezeichen für SQL-Skripten ist veraltet und es wird empfohlen, dass Sie benannte Parameter verwenden. Wenn Fragezeichen verwendet werden, muss jede räumliche Spalte zwei Fragezeichen enthalten. Das erste davon steht für den WKB-Wert und das zweite für den SRID-Wert.

Wenn benannte Parameter verwendet werden, müssen die benannten Parameter der folgenden Konvention entsprechen:

```
{ml r.column_name:data}
{ml r.column_name:srid}
```

Der erste benannte Parameter repräsentiert den WKB-Wert und der zweite benannte Parameter die SRID.

## Hinweise zu Oracle

Beachten Sie beim Download von räumlichen Daten aus Oracle Folgendes:

- Oracle-Server generiert keine Well-Known-Binary-Werte (WKB), wenn ein Geometriewert mithilfe des nativen Oracle-Formats in eine Oracle-Datenbank eingegeben wird. Das folgende Beispiel zeigt eine Behelfslösung.

Daten im nativen Oracle-Format für räumliche Daten werden folgendermaßen eingegeben:

```
CREATE TABLE cola_markets (
  mkt_id NUMBER PRIMARY KEY,
  name VARCHAR2(32),
  shape SDO_GEOMETRY);
```

```

INSERT INTO cola_markets VALUES(
  1,
  'cola_a',
  SDO_GEOMETRY(
    2003, -- two-dimensional polygon
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,1003,3), -- one rectangle (1003 = exterior)
    SDO_ORDINATE_ARRAY(1,1, 5,7) -- only 2 points needed to
      -- define rectangle (lower left and upper right) with
      -- Cartesian-coordinate data
  )
);

```

Der Download-Cursor muss dann wie folgt geschrieben sein oder der standardmäßige Download-Cursor gibt 1 anstelle des tatsächlichen binären Werts zurück:

```

COLA_MARKETS (SCOTT): download_cursor
SELECT t.MKT_ID,
       t.NAME,

       SDO_UTIL.TO_WKBGEOMETRY(SDO_UTIL.FROM_WKTGEOMETRY(SDO_UTIL.TO_WKTGEOMETRY(t
       .SHAPE))),
       t.SHAPE.sdo_srid
FROM SCOTT.COLA_MARKETS t

```

- Das standardmäßige Downloadskript funktioniert korrekt, wenn die Daten im WKT-Format eingegeben wurden, und zwar folgendermaßen:

```

INSERT INTO cola_markets VALUES(
  1,
  'cola_a',
  SDO_GEOMETRY( 'polygon (( 1 1, 5 1, 5 7, 1 7, 1 1))',4326));
INSERT INTO SCOTT.cola_markets VALUES(
  2,
  'cola_b',
  SDO_GEOMETRY( 'polygon (( 5 1, 8 1, 8 6, 5 7, 5 1))',4326));
INSERT INTO SCOTT.cola_markets VALUES(
  3,
  'cola_c',
  SDO_GEOMETRY( 'polygon (( 3 3, 6 3, 6 5, 4 5, 3 3))',4326));

```

- Oracle verfügt nicht über eine SRID von 0, sodass alle planen kartesischen Ebenendaten mit einer SRID von 4326 eingegeben werden müssen.

### Siehe auch

- „Räumliche Daten“ [[SQL Anywhere Server - Unterstützung für räumliche Daten](#)]
- Benannte Skriptparameter auf Seite 236
- „Benutzerdefinierte benannte Parameter“ auf Seite 252

## Upload- und Download-Skripten

Wenn eine Tabelle räumliche Spalten enthält, können die Upload- und Download-Skripten sehr unterschiedlich aussehen, abhängig vom Typ der verwendeten konsolidierten Datenbank. Die folgenden Beispiele zeigen einige Upload- und Download-Beispielskripten für räumliche Daten für die konsolidierten Datenbanken, die zurzeit vom MobiLink-Server unterstützt werden.

In den Beispielen unten wird angenommen, dass die Synchronisationstabelle wie folgt aussieht:

```
create table test (c1 int not null primary key, c2 st_geometry )
```

Konkrete Hinweise zu den verschiedenen Typen von konsolidierten Datenbanken finden Sie unter:

- „Konsolidierte SQL Anywhere-Datenbank“
- „Konsolidierte Microsoft SQL Server-Datenbank“
- „Konsolidierte MySQL-Datenbank“
- „Konsolidierte Oracle-Datenbank“
- „Konsolidierte SAP Sybase IQ-Datenbank“

### SQL Anywhere-Beispielskripten

Nachstehend finden Sie ein Beispiel für ein upload\_insert-Skript für SQL Anywhere:

```
INSERT INTO test VALUES( {ml r.c1}, ST_Geometry::ST_GeomFromBinary({ml  
r.c2:data},{ml r.c2:srid}) )
```

Nachstehend finden Sie ein Beispiel für ein download\_cursor-Skript für SQL Anywhere:

```
SELECT c1, c2.ST_AsBinary(), c2.ST_SRID() FROM test
```

### Microsoft SQL Server-Beispielskripten

Nachstehend finden Sie ein Beispiel für ein upload\_insert-Skript für Microsoft SQL Server:

```
BEGIN
    DECLARE @c1 INTEGER
    DECLARE @v1 VARBINARY(max)
    DECLARE @s1 INTEGER
    DECLARE @g1 geometry
    SELECT @c1 = {ml r.c1}
    SELECT @v1 = {ml r.c2:data}
    SELECT @s1 = {ml r.c2:srid}
    IF @v1 IS NULL
        SELECT @g1 = NULL
    ELSE
        SELECT @g1 = Geometry::STGeomFromWKB(@v1,@s1)
    INSERT INTO test VALUES( @c1, @g1 )
END
```

Nachstehend finden Sie ein Beispiel für ein download\_cursor-Skript für Microsoft SQL Server:

```
SELECT c1, c2.STAsBinary(), c2.STSrid FROM test
```

### Oracle-Beispielskripten

Nachstehend finden Sie ein Beispiel für ein upload\_insert-Skript für Oracle:

```
DECLARE
    v_c1 INTEGER;
    v_v1 sdo_geometry;
    v_s1 INTEGER;
    v_u1 blob;
BEGIN
    v_c1 := {ml r.c1};
    v_u1 := {ml r.c2:data};
    v_s1 := {ml r.c2:srid};
```

```
IF v_ul IS NULL THEN
    v_vl := NULL;
ELSE
    v_vl := sdo_geometry( v_ul, v_sl );
END IF;
INSERT INTO test VALUES( v_c1, v_vl );
END;
```

Nachstehend finden Sie ein Beispiel für ein download\_cursor-Skript für Oracle:

```
SELECT c1, g.c2.Get_WKB(), g.c2.sdo_srid FROM test g
```

### IBM DB2-Beispielskripten

Nachstehend finden Sie ein Beispiel für ein upload\_insert-Skript für IBM DB2:

```
BEGIN ATOMIC
DECLARE v_c1 INTEGER;
DECLARE v_v1 BLOB(10m);
DECLARE v_sl INTEGER;
SET v_c1 = {ml r.c1}; SET v_v1 = {ml r.c2:data}; SET v_sl = {ml
r.c2:srid};
INSERT INTO test VALUES( v_c1, ST_Geometry( v_v1, v_sl ) );
END
```

Nachstehend finden Sie ein Beispiel für ein download\_cursor-Skript für IBM DB2:

```
SELECT c1, ST_AsBinary( c2 ), ST_SRID( c2 ) FROM test
```

### MySQL-Beispielskripten

Nachstehend finden Sie ein Beispiel für ein upload\_insert-Skript für MySQL:

```
INSERT INTO test VALUES( {ml r.c1}, GeometryFromWKB({ml r.c2:data},{ml
r.c2:srid}) )
```

Nachstehend finden Sie ein Beispiel für ein download\_cursor-Skript für MySQL:

```
SELECT c1, AsBinary( c2 ), SRID( c2 ) FROM test
```

## Konsolidierte Adaptive Server Enterprise-Datenbank

### Voraussetzungen

Vor dem Ausführen des Setupskripts sollten Sie die folgenden Anforderungen berücksichtigen:

- Der gleiche Datenbankbenutzer, der das Setupskript ausführt, wird als Benutzer erwartet, der die MobiLink-Systemtabellen während der Synchronisation aktualisiert. Dieser Benutzer muss verwendet werden, um den MobiLink-Server zu starten und MobiLink-Anwendungen zu konfigurieren. Siehe [Erforderliche Privilegien auf Seite 31](#).
- Der RDBMS-Benutzer, der mithilfe des MobiLink-Servers eine Verbindung mit der konsolidierten Datenbank herstellt, muss in der Lage sein, die MobiLink-Systemtabellen, -Prozeduren usw. ohne Qualifizierer zu verwenden (Beispiel: SELECT \* from ml\_user). Siehe [„Systemtabellen des MobiLink-Servers“ auf Seite 4](#).

- Die Login-ID für den MobiLink-Server muss ein SELECT-Privileg für MASTER haben. SYSTRANSACTIIONS besitzen.
- Die Login-ID für den MobiLink-Server muss die dtm\_tm\_role-Rolle haben, wenn die Option -cs für mlsrv16 verwendet wird.
- Sie müssen die sp\_dboption-Option verwenden, um die SELECT INTO-Option auf TRUE zu setzen. Führen Sie beispielsweise das folgende Skript in Interactive SQL aus, um das SELECT INTO-Privileg für **your-database-name** auf TRUE zu setzen:

```
sp_dboption your-database-name, "SELECT INTO", true
go
```

### Adaptive Server Enterprise als konsolidierte Datenbank einrichten

Um Adaptive Server Enterprise für die Verwendung als MobiLink-konsolidierte Datenbank einzurichten, müssen Sie eine Einrichtungsprozedur durchführen, die Systemtabellen, gespeicherte Prozeduren, Trigger und Ansichten hinzufügt, die für die MobiLink-Synchronisation erforderlich sind. Es gibt mehrere Möglichkeiten, dies durchzuführen:

- Führen Sie das Setupskript *syncase.sql* aus, das sich im Verzeichnis *%SQLANY16%\MobiLink\Setup* befindet.
- Prüfen und Aktualisieren der MobiLink-Systemeinrichtung von Sybase Central. Siehe „[MobiLink-Systemkonfiguration](#)“ [[MobiLink - Erste Orientierung](#)].

### ODBC-Treiber

Für die konsolidierte Adaptive Server Enterprise-Datenbank muss eine ODBC-Datenquelle (DSN) eingerichtet werden. Verwenden Sie dazu den ODBC-Treiber, der mit der Adaptive Server Enterprise-Datenbank zur Verfügung gestellt wurde. Siehe:

- <http://www.sybase.com/detail?id=1011880>
- Adaptive Server Enterprise-Dokumentation

### Hinweise zu Adaptive Server Enterprise

- **Konfigurationsparameter Enable functionality group** Wenn die Konfigurationsparameter **enable functionality group** auf dem ASE 15.7-Server aktiviert ist, verwendet der MobiLink-Server die Funktion "select ... for update", um entfernte IDs zu sperren und so zu verhindern, dass redundante Synchronisationen für dieselbe entfernte ID gleichzeitig ausgeführt werden. Wenn Sie den Parameter **enable functionality group** deaktivieren, müssen Sie alle derzeit mit dem ASE-Server verbundenen MobiLink-Server neu starten, um zu verhindern, dass Synchronisationsanforderungen fehlschlagen.
- **Datentypen zuordnen** Die Datentypen von Spalten müssen korrekt zwischen der konsolidierten und der entfernten Datenbank abgestimmt und zugeordnet werden. Siehe „[Zuordnung der Datentypen von Adaptive Server Enterprise](#)“ auf Seite 714.
- **CHAR-Spalten** In Adaptive Server Enterprise haben CHAR-Datentypen eine feste Länge und werden bis zur vollen Länge der Zeichenfolge mit Leerzeichen aufgefüllt. In entfernten MobiLink-Datenbanken (SQL Anywhere oder UltraLite) entspricht CHAR dem Datentyp VARCHAR: Werte

werden nicht bis zu einer bestimmten Länge mit Leerzeichen aufgefüllt. Verwenden Sie in der konsolidierten Datenbank unbedingt VARCHAR anstelle von CHAR. Wenn Sie CHAR verwenden müssen, können Sie mit der mlsrv16-Befehlszeilenooption -b während der Synchronisation nachgestellte Leerzeichen aus Zeichenfolgen entfernen. Diese Option ist wichtig für Vergleiche von Zeichenfolgen, mit denen Konflikte erkannt werden können.

Weitere Hinweise finden Sie unter „[mlsrv16-Option -b](#)“ auf Seite 51.

- **BLOB-Größen** Zum Herunterladen von Daten mit einer Größe von mehr als 32 kB (Standardgröße), führen Sie folgende Schritte aus:
  - Unter Windows legen Sie im Fenster für die Konfiguration des Adaptive Server Enterprise ODBC-Treibers auf der Seite **Advanced (Erweitert)** die Option **Text Size (Textgröße)** auf einen Wert fest, der größer ist als die größte erwartete BLOB-Größe.
  - Unter Linux setzen Sie den TextSize-Eintrag in der Datei *odbc.ini* auf einen Wert, der größer ist als die größte zu erwartende BLOB-Größe.
- **Einschränkungen für VARBIT** Die Synchronisation von VARBIT- oder LONG VARBIT-Werten mit einer Länge von 0 (d.h. leer) für eine konsolidierte Adaptive Server Enterprise-Datenbank wird von MobiLink nicht unterstützt. Da Adaptive Server Enterprise keine VARBIT-Typen unterstützt, werden diese normalerweise in eine VARCHAR- oder TEXT-Spalte in der Adaptive Server Enterprise-Datenbank synchronisiert. In Adaptive Server Enterprise werden leere Zeichenfolgenwerte in ein einzelnes Leerzeichen konvertiert. In SQL Anywhere sind Leerzeichen in einer VARBIT-Spalte nicht zulässig. Ein Downloaden dieser Werte verursacht also einen Fehler in der entfernten Datenbank.

## Isolationsstufe

Siehe „[MobiLink-Isolationsstufen](#)“ auf Seite 140.

# Konsolidierte IBM DB2 LUW-Datenbank

MobiLink unterstützt IBM DB2 LUW für Linux und Windows.

## IBM DB2 LUW als konsolidierte Datenbank einrichten

Um IBM DB2 für die Verwendung als konsolidierte MobiLink-Datenbank einzurichten, müssen Sie eine Setupprozedur durchführen, die Systemtabellen, gespeicherte Prozeduren, Trigger und Ansichten hinzufügt, die für die MobiLink-Synchronisation erforderlich sind.

## Voraussetzungen

Vor dem Ausführen des Setupskripts sollten Sie die folgenden Anforderungen berücksichtigen:

- Der gleiche Datenbankbenutzer, der das Setupskript ausführt, wird als Benutzer erwartet, der die MobiLink-Systemtabellen während der Synchronisation aktualisiert. Dieser Benutzer muss verwendet werden, um den MobiLink-Server zu starten und MobiLink-Anwendungen zu konfigurieren. Siehe [Erforderliche Privilegien auf Seite 31](#).
- Der RDBMS-Benutzer, der mithilfe des MobiLink-Servers eine Verbindung mit der konsolidierten Datenbank herstellt, muss in der Lage sein, die MobiLink-Systemtabellen, -Prozeduren usw. ohne

Qualifizierer zu verwenden (Beispiel: `SELECT * from ml_user`). Siehe „Systemtabellen des MobiLink-Servers“ auf Seite 4.

## Kontext und Bemerkungen

Sie können die MobiLink-Systemeinrichtung auch über Sybase Central prüfen und aktualisieren. Siehe „MobiLink-Systemkonfiguration“ [*MobiLink - Erste Orientierung*].

## Aufgabe

1. Um MobiLink-Systemtabellen mit dem Setupskript zu installieren, müssen Sie ein IBM DB2 LUW-Tablespace mit mindestens 8-kB-Seiten verwenden. Falls der Tablespace keine 8-kB-Seiten verwendet, führen Sie folgende Schritte aus.

- Überprüfen Sie anhand mindestens eines Ihrer Pufferpools, ob 8-kB-Seiten verwendet werden. Wenn nicht, erstellen Sie einen Pufferpool mit 8-kB-Seiten.
- Erstellen Sie einen neuen Tablespace und einen temporären Tablespace, die einen Pufferpool mit Seiten von 8 kB verwenden.

Weitere Hinweise finden Sie in Ihrer IBM DB2 LUW-Dokumentation.

2. Ergänzen Sie `syncdb2.sql` mit Ihren Verbindungsdaten:

- a. Kopieren Sie `syncdb2.sql` an einen neuen Speicherort, wo die Datei geändert und wieder abgespeichert werden kann.
- b. Das Skript `syncdb2.sql` enthält die Standard-Verbindungsanweisung `connect to DB2Database`. Ändern Sie diese Zeile, um eine Verbindung zu Ihrer IBM DB2-Datenbank herzustellen. Verwenden Sie die folgende Syntax:

```
connect to DB2Database user userid using password ~
```

Dabei gilt: *DB2Database*, *userid* und *password* sind von Ihnen bereitgestellte Werte. (Im Skript `syncdb2.sql` wird die Tilde (~) als Trennzeichen zwischen Befehlen verwendet.)

3. Führen Sie `syncdb2.sql` aus:

```
db2 -c -ec -td~ +s -v -f syncdb2.sql
```

## Ergebnisse

MobiLink-Systemtabellen, gespeicherte Prozeduren, Trigger und Ansichten werden installiert und IBM DB2 LUW kann als konsolidierte Datenbank verwendet werden.

## Nächste Schritte

Für die konsolidierte IBM DB2-Datenbank muss eine ODBC-Datenquelle (DSN) eingerichtet werden. Verwenden Sie dazu den ODBC-Treiber, der mit der IBM DB2-Datenbank zur Verfügung gestellt wurde. Siehe:

- <http://www.sybase.com/detail?id=1011880>
- IBM DB2 LUW-Dokumentation

## Hinweise zu IBM DB2 LUW

- **Sperrenerweiterung** Um die Datenkonsistenz zwischen konsolidierter und entfernter Datenbank aufrechtzuerhalten, gibt der MobiLink-Server die folgende Abfrage über die gespeicherte Prozedur `ml_lock_rid` aus, um die entfernte ID in jeder Synchronisationsphase zu sperren.

```
SELECT sync_key into p_sync_key FROM ml_database WHERE rid =  
a_given_remote_id WITH RR USE AND KEEP EXCLUSIVE LOCKS;
```

Diese Abfrage sperrt die entfernte ID im Exklusivmodus, um gleichzeitige Synchronisationen mit derselben entfernten ID zu verhindern.

Wenn Sperrenfehler für entfernte MobiLink-IDs auftreten, wenn keine gleichzeitigen Synchronisationen derselben entfernten ID vorliegen, z.B. wenn der MobiLink-Fehlercode -10341 im Log des MobiLink-Servers ausgegeben wird, sollten Sie die DB2-Konfigurationsparameter `maxlocks` und `locklist` ändern, um eine Sperrenausweitung zu verhindern. Weitere Hinweise finden Sie in Ihrer DB2-Dokumentation.

- **Datentypen zuordnen** Die Datentypen von Spalten müssen korrekt zwischen der konsolidierten und der entfernten Datenbank abgestimmt und zugeordnet werden. Siehe „IBM DB2 LUW-Datentypzuordnung“ auf Seite 723.
- **CHAR-Spalten** In IBM DB2 LUW haben CHAR-Datentypen eine feste Länge und werden bis zur vollen Länge der Zeichenfolge mit Leerzeichen aufgefüllt. In entfernten MobiLink-Datenbanken (SQL Anywhere oder UltraLite) entspricht CHAR dem Datentyp VARCHAR: Werte werden nicht bis zu einer bestimmten Länge mit Leerzeichen aufgefüllt. Verwenden Sie in der konsolidierten Datenbank unbedingt VARCHAR anstelle von CHAR. Wenn Sie CHAR verwenden müssen, können Sie mit der `mlsrv16`-Befehlszeilenoption `-b` während der Synchronisation nachgestellte Leerzeichen aus Zeichenfolgen entfernen. Diese Option ist wichtig für Zeichenfolgenvergleiche, mit denen Konflikte erkannt werden.

Siehe „`mlsrv16`-Option `-b`“ auf Seite 51.

- **Tablespace-Kapazität** Der Tablespace und der temporäre Tablespace einer IBM DB2 LUW-Datenbank, die als konsolidierte Datenbank verwendet werden soll, muss mindestens 8-kB-Seiten verwenden.

Außerdem gibt es Spalten, für die ein Tablespace vom Typ LONG erforderlich sind. Wenn es in der Standardkonfiguration keinen Tablespace vom Typ LONG gibt, müssen die Anweisungen zum Erstellen der Tabelle mit diesen Spalten entsprechend angegeben werden. Dazu ein Beispiel:

Ein Beispiel in der Beispielanwendung finden Sie unter „MobiLink-Beispiel CustDB“ [*MobiLink - Erste Orientierung*].

- ## Isolationsstufe

# Konsolidierte Microsoft SQL Server-Datenbank

Vor dem Ausführen des Setupskripts sollten Sie die folgenden Anforderungen berücksichtigen:

- ```
grant view server state to user_name
```

- ## Microsoft SQL Server als konsolidierte Datenbank einrichten

Copyright © 2013, SAP AG oder ein SAP-Konzernunternehmen. - SAP Sybase SQL Anywhere 16.10

- Führen Sie das Setupskript *syncmss.sql* aus, das sich im Verzeichnis *%SQLANY16%\MobiLink\Setup* befindet.
- Prüfen und Aktualisieren der MobiLink-Systemeinrichtung von Sybase Central. Siehe „[MobiLink-Systemkonfiguration](#)“ [*MobiLink - Erste Orientierung*].

### ODBC-Treiber

Für die konsolidierte SQL Server-Datenbank muss eine ODBC-Datenquelle (DSN) eingerichtet werden. Verwenden Sie dazu den ODBC-Treiber, der mit der SQL Server-Datenbank zur Verfügung gestellt wurde. Siehe:

- <http://www.sybase.com/detail?id=1011880>
- Microsoft SQL Server-Dokumentation

### Hinweise zu SQL Server

- **Datentypen zuordnen** Die Datentypen von Spalten müssen korrekt zwischen der konsolidierten und der entfernten Datenbank abgestimmt und zugeordnet werden. Siehe „[Microsoft SQL Server-Datentypzuordnung](#)“ auf Seite 733.
- **BLOB-Spalten** Aufgrund der Einschränkungen der SQL Server ODBC-Treiber sollten beim Definieren einer Synchronisationstabelle alle BLOB-Spalten am Ende von Spalten platziert werden, insbesondere wenn das *download\_cursor*-Skript für die Tabelle als gespeicherter Prozeduraufruf oder als Batch von SQL-Anweisungen geschrieben werden muss. Diese Einschränkung kann ignoriert werden, wenn das *download\_cursor*-Skript für eine Synchronisationstabelle mit BLOB-Spalten wird als einzelne SELECT-Anweisung geschrieben wird. Siehe <http://support.microsoft.com/kb/238080>.
- **CHAR-Spalten** In Microsoft SQL Server haben CHAR-Datentypen eine feste Länge und werden bis zur vollen Länge der Zeichenfolge mit Leerzeichen aufgefüllt. In entfernten MobiLink-Datenbanken (SQL Anywhere oder UltraLite) entspricht CHAR dem Datentyp VARCHAR: Werte werden nicht bis zu einer bestimmten Länge mit Leerzeichen aufgefüllt. Es wird empfohlen, in der konsolidierten Datenbank VARCHAR statt CHAR zu verwenden. Wenn Sie CHAR verwenden müssen, können Sie mit der *mlsrv16*-Befehlszeilenoption *-b* während der Synchronisation nachgestellte Leerzeichen aus Zeichenfolgen entfernen. Diese Option ist wichtig für Zeichenfolgenvergleiche, mit denen Konflikte erkannt werden.

Siehe „[mlsrv16-Option -b](#)“ auf Seite 51.

- **SET NOCOUNT ON** Für Microsoft SQL Server müssen Sie SET NOCOUNT ON als erste Anweisung in allen gespeicherten Prozeduren oder SQL-Batches angeben, die über ODBC ausgeführt werden.
- **Prozeduraufrufe** Microsoft SQLServer erfordert, dass Prozeduraufrufe mit Parametern unter Verwendung der ODBC-Syntax geschrieben werden:  

```
{ CALL procedure_name( {ml param1}, {ml param2}, ... ) }
```
- **Die Beispieldatenbank verwendet berechnete Spalten** Die SQL Server-Beispieldatenbank AdventureWorks enthält berechnete Spalten. Sie können keinen Upload in eine berechnete Spalte durchführen. Sie können die Spalte für reinen Download definieren oder die Spalte von der Synchronisation ausschließen.

- **Konflikterkennung in einem upload\_update-Skript implementieren** Für SQL Server müssen Sie die Konflikterkennung und -lösung im upload\_update-Skript durchführen. Siehe „[Konfliktlösung mit upload\\_update-Skripten](#)“ auf Seite 130.

### Isolationsstufe

Siehe „[MobiLink-Isolationsstufen](#)“ auf Seite 140.

## Konsolidierte MySQL-Datenbank

Der MobiLink-Server unterstützt MySQL Community- und Enterprise-Server der Version 5.1.3 oder höher.

### Privilegien

Vor dem Ausführen des Setupskripts sollten Sie die folgenden Anforderungen berücksichtigen:

- Der gleiche Datenbankbenutzer, der das Setupskript ausführt, wird als Benutzer erwartet, der die MobiLink-Systemtabellen während der Synchronisation aktualisiert. Dieser Benutzer muss verwendet werden, um den MobiLink-Server zu starten und MobiLink-Anwendungen zu konfigurieren. Siehe [Erforderliche Privilegien auf Seite 31](#).
- Der RDBMS-Benutzer, der mithilfe des MobiLink-Servers eine Verbindung mit der konsolidierten Datenbank herstellt, muss in der Lage sein, die MobiLink-Systemtabellen, -Prozeduren usw. ohne Qualifizierer zu verwenden (Beispiel: SELECT \* from ml\_user). Siehe „[Systemtabellen des MobiLink-Servers](#)“ auf Seite 4.

### Speicher-Engine

Der MobiLink-Server erfordert, dass die Standard-Speicher-Engine ACID-kompatibel ist. Wenn die Standard-Speicher-Engine nicht ACID-kompatibel ist, müssen Sie sicherstellen, dass alle Tabellen des MobiLink-Servers mit einer ACID-kompatiblen Speicher-Engine erstellt werden, wie z.B. InnoDB und Falcon. Andernfalls können Dateninkonsistenzen auftreten.

Vor dem Übernehmen der Datei für Ihre MySQL-Datenbank müssen Sie ggf. die Skriptdatei *syncmys.sql* bearbeiten und die folgenden zwei Zeilen nach der Zeile **delimiter //** hinzufügen, wobei *engine\_name* für eine ACID-konforme Speicher-Engine steht.

```
set storage_engine = [engine_name]
//
```

### MySQL mit dem Setupskript als konsolidierte Datenbank einrichten

Um MySQL für die Verwendung als MobiLink-konsolidierte Datenbank einzurichten, müssen Sie eine Einrichtungsprozedur durchführen, die Systemtabellen, gespeicherte Prozeduren, Trigger und Ansichten hinzufügt, die für die MobiLink-Synchronisation erforderlich sind.

### Voraussetzungen

Ihre MySQL-Benutzer-ID muss Privilegien zum Erstellen von Tabellen, Prozeduren, Funktionen, Ansichten und Triggern haben.

## Kontext und Bemerkungen

Sie können auch das MobiLink 16-Plug-In für Sybase Central verwenden, um MySQL als konsolidierte Datenbank einzurichten.

## Aufgabe

- Führen Sie mithilfe des MySQL-Befehlszeilentools oder des MySQL Query Browsers das Setupskript *syncmys.sql* aus, das sich im Verzeichnis *%SQLANY16%\MobiLink\Setup* befindet.

## Ergebnisse

Die MySQL-Datenbank ist bereit für die Verwendung als konsolidierte MobiLink-Datenbank.

## MySQL mit Sybase Central als konsolidierte Datenbank einrichten

Um MySQL für die Verwendung als MobiLink-konsolidierte Datenbank einzurichten, müssen Sie eine Einrichtungsprozedur durchführen, die Systemtabellen, gespeicherte Prozeduren, Trigger und Ansichten hinzufügt, die für die MobiLink-Synchronisation erforderlich sind.

## Voraussetzungen

Ihre MySQL-Benutzer-ID muss Privilegien zum Erstellen von Tabellen, Prozeduren, Funktionen, Ansichten und Triggern haben.

## Kontext und Bemerkungen

Sie können auch das Setupskript *syncmys.sql* verwenden, um MySQL als konsolidierte Datenbank einzurichten.

## Aufgabe

1. Wählen Sie im MobiLink 16-Plug-In für Sybase Central im Menü **Ansicht** die Option **Ordner**.
2. Öffnen Sie Ihr MobiLink-Projekt und erweitern Sie **Konsolidierte Datenbanken**.
3. Klicken Sie mit der rechten Maustaste auf den Datenbanknamen, und klicken Sie auf **MobiLink-Systemeinrichtung überprüfen**. Wenn die Datenbank ein Setup erfordert, werden Sie zum Fortfahren aufgefordert.

Wenn Sie eine vorhandene MobiLink-Systemeinrichtung verwenden möchten, muss *default\_schema* das Schema der MobiLink-Systemeinrichtung sein.

## Ergebnisse

Die MySQL-Datenbank ist bereit für die Verwendung als konsolidierte MobiLink-Datenbank.

## ODBC-Treiber

Für die konsolidierte MySQL-Datenbank muss eine ODBC-DSN eingerichtet werden. Verwenden Sie dazu den ODBC-Treiber, der auf der MySQL-Website zur Verfügung steht. Der MobiLink-Server unterstützt MySQL ODBC-Treiber 5.1.3 oder höher. Siehe:

- <http://www.sybase.com/detail?id=1011880>

Führen Sie zum Festlegen der ODBC-Konfigurationsdatei in UNIX eine der folgenden Aktionen aus:

- Speichern Sie die Datei *ODBC.INI* im Benutzerverzeichnis des aktuellen Benutzers.
- Erstellen Sie die Umgebungsvariable **ODBCINI** und setzen Sie sie auf den Pfad zum Verzeichnis der Datei *ODBC.INI* fest.

## Hinweise zu MySQL

- **Datentyp-Zuordnungen** Die Datentypen von Spalten müssen korrekt zwischen der konsolidierten Datenbank und den entfernten Datenbanken zugeordnet werden. Siehe „MySQL-Datenzuordnung“ auf Seite 740.
- **Gespeicherte Prozeduren** Es ist nicht möglich, die Parameter INOUT und OUT in Aufrufen aus gespeicherten Prozeduren zu verwenden. Prozeduren, die diese Parameter erfordern, müssen als Funktionen implementiert werden, die einen OUT-Wert zurückgeben.

Serverereignisse, die INOUT-Parameter erfordern, z.B. `authenticate_user` und `modify_user`, müssen als Funktionen implementiert werden und mit einer SELECT-Anweisung anstelle einer CALL-Anweisung ausgeführt werden.

- **Benannte Parameter** Benutzerdefinierte benannte Parameter werden nicht unterstützt.
- **Cursor-Skripten** Die Ereignisse `upload_fetch`, `download_cursor` und `download_delete_cursor` müssen mithilfe einer SELECT-Anweisung aufgerufen werden, die vom MobiLink-Server unter Verwendung einer Isolationsstufe zum Lesen festgeschriebener Anweisungen ausgeführt werden. Ein Bug im MySQL ODBC-Treiber führt dazu, dass nicht festgeschriebene Anweisungen, wie z.B. INSERT, UPDATE und DELETE-Anweisungen, vom Server gelesen werden. Dies führt zu dazu, dass die Daten in der konsolidierten und der entfernten Datenbank inkonsistent sind.

Als Behelfslösung für dieses Problem sollten Sie eine LOCK IN SHARE MODE-Klausel in Ihre SELECT-Anweisungen einbeziehen. Beispiel:

```
SELECT column1 FROM table1 WHERE id > 0 LOCK IN SHARE MODE
```

Diese Klausel schützt die SELECT-Anweisung vor nicht festgeschriebenen Vorgängen.

- **Massen-Upload** Der MySQL-ODBC-Treiber unterstützt den Massen-Upload derzeit nicht ordnungsgemäß.
- **MLSD** Der MySQL-ODBC-Treiber umfasst zurzeit keine Unterstützung für MSDTC, daher wird eine separate MLSD nicht unterstützt.
- **MySQL-Serverkonfiguration** Die MobiLink-Synchronisationsskripten sind in der Tabelle `ml_script` gespeichert und werden bei Bedarf von dort abgerufen. Sie müssen `max_allowed_packet` in der Datei *my.ini* möglicherweise auf 16 M oder höher festlegen.

- **Konflikterkennung** Die Skripten, die für die Konfliktlösung mit einer konsolidierten MySQL-Datenbank generiert werden, enthalten mehrere Anweisungen. Falls Sie mit der Konflikterkennung arbeiten, müssen Sie das Kontrollkästchen **Allow multiple statements** (Mehrere Anweisungen zulassen) auf der Seite **Flags 3** (Parameter 3) des Fensters **MySQL Connector/ODBC Data Source Configuration** (MySQL-Konnektor/ODBC-Datenquellen Konfiguration) aktivieren, wenn Sie einen DSN konfigurieren, die der MobiLink-Server für Verbindungen zu Ihrer MySQL-Datenbank verwendet.
- **Mehrere Anweisungen** Falls eines Ihrer Synchronisationsskripten SQL-Befehle in durch Semikolons getrennten Batches enthält, müssen Sie möglicherweise das Kontrollkästchen **Allow multiple statements** (Mehrere Anweisungen zulassen) auf der Seite **Flags 3** (Parameter 3) des Fensters **MySQL Connector/ODBC Data Source Configuration** (MySQL-Konnektor/ODBC-Datenquellen Konfiguration) aktivieren, wenn Sie einen DSN konfigurieren, die der MobiLink-Server für Verbindungen zu Ihrer MySQL-Datenbank verwendet.

### Isolationsstufe

Siehe „[MobiLink-Isolationsstufen](#)“ auf Seite 140.

## Konsolidierte Oracle-Datenbank

### Privilegien

Vor dem Ausführen des Setupskripts sollten Sie die folgenden Anforderungen berücksichtigen:

- Der gleiche Datenbankbenutzer, der das Setupskript ausführt, wird als Benutzer erwartet, der die MobiLink-Systemtabellen während der Synchronisation aktualisiert. Dieser Benutzer muss verwendet werden, um den MobiLink-Server zu starten und MobiLink-Anwendungen zu konfigurieren. Siehe [Erforderliche Privilegien auf Seite 31](#).
- Der RDBMS-Benutzer, der mithilfe des MobiLink-Servers eine Verbindung mit der konsolidierten Datenbank herstellt, muss in der Lage sein, die MobiLink-Systemtabellen, -Prozeduren usw. ohne Qualifizierer zu verwenden (Beispiel: SELECT \* from ml\_user).

Der RDBMS-Benutzer benötigt außerdem das SELECT-Privileg für GV\$TRANSACTION, GV\$SESSION, GV\$LOCK und DBA\_OBJECTS sowie EXECUTE-Privilegien für DBMS\_UTILITY. Sie können eine Berechtigung nicht direkt für die Synonyme GV\$TRANSACTION, GV\$SESSION und GV\$LOCK erteilen. Stattdessen müssen Sie die Berechtigung für die zugrunde liegenden dynamischen Performanceansichten GV\_\$TRANSACTION, GV\_\$SESSION und GV\_\$LOCK erteilen. Dazu müssen Sie eine Verbindung als SYS herstellen. Die Oracle-Syntax zur Erteilung dieser Berechtigung lautet:

```
grant select on SYS.GV_$TRANSACTION to user-name;

grant select on SYS.GV_$SESSION to user-name;

grant select on SYS.GV_$LOCK to user-name;

grant execute on SYS.DBMS_UTILITY to user-name;
```

## Oracle als konsolidierte Datenbank einrichten

Um Oracle für die Verwendung als MobiLink-konsolidierte Datenbank einzurichten, müssen Sie eine Einrichtungsprozedur durchführen, die Systemtabellen, gespeicherte Prozeduren, Trigger und Ansichten hinzufügt, die für die MobiLink-Synchronisation erforderlich sind. Es gibt mehrere Möglichkeiten, dies durchzuführen:

- Führen Sie das Setupskript *syncora.sql* aus, das sich im Verzeichnis *%SQLANY16%\MobiLink\Setup* befindet.
- Prüfen und Aktualisieren der MobiLink-Systemeinrichtung von Sybase Central. Siehe „[MobiLink-Systemkonfiguration](#)“ [[MobiLink - Erste Orientierung](#)].

## ODBC-Treiber

Für die konsolidierte Oracle-Datenbank muss eine ODBC-DSN eingerichtet werden. Siehe:

- „[SQL Anywhere 16 - Oracle-ODBC-Treiber](#)“
- <http://www.sybase.com/detail?id=1011880>

## Hinweise zu Oracle

- **MobiLink-Synchronisation und zeitstempelbasierte Downloads mit einem Oracle Real Application Cluster** Zeilen in der konsolidierten Datenbank, die auf einem Oracle RAC ausgeführt wird, können ausgelassen werden, wenn die Systemuhren der Knoten des Oracle-Clusters sich um mehr unterscheiden als die Zeit, die zwischen dem Abrufen des vorletzten Download-Zeitstempels durch den MobiLink-Server und dem Abrufen der herunterzuladenden Zeilen verstrichen ist. Dieses Problem ist auf einem RAC-System mit synchronisierten Knotensystemuhren unwahrscheinlich, aber die Wahrscheinlichkeit erhöht sich bei größeren Unterschieden von Systemuhren verschiedener Knoten. Eine Behelfslösung ist die Erstellung eines *modify\_next\_last\_download\_timestamp*- oder *modify\_last\_download\_timestamp*-Skripts zur Subtrahierung des maximalen Unterschieds der Knotensystemuhren.

Mindestens seit Version 10i empfiehlt Oracle die Verwendung des NTP-Protokolls (Network Time Protocol) für die Synchronisation der Systemuhren auf allen Knoten in einem Cluster. NTP läuft normalerweise standardmäßig unter Unix und Linux. Wenn Cluster-Knoten korrekt für die Verwendung von NTP konfiguriert sind, sollten alle Systemuhren innerhalb von 200 Mikrosekunden bis zu 10 Millisekunden liegen (abhängig von der Nähe des NT-Servers). Seit Windows Server 2003 implementiert der Windows Time Service die NTP Version 3, die standardmäßig ausgeführt wird. Ab Version 11gR2 enthält die Oracle Clusterware den Oracle Cluster Time Synchronization Service (CTSS) zur Überwachung der Synchronisation der Systemuhren oder, wenn weder NTP noch Windows Time Service ausgeführt wird, zur aktiven Aufrechterhaltung der Systemuhr-Synchronisation. CTSS und Windows Time Service sind jedoch weniger präzise als NTP.

Zur Vermeidung von fehlenden Zeilen wird folgendermaßen vorgegangen: Wenn Systemuhren von Oracle RAC-Knoten um bis zu eine Sekunde mehr voneinander abweichen, als die Zeit, die zwischen dem Abrufen des *next\_last\_download\_timestamp*-Werts und dem Herunterladen der Zeilen liegt, subtrahiert der MobiLink-Server eine Sekunde vom *next\_last\_download\_timestamp*-Wert, der aus der konsolidierten Datenbank abgerufen wurde, wenn die folgenden Bedingungen erfüllt sind:

- Das vom MobiLink-Server verwendete Oracle-Konto hat das EXECUTE-Privileg für SYS.DBMS\_UTILITY.

- Die konsolidierte Datenbank ist ein Oracle RAC-System.
- Für MobiLink-Versionen 12.0.0 und höher ist kein generate\_next\_last\_download\_timestamp-Skript vorhanden.

Bei Oracle RAC-Knoten, deren Systemuhren größere Unterschiede aufweisen, können Sie das Problem vermeiden, indem Sie ein generate\_next\_last\_download\_timestamp-, modify\_next\_last\_download\_timestamp- oder modify\_last\_download\_timestamp-Skript definieren, um den größten Systemuhrunterschied der Knoten zu beheben.

- **Datentypen zuordnen** Die Datentypen von Spalten müssen korrekt zwischen der konsolidierten und der entfernten Datenbank abgestimmt und zugeordnet werden. Siehe „[Oracle-Datentypzuordnung](#)“ auf Seite 746.
- **XMLTYPE-Datentyp** Die Verwendung des Oracle-Datentyps XMLTYPE in SQL Anywhere oder UltraLite erfordert besondere Vorsicht. Siehe [XMLTYPE-Datentyp in Oracle](#) auf Seite 23.
- **CHAR-Spalten** In Oracle haben CHAR-Datentypen eine feste Länge und werden bis zur vollen Länge der Zeichenfolge mit Leerzeichen aufgefüllt. In entfernten MobiLink-Datenbanken (SQL Anywhere oder UltraLite) entspricht CHAR dem Datentyp VARCHAR: Werte werden nicht bis zu einer bestimmten Länge mit Leerzeichen aufgefüllt. Verwenden Sie in der konsolidierten Datenbank unbedingt VARCHAR anstelle von CHAR. Wenn Sie CHAR verwenden müssen, können Sie mit der mlsrv16-Befehlszeilenooption -b während der Synchronisation nachgestellte Leerzeichen aus Zeichenfolgen entfernen. Diese Option ist wichtig für Zeichenfolgenvergleiche, mit denen Konflikte erkannt werden.

Siehe „[mlsrv16-Option -b](#)“ auf Seite 51.

- **Zeitstempel** Der MobiLink-Server generiert mithilfe von gv\$transaction einen Zeitstempel, der von der entfernten Datenbank bei der nächsten Synchronisation verwendet werden soll. Daher muss die Login-ID für den MobiLink-Server ein SELECT-Privileg für gv\$transaction haben. Oracle gestattet es Ihnen nicht, den Zugriff direkt auf gv\$transaction zu erteilen. Stattdessen müssen Sie das SELECT-Privileg für die zugrunde liegende Ansicht gv\_\$transaction erteilen. Siehe [Privilegien](#) auf Seite 20.
- **Gespeicherte Prozeduren** Wenn Sie gespeicherte Prozeduren verwenden, um Ergebnismengen zurückzugeben oder VARRAY-Parameter zu akzeptieren, müssen Sie die Option **Prozedur gibt Ergebnisse zurück oder verwendet VARRAY-Parameter** für den SQL Anywhere 16 - Oracle-ODBC-Treiber aktivieren. Außerdem benötigt Sybase Central Prozeduren für die Rückgabe von Ergebnissen, um die zentrale Administration von entfernten Datenbanken verwenden zu können, sodass diese Option bei Verwendung der zentralen Administration aktiviert werden muss.

Siehe „[SQL Anywhere 16 - Oracle-ODBC-Treiber](#)“ auf Seite 768.

- **Variablen für die gesamte Sitzung** Sie können sitzungsbezogene Daten in Variablen in Oracle-Paketen speichern. Mit Oracle-Paketen können Variablen erstellt, geändert und gelöscht werden und diese Variablen können verfügbar sein, so lange das betreffende Oracle-Paket aktuell ist.
- **Methoden für automatisches Inkrementieren** Um die Eindeutigkeit von Primärschlüsseln zu erhalten, können Sie mit einer Oracle-Sequenz eine Liste von Schlüsseln erstellen, die der in einem

SQL Anywhere-autoincrement-Feld ähnelt. In der Beispieldatenbank CustDB finden Sie Code-Beispiele unter *Samples\MobiLink\CustDB\custora.sql*. Im Unterschied zu einer autoincrement-Spalte müssen Sie jedoch die Sequenz explizit referenzieren. Bei SQL Anywhere-autoincrement-Spalten wird automatisch ein Spaltenwert eingefügt, wenn die Spalte nicht in einer INSERT-Anweisung referenziert ist.

- **Oracle unterstützt keine leeren Zeichenfolgen** In Oracle wird eine leere Zeichenfolge wie der Wert NULL behandelt. In SQL Anywhere und UltraLite haben leere Zeichenfolgen eine andere Bedeutung als der Wert NULL. Verwenden Sie daher keine leeren Zeichenfolgen in Ihren Client-Datenbanken, wenn Sie mit einer konsolidierten Oracle-Datenbank arbeiten.

### XMLTYPE-Datentyp in Oracle

Der Oracle-Datentyp XMLTYPE kann dem XML-Datentyp von SQL Anywhere zugeordnet werden oder den UltraLite-Datentypen LONG VARCHAR und VARCHAR(*n*). Es ist wichtig zu beachten, dass der Oracle-Datenbankserver die Daten vor dem Speichern in einer XMLTYPE-Spalte validiert, während SQL Anywhere und UltraLite dies nicht tun. Sie müssen daher sicherstellen, dass die hochzuladenden XML-Dokumente gültigen XML-Code enthalten.

Kleine XML-Dokumente mit einer Länge von höchstens 32 kB können mit Oracle PL/SQL-Anweisungen in eine Oracle-Datenbank hochgeladen und aus ihr heruntergeladen werden. Wenn die Länge von XML-Dokumenten größer ist als 32 kB, müssen XML-Dokumente für den Upload möglicherweise mit den Skripten `upload_insert` und `upload_update` in eine globale temporäre Tabelle hochgeladen werden. Anschließend können die Upload-Daten mit dem Skript `end_upload_rows` oder `end_upload` konvertiert und in der eigentlichen Synchronisationstabelle gespeichert werden.

Die folgenden Beispiele enthalten Upload- und Download-Beispielskripten für das Hoch- und Herunterladen von XMLTYPE-Objekten zwischen einer konsolidierten Oracle-Datenbank und entfernten SQL Anywhere-Datenbanken. In diesen Beispielen ist die Upload-Tabelle in der konsolidierten Oracle-Datenbank folgendermaßen definiert:

```
create table test (pk int not null primary key, c1 XMLTYPE)
```

Die Upload-Tabelle ist in der entfernten SQL Anywhere-Datenbank folgendermaßen definiert:

```
create table test (pk int not null primary key, c1 XML)
```

- **Wenn alle XML-Dokumente höchstens 32 kB groß sind, können die Upload- und Download-Skripten folgendermaßen geschrieben werden** `upload_insert`

```
declare v_pk integer; v_c1 clob; x_c1 xmltype;
begin
    v_pk := {ml r.pk};
    v_c1 := {ml r.c1};
    x_c1 := XMLTYPE.createXML( v_c1 );
    insert into test values( v_pk, x_c1 );
end;
```

`download_cursor`

```
select pk, XMLSERIALIZE( content c1 ) from test
```

Dieses upload\_insert-Skript funktioniert gut, wenn die XML-Datenlänge kleiner oder gleich 32 kB ist. Wenn jedoch die XML-Datenlänge größer ist als 32 kB, gibt der Oracle Server möglicherweise eine Fehlermeldung aus.

- **Falls XML-Dokumente länger sind als 32 kB, müssen die für den Upload vorgesehenen XML-Daten in eine globale temporäre Tabelle hochgeladen werden** Das upload\_insert-Skript lädt die XML-Dokumente in eine globale temporäre Tabelle in der konsolidierten Oracle-Datenbank hoch. Die globale temporäre Tabelle ist folgendermaßen definiert:

```
create global temporary table tmp_test (pk int, c1 CLOB)
```

Die upload\_insert-Skript kann folgendermaßen geschrieben werden:

```
insert into tmp_test values( {ml r.pk}, {ml r.c1} )
```

#### Hinweis

Die c1-Spalte in der temporären Tabelle muss den CLOB-Datentyp aufweisen.

Das end\_upload\_rows-Skript ruft die XML-Dokumente aus der globalen temporären Tabelle ab, konvertiert sie in XML-Dokumente und speichert anschließend die XML-Daten in der Testtabelle. Im Folgenden sehen Sie das end\_upload\_rows-Skript:

```
insert into test (pk, c1) (select pk, XMLTYPE.createXML(c1) from tmp_test
```

### Isolationsstufe

Siehe „[MobiLink-Isolationsstufen](#)“ auf Seite 140.

## Oracle-VARRAY

Der SQL Anywhere 16 - Oracle-ODBC-Treiber unterstützt die Verwendung des Oracle-Datentyps VARRAY in gespeicherten Prozeduren. Die Verwendung von VARRAY in Upload-Skripten (upload\_insert, upload\_update und upload\_delete), die in gespeicherten Prozeduren geschrieben sind, können die Performance des MobiLink-Servers im Vergleich zu Upload-Skripten verbessern, die in gespeicherten Prozeduren geschrieben sind und VARRAY nicht verwenden. Einfache SQL-Anweisungen, wie INSERT, UPDATE und DELETE, ohne gespeicherte Prozeduren, bieten für gewöhnlich die beste Performance. Die Verwendung von gespeicherten Prozeduren mit der VARRAY-Technik bietet jedoch eine Möglichkeit zur Anwendung von Geschäftslogik, die von einfachen Anweisungen nicht zur Verfügung gestellt wird.

### VARRAY-Beispiel

Das folgende einfache Beispiel verwendet VARRAY:

1. Erstellen Sie eine Tabelle mit dem Namen my\_table, die 3 Spalten enthält.

```
create table my_table ( pk integer primary key, c1 number(20), c2  
varchar2(4000) )
```

2. Erstellen Sie mithilfe von VARRAYs benutzerdefinierte Sammlungstypen.

```
create type my_integer is varray(100) of integer;
create type my_number is varray(100) of number(20);
create type my_varchar is varray(100) of varchar2(8000);
```

Der Datentyp `my_varchar` ist als `VARRAY` definiert, das 100 Elemente enthält, und jedes Element ist ein `varchar2`-Datentyp mit der Breite 8000. Die Breite muss doppelt so groß sein wie die für `my_table` festgelegte Breite.

3. Erstellen Sie gespeicherte Prozeduren für das Einfügen.

```
create or replace procedure my_insert_proc( pk_v my_integer, c1_v
my_number, c2_v my_varchar )
is
c2_value my_varchar;
begin
    c2_value := c2_v; -- Work around an Oracle bug
    FORALL i in 1 .. pk_v.COUNT
        insert into my_table ( pk, c1, c2 ) values( pk_v(i),
c1_v(i), c2_value(i) );
end;
```

## VARRAY-Einschränkungen

Die folgenden Einschränkungen gelten, wenn `VARRAY` in gespeicherten Prozeduren verwendet wird:

- Für die ODBC-Datenquelle muss das Kontrollkästchen **Verteilte Microsoft-Transaktionen aktivieren** deaktiviert sein.
- BLOB- und CLOB-VARRAYs werden nicht unterstützt.
- Wenn ein `VARRAY` vom Datentyp `CHAR`, `VARCHAR`, `NCHAR` oder `NVARCHAR` ist, muss der benutzerdefinierte `VARRAY`-Typ doppelt so groß sein wie die für die Tabellenspalte angegebene Länge.
- Die Anzahl der Zeilen im `VARRAY`, die vom MobiLink-Server an die konsolidierte Oracle-Datenbank gesendet werden, wird durch die Option `-s` festgelegt, nicht durch die im `VARRAY`-Typ deklarierten Größe. Die Option `-s` darf die Größe des kleinsten von den Synchronisationsskripten verwendeten `VARRAY`-Typs nicht übersteigen. Falls sie größer ist, wird vom MobiLink-Server ein Fehler ausgegeben. Siehe „[mlsrv16-Option -s](#)“ auf Seite 77.

## Konsolidierte SAP HANA-Datenbank

### Privilegien

Vor dem Ausführen des Setupskripts sollten Sie die folgenden Anforderungen berücksichtigen:

- Der Datenbankbenutzer, der das Setupskript ausführt, muss derselbe Benutzer sein, der während der Synchronisation auf die MobiLink-Systemtabellen zugreift. Dieser Benutzer muss verwendet werden, um den MobiLink-Server zu starten und MobiLink-Anwendungen zu konfigurieren. Siehe [Erforderliche Privilegien auf Seite 31](#).
- Der RDBMS-Benutzer, der mithilfe des MobiLink-Servers eine Verbindung mit der konsolidierten Datenbank herstellt, muss in der Lage sein, die MobiLink-Systemtabellen, -Prozeduren usw. ohne Qualifizierer zu verwenden (Beispiel: `SELECT * from ml_user`). Siehe „[Systemtabellen des MobiLink-Servers](#)“ auf Seite 4.

- Der RDBMS-Benutzer benötigt außerdem das CATALOG READ-Privileg.

### SAP HANA als konsolidierte Datenbank einrichten

Um SAP HANA für die Verwendung als konsolidierte MobiLink-Datenbank einzurichten, müssen Sie eine Setupprozedur ausführen, die MobiLink-Systemtabellen, gespeicherte Prozeduren und Ansichten hinzufügt, die für die MobiLink-Synchronisation erforderlich sind. Verwenden Sie die folgende Methode, um SAP HANA einzurichten.

- Führen Sie das Setupskript *synchana.sql* mit SAP HANA Studio aus. Die Skriptdatei befindet sich im Verzeichnis `%SQLANY16%\MobiLink\`.

### ODBC-Treiber

Der MobiLink-Server kann für SAP HANA-Datenbanken nur den ODBC-Treiber aus SAP HDBODBC erkennen. Siehe:

- <http://www.sybase.com/detail?id=1011880>
- Ihre SAP HANA-Dokumentation

### Gespeicherte Prozeduren

SAP HANA unterstützt nicht das Aufrufen von gespeicherten Prozeduren mit NULL-Werten für skalare Eingabeparameter. Deshalb sollten Sie die Skripten `upload_insert` und `upload_update` nicht als Aufrufe von gespeicherten Prozeduren implementieren, wenn Synchronisationstabellen nullwertfähige Spalten enthalten. Der MobiLink-Server ersetzt NULL-Werte für Parameter vom Typ CHAR oder BINARY durch Zeichenfolgen bzw. binäre Werte mit Nulllänge und übergibt diese dann beim Aufrufen der Benutzerauthentifizierungsskripten an die SAP HANA-Datenbank. Berücksichtigen Sie dieses Verhalten des MobiLink-Servers, wenn Sie die gespeicherten Prozeduren für die Benutzerauthentifizierung schreiben.

Die benutzerdefinierten Parameter werden ebenfalls auf Zeichenfolgen mit Nulllänge gesetzt, wenn sie erstmals referenziert werden.

### Hinweise zu SAP HANA

- **MobiLink-Systemdatenbank (MLSD)** Der MobiLink-Server unterstützt nicht MLSD für SAP HANA.
- **MobiLink-Server-Systemobjekte** Die Primärschlüsselspalten in den Systemtabellen des MobiLink-Servers werden von SAP HANA SEQUENCES durch Aufrufe gespeicherter Prozeduren verwaltet.

Verwenden Sie die Systemprozeduren `ml_add_connection_script` und `ml_add_table_script` zum Hinzufügen, Ändern oder Löschen von Verbindungs- und Tabellenskripten sowie die Systemprozeduren `ml_add_user` und `ml_add_database` zum Hinzufügen von MobiLink-Benutzern und entfernten Datenbanken. Fügen Sie Zeilen nicht direkt in die Systemtabellen des MobiLink-Servers ein.

Siehe:

- „ml\_add\_connection\_script-Systemprozedur“ auf Seite 651
- „ml\_add\_table\_script-Systemprozedur“ auf Seite 668
- „ml\_add\_user-Systemprozedur“ auf Seite 670

- **Zeitstempelbasierte Downloads** Da SAP HANA die Snapshot-Isolation unterstützt, verwendet der MobiLink-Server die Anfangszeit der ältesten offenen Transaktion mit dem aktuellen Benutzer als nächsten last\_download\_timestamp-Wert, sodass die Skripten download\_cursor und download\_delete\_cursor mithilfe der Informationen zu den zuletzt geänderten Spalten einen zeitstempelbasierten Download generieren können. Im Folgenden finden Sie ein Beispiel.

1. Erstellen Sie eine Synchronisationstabelle in SAP HANA.

```
create COLUMN table test (pk int primary key, cl int, last_modified
timestamp generated always as ( now() ) )
```

Die Klausel **generated always as ( now() )** bewirkt, dass der SAP HANA-Server die last\_modified-Spalte mit dem aktuellen Zeitstempel aktualisiert, sobald diese Zeile eingefügt oder aktualisiert wird.

2. Das download\_cursor-Skript kann folgendermaßen geschrieben werden.

```
select pk,cl from test where last_modified > {ml
s.last_table_download}
```

SAP HANA unterstützt keine Trigger, sodass Sie möglicherweise logische Löschungen verwenden müssen, um einen Löschungen-Download aus SAP HANA in die entfernten Datenbanken zu generieren.

Alternativ können die Skripten download\_cursor und download\_delete\_cursor auf Basis der ausgeblendeten Spalten \$validto\$ und \$validfrom\$ geschrieben werden, wenn die Tabelle mit der HISTORY COLUMN-Klausel erstellt wird, und der nächste last\_download\_timestamp-Wert kann vom generate\_next\_last\_download\_timestamp-Skript generiert werden. Weitere Hinweise zu diesen ausgeblendeten Spalten finden Sie in der SAP HANA-Dokumentation.

Wenn das Skript download\_cursor bzw. download\_delete\_cursor auf Tabellen zugreift, deren Eigentümer andere Datenbankbenutzer sind als der Benutzer, der vom MobiLink-Server verwendet wird, um sich bei der SAP HANA-Datenbank anzumelden, müssen Sie mit der folgenden Anweisung dem Login des MobiLink-Servers das **CATALOG READ**-Privileg erteilen:

```
GRANT CATALOG READ TO MobiLink_server_login_user_name
```

Dies ermöglicht es dem Login-Benutzer des MobiLink-Servers, alle offenen Transaktionen in der Datenbank anzuzeigen und nicht nur offene Transaktionen für den aktuellen Benutzer, wenn er versucht, die ältesten offenen Transaktion der Datenbank abzurufen.

- **Datentypen zuordnen** Die Datentypen von Spalten müssen korrekt zwischen der konsolidierten und der entfernten Datenbank abgestimmt und zugeordnet werden. Siehe „[Datentypzuordnung für SAP HANA-Datenbankserver](#)“ auf Seite 757.

### Isolationsstufe

Siehe „[MobiLink-Isolationsstufen](#)“ auf Seite 140.

## Konsolidierte SQL Anywhere-Datenbank

### Privilegien

Vor dem Ausführen des Setupskripts sollten Sie die folgenden Anforderungen berücksichtigen:

- Der gleiche Datenbankbenutzer, der das Setupskript ausführt, wird als Benutzer erwartet, der die MobiLink-Systemtabellen während der Synchronisation aktualisiert. Dieser Benutzer muss verwendet werden, um den MobiLink-Server zu starten und MobiLink-Anwendungen zu konfigurieren. Siehe [Erforderliche Privilegien auf Seite 31](#).
- Der RDBMS-Benutzer, der mithilfe des MobiLink-Servers eine Verbindung mit der konsolidierten Datenbank herstellt, muss in der Lage sein, die MobiLink-Systemtabellen, -Prozeduren usw. ohne Qualifizierer zu verwenden (Beispiel: `SELECT * from ml_user`). Siehe „[Systemtabellen des MobiLink-Servers](#)“ auf Seite 4.

### SQL Anywhere als konsolidierte Datenbank einrichten

Um SQL Anywhere für die Verwendung als MobiLink-konsolidierte Datenbank einzurichten, müssen Sie eine Einrichtungsprozedur durchführen, die Systemtabellen, gespeicherte Prozeduren, Trigger und Ansichten hinzufügt, die für die MobiLink-Synchronisation erforderlich sind. Es gibt mehrere Möglichkeiten, dies durchzuführen:

- Führen Sie das Setupskript *syncsa.sql* aus, das sich im Verzeichnis `%SQLANY16%\MobiLink\Setup` befindet.
- Prüfen und Aktualisieren der MobiLink-Systemeinrichtung von Sybase Central. Siehe „[MobiLink-Systemkonfiguration](#)“ [[MobiLink - Erste Orientierung](#)].

### ODBC-Treiber einrichten

Für die konsolidierte SQL Anywhere-Datenbank muss eine ODBC-Datenquelle (DSN) eingerichtet werden. Der ODBC-Treiber für SQL Anywhere wird mit SQL Anywhere installiert.

Informationen zum SQL Anywhere ODBC-Treiber finden Sie unter „[ODBC-Datenquellen](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

### Isolationsstufe

Siehe „[MobiLink-Isolationsstufen](#)“ auf Seite 140.

## Konsolidierte SAP Sybase IQ-Datenbank

### Privilegien

Vor dem Ausführen des Setupskripts sollten Sie die folgenden Anforderungen berücksichtigen:

- Der gleiche Datenbankbenutzer, der das Setupskript ausführt, wird als Benutzer erwartet, der die MobiLink-Systemtabellen während der Synchronisation aktualisiert. Dieser Benutzer muss verwendet werden, um den MobiLink-Server zu starten und MobiLink-Anwendungen zu konfigurieren. Siehe [Erforderliche Privilegien auf Seite 31](#).
- Der RDBMS-Benutzer, der mithilfe des MobiLink-Servers eine Verbindung mit der konsolidierten Datenbank herstellt, muss in der Lage sein, die MobiLink-Systemtabellen, -Prozeduren usw. ohne Qualifizierer zu verwenden (Beispiel: `SELECT * from ml_user`). Siehe [„Systemtabellen des MobiLink-Servers“ auf Seite 4](#).
- Die Login-ID für den MobiLink-Server muss das EXECUTE-Privileg für die SP\_IQTRANSACTION-Systemprozedur für SAP Sybase IQ haben.

### SAP Sybase IQ als konsolidierte Datenbank einrichten

Um SAP Sybase IQ für die Verwendung als konsolidierte MobiLink-Datenbank einzurichten, müssen Sie eine Setupprozedur ausführen, die MobiLink-Systemtabellen, gespeicherte Prozeduren, Trigger und Ansichten hinzufügt, die für die MobiLink-Synchronisation erforderlich sind. Es gibt mehrere Möglichkeiten, dies durchzuführen:

- Führen Sie das Setupskript `synciq.sql` aus, das sich im Verzeichnis `%SQLANY16%\MobiLink\Setup` befindet.
- Prüfen und Aktualisieren der MobiLink-Systemeinrichtung von Sybase Central. Siehe [„MobiLink-Systemkonfiguration“ \[MobiLink - Erste Orientierung\]](#).

### ODBC-Treiber einrichten

Für die konsolidierte SAP Sybase IQ-Datenbank müssen Sie einen ODBC-DSN einrichten. Der ODBC-Treiber für SAP Sybase IQ wird mit Sybase IQ installiert.

Weitere Hinweise zum ODBC-Treiber für SAP Sybase IQ finden Sie in der SAP Sybase IQ-Dokumentation.

### Hinweise zu SAP Sybase IQ

- Wenn Uploads Daten enthalten, durch die im SAP Sybase IQ-Speicher definierte Synchronisationstabellen geändert werden, und wenn der MobiLink-Server mit mehreren gleichzeitigen Datenbank-Worker-Threads ausgeführt wird, müssen alle Uploads serialisiert werden, weil der SAP Sybase IQ 15-Server es jeweils nur einer einzigen Verbindung erlaubt, eine bestimmte Tabelle im SAP Sybase IQ-Speicher zu ändern.

Diese Anforderung kann erfüllt werden, wenn das `begin_upload`-Verbindungsskript so geschrieben wird, dass es die folgende SQL-Anweisung enthält oder verwendet:

```
LOCK TABLE table_name IN WRITE MODE WAIT time_string
```

Dabei ist `table_name` der Name einer Tabelle, die im SAP Sybase IQ-Speicher definiert ist, und `time_string` gibt die maximale Zeitspanne an, für die die Tabelle gesperrt werden kann. Die Tabelle kann einfach gehalten werden, wie in der folgenden Definition:

```
create table coordinate_upload ( c1 int )
```

Die Tabelle muss keine Daten enthalten.

- Alle Transaktionen, die SAP Sybase IQ-Tabellen ändern, müssen serialisiert werden, unabhängig davon, ob sie auf MobiLink-Serververbindungen oder auf anderen Verbindungen mit der SAP Sybase IQ-Datenbank auftreten. Für MobiLink-Servertransaktionen kann die oben beschriebene Logik ebenfalls verwendet werden. Diese Methode ist effizienter, als den MobiLink-Server jede Transaktion automatisch wiederholen zu lassen, und liefert eine bessere Performance.
- Beim Erstellen eines Synchronisationsmodells für eine konsolidierte SAP Sybase IQ-Datenbank werden die Tabellenzuordnungen für SAP Sybase IQ-Tabellen standardmäßig auf "Nur Download" gesetzt. Wenn Sie Zuordnungen in "Bidirektional" oder "Nur Upload" ändern möchten, müssen Sie sicherstellen, dass Änderungen an diesen SAP Sybase IQ-Tabellen serialisiert werden. Dies ist z.B. durch das Hinzufügen eines begin\_upload-Ereignisses möglich, wie oben beschrieben.

### Isolationsstufe

Siehe „[MobiLink-Isolationsstufen](#)“ auf Seite 140.

## MobiLink-Server

Sämtliche MobiLink-Clients werden mithilfe des MobiLink-Servers synchronisiert. Keiner von ihnen kann sich direkt mit einem Datenbankserver verbinden. Daher müssen Sie zuerst den MobiLink-Server starten, bevor Sie einen MobiLink-Client synchronisieren können.

Der MobiLink-Server öffnet über ODBC Datenbankverbindungen zum Server Ihrer konsolidierten Datenbank. Danach akzeptiert er Verbindungen von entfernten Anwendungen und steuert den Synchronisationsprozess.

#### Hinweis

Mithilfe der mlsrv16-Optionen können Sie die Arbeitsweise des MobiLink-Servers festlegen. Um die Aktionen des Servers während der Synchronisation zu steuern, erstellen Sie Skripten, die bei Synchronisationsereignissen aufgerufen werden. Siehe „[Synchronisationsereignisse](#)“ auf Seite 272.

Führen Sie mlsrv16 aus, um den MobiLink-Server zu starten. Geben Sie die ODBC-Verbindungsparameter für Ihre konsolidierte Datenbank mit der Option -c an. Eine Liste der mlsrv16-Befehlszeilenoptionen finden Sie unter „[MobiLink-Serveroptionen](#)“ auf Seite 44.

Sie müssen Verbindungsparameter angeben. Es stehen noch weitere Optionen zur Verfügung, diese sind aber optional. Mit diesen Optionen können Sie festlegen, wie der Server arbeitet. Sie können zum Beispiel die Cachegröße und Protokollierungsoptionen angeben. Weitere Hinweise zu den Verbindungsparametern finden Sie unter „[mlsrv16-Option -c](#)“ auf Seite 53.

Der MobiLink-Server benötigt einen ODBC-Datenquellennamen (DSN) für die Kommunikation mit der konsolidierten Datenbank. Ein DSN enthält Informationen, aus denen der ODBC-Treiber-Manager entnimmt, wo er den ODBC-Treiber laden soll. Unter Windows können ODBC-Datenquellen mit dem ODBC-Datenquellen-Administrator von Microsoft erstellt werden. Der Bitwert des MobiLink-Servers muss mit dem Bitwert des DSN übereinstimmen. Genauer gesagt muss ein 64-Bit-MobiLink-Server einen 64-Bit-DSN verwenden, der über ODBC-Datenquellenadministrator (64 Bit) erstellt wurde. Siehe „[Lektion 1: Einrichten einer konsolidierten MobiLink-Datenbank](#)“ [[MobiLink - Erste Orientierung](#)].

## Beispiel

Mit dem folgenden Befehl wird der MobiLink-Server gestartet. Dabei dient die ODBC-Datenquelle *SQL Anywhere 16 CustDB* zur Identifizierung der konsolidierten Datenbank. Geben Sie den folgenden Befehl auf einer Zeile ein.

```
mlsrv16 -c "DSN=SQL Anywhere 16 CustDB;UID=ml_server;PWD=sql" -zs MyServer -o mlsrv.log -vcr -x tcpip(port=3303)
```

In diesem Beispiel liefert die Option `-c` eine Verbindungszeichenfolge, die einen ODBC-Datenquellennamen (DSN) und die Authentifizierung enthält. Die Option `-zs` liefert einen Servernamen. Die Option `-o` legt fest, dass die Logdatei *mlsrv.log* heißen soll. Der Inhalt von *mlsrv.log* wird aufgrund der Option `-vcr` ausführlich dargestellt. Die Option `-x` öffnet einen Port für Clients der Version 10 und höher.

Weitere Hinweise zu den im vorherigen Beispiel beschriebenen Optionen finden Sie unter:

- „mlsrv16-Option `-c`“ auf Seite 53
- „mlsrv16-Option `-zs`“ auf Seite 103
- „mlsrv16-Option `-o`“ auf Seite 67
- „mlsrv16-Option `-v`“ auf Seite 86
- „mlsrv16-Option `-x`“ auf Seite 93

Sie können den MobiLink-Server auch als Windows-Dienst oder Unix-Daemon ausführen. Siehe „Verwendung des MobiLink-Servers außerhalb der aktuellen Sitzung“ auf Seite 34.

## Erforderliche Privilegien

Sie müssen einen Datenbankbenutzer angeben, damit der MobiLink-Server eine Verbindung mit dem Datenbankserver herstellt. Sie können den Datenbankbenutzer mit der *mlsrv16*-Option `-c` oder in der ODBC-Datenquelle angeben. Siehe „mlsrv16-Option `-c`“ auf Seite 53.

Dieser Datenbankbenutzer muss vollständige SELECT-, INSERT-, UPDATE- und DELETE-Privilegien für die MobiLink-Systemtabellen haben und benötigt außerdem das EXECUTE ANY PROCEDURE-Privileg für die MobiLink-Systemprozeduren. Standardmäßig hat der Datenbankbenutzer, der das MobiLink-Setupskript ausführt, diese Privilegien. Wenn Sie beim Ausführen des MobiLink-Servers einen anderen Datenbankbenutzer verwenden möchten, müssen Sie diesem Benutzer diese Privilegien für die *ml\_\**-Tabellen und die *ml\_add\_\**-script-Systemprozeduren erteilen.

Eine Liste aller MobiLink-Systemprozeduren finden Sie unter „Systemprozeduren des MobiLink-Servers“ auf Seite 648.

Der Datenbankbenutzer benötigt außerdem das entsprechende Privileg für alle in den MobiLink-Skripten referenzierten Tabellen sowie EXECUTE-Privilegien für alle in den MobiLink-Skripten referenzierten Prozeduren.

Einige Typen von konsolidierten MobiLink-Datenbanken verlangen, dass der vom MobiLink-Server verwendete Datenbankbenutzer bestimmte Privilegien für Systemtabellen bzw. Ansichten hat. Hinweise zu bestimmten konsolidierten Datenbanken finden Sie in den folgenden Themen:

- „Konsolidierte Adaptive Server Enterprise-Datenbank“
- „Konsolidierte IBM DB2 LUW-Datenbank“
- „Konsolidierte Microsoft SQL Server-Datenbank“
- „Konsolidierte MySQL-Datenbank“
- „Konsolidierte Oracle-Datenbank“

Weitere Hinweise zu Setupskripten finden Sie unter „[Einrichten einer konsolidierten Datenbank](#)“ auf Seite 3.

### MobiLink-Konnektivität

Wenn Sie HTTP oder HTTPS verwenden, mit oder ohne Relay Server, können Sie mithilfe eines Webbrowsers überprüfen, ob der MobiLink-Server auf Anforderungen wartet. Wenn beispielsweise Ihre MobiLink-Server-Befehlszeile wie folgt lautet:

```
mlsrv16 ... -x http(port=8080)
```

und der Computer *ml1.mycorp.com* heißt, können Sie einen Webbrowser öffnen und darin auf **http://ml1.mycorp.com:8080** zeigen.

Der MobiLink-Server antwortet mit einem Fehler (**404 Not Found**), der auch die Hauptversion des MobiLink-Servers enthält.

Siehe „[Einführung in den Relay Server](#)“ [*Relay Server*].

## MobiLink-Server herunterfahren

Der MobiLink-Server kann von dem Computer aus gestoppt werden, von dem aus der Server gestartet wurde. Sie können den MobiLink-Server auf folgende Weisen stoppen:

- Verwenden Sie das Dienstprogramm MobiLink-Serverstopp (mlstop).
- Klicken Sie im Meldungsfenster des MobiLink-Servers auf **Herunterfahren**.
- Klicken Sie unter Windows mit der rechten Maustaste auf das Symbol des MobiLink-Servers in der Taskleiste und klicken Sie auf **Herunterfahren**.
- Beim Ausführen unter Unix ohne das Meldungsfenster des MobiLink-Servers geben Sie den Befehl Q ein.
- Verwenden Sie die Shutdown-Methode (Herunterfahren) in der API von MobiLink-Server.

**Siehe auch**

- [„Dienstprogramm zum Stoppen von MobiLink \(mlstop\)“ auf Seite 698](#)
- [ServerContext.shutdown-Methode \[MobiLink-Server Java\] auf Seite 514](#)
- [ServerContext.Shutdown-Methode \[MobiLink-Server .NET\] auf Seite 607](#)

## Protokollierung für den MobiLink-Server

Die Protokollierung der Serveraktionen ist vor allem bei der Entwicklung und der Fehlerbehandlung hilfreich. Für die normale Verwendung in einer Produktionsumgebung wird die ausführliche Ausgabe nicht empfohlen, da sie die Performance beeinträchtigen kann.

**Protokoll in eine Datei schreiben**

Bestimmte Ausgaben der Protokollierung werden im Meldungsfenster des MobiLink-Servers angezeigt. Mithilfe der Option -o können Sie die Ausgabe zusätzlich in eine Meldungslogdatei schreiben. Der folgende Befehl schreibt die Ausgabe in eine Meldungslogdatei mit dem Namen *mlsrv.log*.

```
mlsrv16 -o mlsrv.log -c ...
```

Sie können die Größe der Logdateien steuern und mit den folgenden Optionen festlegen, was geschehen soll, wenn eine Datei die Maximalgröße erreicht:

- Verwenden Sie die Option -o, um festzulegen, dass eine Logdatei verwendet werden soll.
- Verwenden Sie die Option -ot, um festzulegen, dass eine Logdatei verwendet werden soll und der vorherige Inhalt der Datei gelöscht werden soll, bevor Meldungen in sie geschrieben werden.
- Zusätzlich zur Option -o oder -ot können Sie die Option -on verwenden, um die Größe anzugeben, bei der die Logdatei mit der Erweiterung *.old* umbenannt und eine neue Datei mit dem ursprünglichen Namen gestartet wird. Diese Option beschränkt den Gesamtspeicherplatz auf der Festplatte, der von den Meldungslogdateien in Anspruch genommen werden darf.
- Zusätzlich zur Option -o oder -ot können Sie Option -os verwenden, um die Größe festzulegen, bei der einer alten Logdatei ein neuer Name zugewiesen wird, der auf dem Datum und einer fortlaufenden Nummer basiert.

Siehe:

- [„mlsrv16-Option -o“ auf Seite 67](#)
- [„mlsrv16-Option -on “ auf Seite 68](#)
- [„mlsrv16-Option -os“ auf Seite 68](#)
- [„mlsrv16-Option -ot “ auf Seite 69](#)

**Umfang der Protokollausgabe steuern**

Mit der Option -v können Sie steuern, welche Informationen in der Meldungslogdatei protokolliert und im Meldungsfenster des MobiLink-Servers angezeigt werden. Siehe [„mlsrv16-Option -v “ auf Seite 86](#).

**Protokollierung von Warnungen steuern**

Sie können steuern, welche Warnungen angezeigt werden.

Weitere Hinweise finden Sie unter:

- „mlsrv16-Option -zw“ auf Seite 105
- „mlsrv16-Option -zwd“ auf Seite 106
- „mlsrv16-Option -zwe“ auf Seite 106

## MobiLink-Serverlogs anzeigen

Es bestehen folgende Möglichkeiten zum Anzeigen der MobiLink-Logs:

- Im MobiLink-Server-Meldungsfenster
- Durch Öffnen der Logdatei
- Mithilfe der **Logdatei-Anzeige für MobiLink-Server** in Sybase Central

Um Logdaten außerhalb des Meldungsfensters des MobiLink-Servers anzuzeigen, müssen Sie die Daten in einer Datei protokollieren. Siehe [Protokoll in eine Datei schreiben auf Seite 33](#).

### Anzeige von MobiLink-Server-Logdateien

Um MobiLink-Serverlogs anzuzeigen, öffnen Sie Sybase Central und klicken Sie auf **Extras » MobiLink 16 » Anzeige von MobiLink-Server-Logdateien**. Sie werden aufgefordert, die anzuzeigende Logdatei zu wählen.

Die **Logdatei-Anzeige für MobiLink-Server** liest die Daten aus den MobiLink-Logdateien. Die Anzeige stellt weder eine Verbindung zum MobiLink-Server her noch ändert sie die Zusammenstellung der Logdateien.

Mit der **Logdatei-Anzeige für MobiLink-Server** können Sie die angezeigten Daten filtern. Außerdem liefert die Anzeige Statistikwerte basierend auf den Daten im Log.

## Verwendung des MobiLink-Servers außerhalb der aktuellen Sitzung

Sie können den MobiLink-Server so einrichten, dass er jederzeit verfügbar ist. Dazu betreiben Sie den MobiLink-Server für Windows und Unix so, dass er weiter läuft, wenn Sie die Sitzung beenden. Wie Sie dies durchführen, hängt von Ihrem Betriebssystem ab.

- **Unix-Daemon** Sie können den MobiLink-Server als Daemon ausführen, indem Sie die mlsrv16-Option -ud verwenden. Damit kann der MobiLink-Server im Hintergrund ausgeführt werden und weiterlaufen, nachdem Sie sich abgemeldet haben.
- **Windows Dienst** Sie können den MobiLink-Server unter Windows als Dienst ausführen.

Um einen MobiLink-Server anzuhalten, der als Service läuft, können Sie "mlstop", "dbsvc" oder den Dienste-Manager von Windows verwenden.

**Siehe auch**

- „mlsrv16-Option -ud“ auf Seite 85
- „MobiLink-Server als Dienst unter Windows“ auf Seite 35
- „Dienstprogramm zum Stoppen von MobiLink (mlstop)“ auf Seite 698
- „Dienstprogramm für Dienste (dbsvc) für Linux“ [*SQL Anywhere Server - Datenbankadministration*]
- „Datenbankserver als Dienst oder Daemon ausführen“ [*SQL Anywhere Server - Datenbankadministration*]

## MobiLink-Server unter Unix als Daemon betreiben

Um den MobiLink-Server unter Unix im Hintergrund und damit unabhängig von der laufenden Sitzung zu betreiben, können Sie ihn als Daemon ausführen.

**Voraussetzungen**

Es gibt keine Voraussetzungen für diese Aufgabe.

**Aufgabe**

- Verwenden Sie die Option -ud, wenn Sie den MobiLink-Server starten. Zum Beispiel:

```
mlsrv16 -c "DSN=SQL Anywhere 16 Demo;UID=DBA;PWD=sql" -ud
```

**Ergebnisse**

Der MobiLink-Server wird unter Unix als Daemon ausgeführt.

**Siehe auch**

- „mlsrv16-Option -ud“ auf Seite 85
- „Dienstprogramm für Dienste (dbsvc) für Linux“ [*SQL Anywhere Server - Datenbankadministration*]

## MobiLink-Server als Dienst unter Windows

Um den MobiLink-Server unter Windows im Hintergrund und damit unabhängig von der laufenden Sitzung zu betreiben, können Sie ihn als Dienst ausführen.

Von der Befehlszeile oder über die Registerkarte **Dienste** in Sybase Central können Sie folgende Dienstverwaltungsaufgaben ausführen:

- Hinzufügen, Bearbeiten und Entfernen von Diensten.
- Dienste starten und stoppen.
- Parameter ändern, die einen Dienst steuern.

**Siehe auch**

- „Dienstprogramm für Dienste (dbsvc) für Windows“ [*SQL Anywhere Server - Datenbankadministration*]

## Mit Diensten arbeiten

Verwenden Sie Sybase Central, um einen neuen Dienst hinzufügen bzw. einen vorhandenen Dienst zu ändern oder zu löschen. Änderungen einer Dienstekonfiguration werden wirksam, wenn der Dienst das nächste Mal gestartet wird.

### Voraussetzungen

Es gibt keine Voraussetzungen für das Ausführen dieser Aufgabe.

### Kontext und Bemerkungen

Die Symbole für die Dienste in Sybase Central zeigen den aktuellen Status der Dienste an, wobei ein Symbol verwendet wird, das anzeigt, ob der Dienst ausgeführt wird oder gestoppt ist.

Sie können den Dienst auch mit dem Dienstprogramm dbsvc erstellen. Siehe „[Dienstprogramm für Dienste \(dbsvc\) für Windows](#)“ [*SQL Anywhere Server - Datenbankadministration*].

### Aufgabe

1. Klicken Sie in Sybase Central im linken Fensterausschnitt auf "MobiLink 16".
2. Im rechten Fensterausschnitt klicken Sie auf die Registerkarte **Dienste**.
3. Rechtsklicken Sie im rechten Fensterausschnitt und klicken Sie auf **Neu » Dienst**.

Wenn Sie einen Dienst löschen möchten, wählen Sie den Dienst und klicken Sie auf **Bearbeiten » Löschen**.

Wenn Sie die Parameter für einen Dienst ändern möchten, rechtsklicken Sie auf den Dienst und klicken Sie auf **Eigenschaften**.

4. Befolgen Sie die Anweisungen des **Assistenten zum Erstellen eines neuen Dienstes**.

### Ergebnisse

Der Dienst wird hinzugefügt, gelöscht bzw. geändert.

## Startoptionen für Dienste

Die folgenden Optionen steuern das Verhalten der MobiLink-Dienste. Sie können diese Optionen auf der Registerkarte **Allgemein** des Fensters **Diensteigenschaften** einstellen.

- **Automatisch** Wenn Sie **Automatisch** auswählen, startet der Dienst, sobald das Windows-Betriebssystem hochgefahren ist. Diese Einstellung ist für Datenbankserver und andere Anwendungen geeignet, die immer laufen sollen.
- **Manuell** Wenn Sie **Manuell** auswählen, startet der Dienst nur, wenn ihn ein Benutzer mit Administratorberechtigung startet. Hinweise zu den Administratorberechtigungen entnehmen Sie Ihrer Windows-Dokumentation.

- **Deaktiviert** Wenn Sie **Deaktiviert** auswählen, wird der Dienst nicht gestartet.

Die Startart wird beim nächsten Start von Windows übernommen.

## Befehlszeilenoptionen

Die Registerkarte **Konfiguration** des Fensters **Eigenschaften des Dienstes** enthält das Textfeld **Dateiname**, in das Sie den Programmpfad eingeben können, und das Textfeld **Parameter** für die Befehlszeilenoptionen für einen Dienst. Geben Sie in das Feld **Parameter** nicht den Namen der Programmdatei ein.

Geben Sie beispielsweise zum Starten eines MobiLink-Synchronisationsdienstes mit ausführlicher Protokollierung Folgendes in das Feld **Parameter** ein:

```
-c "DSN=SQL Anywhere 16 Demo;UID=DBA;PWD=sql"  
-vc
```

Die Befehlszeilenoptionen für einen Dienst sind identisch mit denen für die entsprechende Programmdatei. Siehe „[MobiLink-Serveroptionen](#)“ auf Seite 44.

## Kontooptionen

Sie können auswählen, unter welchem Konto der Dienst laufen soll. Die meisten Dienste laufen unter dem lokalen Systemkonto (LocalSystem), was auch die Standardeinstellung für Dienste ist. Sie können den Dienst so einrichten, dass er unter einem anderen Konto startet, indem Sie die Registerkarte **Konto** des Fensters **Eigenschaften des Dienstes** öffnen und die gewünschten Kontoinformationen eingeben.

Wenn der Dienst unter einem anderen Konto als LocalSystem laufen soll, muss dieses Konto das Privileg "Als Dienst anmelden" haben. Weitere Hinweise zu erweiterten Privilegien finden Sie in der Microsoft Windows-Dokumentation.

Ob ein Symbol für den Dienst in der Taskleiste oder auf dem Desktop angezeigt wird, hängt vom ausgewählten Konto ab und davon, ob die Option **Interaktive Beziehung mit Desktop erlauben** aktiviert ist:

- Wenn ein Dienst unter "LocalSystem" läuft und im Fenster **Diensteigenschaften** die Option **Interaktion mit dem Desktop zulassen** aktiviert wurde, erscheint ein Symbol auf dem Desktop jedes Benutzers, der sich auf dem Windows XP-/2003-Computer anmeldet, auf dem der Dienst läuft. Jeder Benutzer kann das Anwendungsfenster öffnen und das Programm stoppen, das als Dienst läuft.
- Wenn ein Dienst unter "LocalSystem" läuft und im Fenster **Diensteigenschaften** die Option **Interaktion mit dem Desktop zulassen** deaktiviert wurde, erscheint kein Symbol auf dem Desktop der Benutzer. Nur Benutzer mit der Berechtigung, den Status eines Dienstes zu ändern, können den Dienst stoppen.
- Wenn ein Dienst unter einem anderen Konto läuft, erscheint kein Symbol auf dem Desktop. Nur Benutzer mit der Berechtigung, den Status eines Dienstes zu ändern, können den Dienst stoppen.

## Programmdatei ändern

Sie können die Programmdatei ändern, die dem Dienst in Sybase Central zugeordnet ist.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Kontext und Bemerkungen

Wenn Sie eine Programmdatei in ein neues Verzeichnis verschieben möchten, müssen Sie diesen Eintrag ändern.

### Aufgabe

1. Klicken Sie im Fenster **Eigenschaften des Dienstes** auf die Registerkarte **Konfiguration**.
2. Geben Sie im Feld **Dateiname** den neuen Pfad und Dateinamen ein.

### Ergebnisse

Die dem Dienst zugeordnete Programmdatei wird aktualisiert.

## Dienste starten und stoppen

### Dienste starten und stoppen

Wenn Sie einen Dienst starten, läuft er, bis Sie ihn wieder stoppen. Wenn Sie Sybase Central schließen oder sich aus dem System abmelden, wird der Dienst nicht gestoppt.

### Voraussetzungen

Es gibt keine Voraussetzungen für das Ausführen dieser Aufgabe.

### Kontext und Bemerkungen

Das Stoppen eines Diensts schließt alle Netzwerkverbindungen und stoppt den MobiLink-Server. Bei anderen Anwendungen wird das Programm heruntergefahren.

### Starten oder Stoppen eines Dienstes

1. Klicken Sie in Sybase Central im linken Fensterausschnitt auf "MobiLink 16" und öffnen Sie anschließend im rechten Fensterausschnitt die Registerkarte **Dienste**.
2. Rechtsklicken Sie auf den Dienst und klicken Sie auf **Start** oder **Stopp**.

### Ergebnisse

Der Dienst wird gestartet bzw. gestoppt.

## Mehrere Dienste

Obwohl Sie den Dienste-Manager von Windows in der **Systemsteuerung** für einige Aufgaben verwenden können, ist es nicht möglich, einen MobiLink-Dienst über den Dienste-Manager von Windows zu installieren oder zu konfigurieren. Sie können Sybase Central für die Dienstverwaltung für MobiLink benutzen.

Wenn Sie den Windows-Dienste-Manager (aus der Systemsteuerung) aufrufen, wird eine Liste von Diensten angezeigt. Die Namen der SQL Anywhere-Dienste werden aus dem Präfix "SQL Anywhere" und dem Dienstenamen gebildet, den Sie beim Einrichten des Dienstes vergeben haben. Alle installierten Dienste erscheinen zusammen in der Liste.

## Abhängigkeiten von Diensten

In manchen Fällen ist es nützlich, mehr als eine Programmdatei als Dienst auszuführen, wobei diese Programmdateien möglicherweise voneinander abhängen. So müssen beispielsweise für die Synchronisierung der MobiLink-Server und der Server der konsolidierten Datenbank gleichzeitig laufen.

Dienste müssen in der richtigen Reihenfolge gestartet werden. Wenn ein MobiLink-Synchronisationsdienst gestartet wird, bevor der Server der konsolidierten Datenbank gestartet wurde, schlägt MobiLink fehl, weil es den Server der konsolidierten Datenbank nicht finden kann. Der Datenbankserver muss bereits laufen, wenn Sie den MobiLink-Server starten. (Dies gilt nicht, wenn der konsolidierte Datenbankserver sich auf einem anderen Computer befindet.)

Sie können dieses Problem durch die Einrichtung von Dienstgruppen vermeiden, die aus Sybase Central verwaltet werden. Siehe [„Dienste oder Gruppen zu einer Liste von Abhängigkeiten hinzufügen“ auf Seite 40](#).

## Dienstgruppen

Sie können jeden Dienst auf Ihrem System als Mitglied in eine Dienstgruppe aufnehmen. Standardmäßig gehört jeder Dienst zu einer Gruppe. Die Standardgruppe für den MobiLink-Server ist SQLANYMobiLink. Siehe [„Gruppen für Dienste überprüfen und ändern“ auf Seite 39](#).

## Gruppen für Dienste überprüfen und ändern

Bevor Sie Ihre Dienste so konfigurieren können, dass sie in der richtigen Reihenfolge starten, müssen Sie darauf achten, dass die Dienste zur richtigen Gruppe gehören. Dies können Sie in Sybase Central prüfen und gegebenenfalls ändern.

## Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

## Aufgabe

1. Klicken Sie in Sybase Central im linken Fensterausschnitt auf "MobiLink 16" und öffnen Sie anschließend im rechten Fensterausschnitt die Registerkarte **Dienste**.
2. Rechtsklicken Sie auf den Dienst und klicken Sie auf **Eigenschaften**.

3. Klicken Sie auf die Registerkarte **Abhängigkeiten**. Im oberen Textfeld wird der Name der Gruppe angezeigt, zu der der Dienst gehört.
4. Klicken Sie auf **Ändern**, um eine Liste der verfügbaren Gruppen auf Ihrem System anzuzeigen.
5. Wählen Sie eine der Gruppen oder geben Sie den Namen einer neuen Gruppe ein.
6. Klicken Sie auf **OK**, um den Dienst dieser Gruppe zuzuordnen.

## Ergebnisse

Ergebnis

## Nächste Schritte

Sie können Ihre Dienste so konfigurieren, dass sie in jedem Fall in der richtigen Reihenfolge gestartet werden. Siehe [„Dienste oder Gruppen zu einer Liste von Abhängigkeiten hinzufügen“](#) auf Seite 40.

## Dienste oder Gruppen zu einer Liste von Abhängigkeiten hinzufügen

Mit Sybase Central können Sie Abhängigkeiten für einen Dienst festlegen. Sie können beispielsweise sicherstellen, dass mindestens eine Gruppe gestartet wurde, bevor der aktuelle Dienst gestartet wird.

## Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

## Aufgabe

1. Klicken Sie in Sybase Central im linken Fensterausschnitt auf "MobiLink 16" und öffnen Sie anschließend im rechten Fensterausschnitt die Registerkarte **Dienste**.
2. Rechtsklicken Sie auf den Dienst und klicken Sie auf **Eigenschaften**.
3. Klicken Sie auf die Registerkarte **Abhängigkeiten**.
4. Klicken Sie auf **Dienste hinzufügen** oder **Dienstgruppen hinzufügen**, um einen Dienst oder eine Gruppe der Liste der Abhängigkeiten hinzuzufügen.
5. Wählen Sie einen der Dienste oder Gruppen aus der Liste.
6. Klicken Sie auf **OK**, um den Dienst oder die Gruppe der Liste der Abhängigkeiten hinzuzufügen.

## Ergebnisse

Der Dienst bzw. die Gruppe wird zu der Liste von Abhängigkeiten hinzugefügt.

## MobiLink-Server in einer Serverfarm

### Hinweis

**Erforderliche getrennt lizenzierbare Komponenten** Die Ausführung des MobiLink-Servers in einer Serverfarm ist eine Funktion der Hochverfügbarkeitsoption von MobiLink, für die eine separate Lizenz erforderlich ist. Siehe „[Getrennt lizenzierbare Komponenten](#)“ [*SQL Anywhere 16 - Einführung*].

Eine MobiLink-Serverfarm ist eine Umgebung, in der mehrere MobiLink-Server dieselbe Gruppe von entfernten Datenbanken mit derselben konsolidierten Datenbank synchronisieren. Dies ist häufig für große Deployments oder die Ausfallsicherung erforderlich. Diese Serverfarm-Deployments von MobiLink erfordern möglicherweise die Verwendung des Relay Servers, wenn eine HTTP-Kommunikationsverbindung verwendet wird. Für TCP-basierte Datenströme kann ein TCP-Lastverteiler eingesetzt werden. Bei Verwendung mehrerer Servern funktioniert kein neu startbarer Download.

Wenn Sie den Notifier mit serverinitiiertem Synchronisation ausführen, legen Sie mit der Option `-lsc` die Einstellungen der lokalen Serververbindung fest. Diese Einstellungen werden an die anderen Server in der Farm übergeben, sodass sie sich miteinander verbinden können, um Benachrichtigungen gemeinsam zu verarbeiten. Beispiel: Die Anwendung wird auf Host **farm\_host22** ausgeführt:

```
mlsrv16 -x tcpip(port=3245) -zs server5 -lsc
tcpip(host=farm_host22;port=3245) -c ...
```

### MobiLink-Arbiterserver

Wenn Sie eine MobiLink-Serverfarm mit serverinitiiertem Synchronisation ausführen, stellen Sie mithilfe des MobiLink-Arbiterservers sicher, dass es immer einen Primärserver in der Farm gibt. Wenn Sie jederzeit einen Primärserver haben, verhindert dies, dass redundante Benachrichtigungen gesendet werden und Meldungen verloren gehen.

Verwenden Sie den Befehl **mlarbiter** zum Starten des MobiLink-Arbiters und verwenden Sie die MobiLink-Serveroption `-ca` zusammen mit der Option `-lsc` zum Starten des MobiLink-Servers mit den Arbitrereinformationen.

### Siehe auch

- „MobiLink-Arbiterserver-Dienstprogramm für Windows (mlarbiter)“ auf Seite 711
- „mlsrv16-Option `-lsc`“ auf Seite 65
- „mlsrv16-Option `-ca`“ auf Seite 53

## Speichernutzung im MobiLink-Server

Der MobiLink-Server verwendet den verfügbaren Speicher in erster Linie auf folgende Weise:

- Caching
- für VMs
- Speichern von Systemdaten

## Cache

In einem typischen MobiLink-Server wird der meisten Speicherplatz vom Cache verwendet, der Zeilendaten sowie die dazugehörigen Datenstrukturen und Netzwerkpuffer speichert. Im Allgemeinen werden im Cache Daten gespeichert, die entweder in der Größe oder in der Menge unbegrenzt sind. Wenn die Menge der im Cache zu speichernden Daten die Größe des Cachespeichers überschreitet, wird der Überschuss auf die Festplatte übertragen, was einen potenziellen Engpass in der Performance der Synchronisation bedeutet. Das Ausmaß, in dem die Auslagerung auf die Festplatte ein Problem darstellt, ist abhängig vom Umfang und der Häufigkeit des Festplatten-I/O und von der Geschwindigkeit der Festplatte. Um diesen potenziellen Engpass zu beseitigen, wird empfohlen, möglichst keine Daten auf die Festplatte auszulagern. Achten Sie auf die Warnung 10082 im MobiLink-Serverlog oder den Alarm "Cache ist voll" im SQL Anywhere-Monitor.

Standardmäßig vergrößert der MobiLink-Server seinen Cache automatisch auf bis zu 60% des verfügbaren Prozess-Adressraumes und verkleinert seinen Cache, wenn andere Prozesse im System mehr Speicher benötigen oder der Nicht-Cache-Speicher des Servers größer wird. Die anfängliche maximale und minimale Cachegröße kann mit den mlsrv16-Serveroptionen -cinit, -cmax und -cmin gesteuert werden. Falls gewünscht, kann der Server die dynamische Cachedimensionierung deaktivieren, indem er für die Optionswerte -cmax und -cmin denselben Wert angibt.

Siehe:

- „mlsrv16-Option -cmax“ auf Seite 55
- „mlsrv16-Option -cmin“ auf Seite 56
- „mlsrv16-Option -cinit“ auf Seite 54

Es gibt mehrere Möglichkeiten, Informationen über den Cache herauszufinden:

- **Server-Cachegröße** Dies ist eine Metrik des SQL Anywhere-Monitors, mit der die aktuelle Cachegröße angegeben wird. Siehe [Liste von Metriken für MobiLink-Serverressourcen \[SQL Anywhere Server - Datenbankadministration\]](#).
- **Prozent gesperrter Seiten** Dies ist eine Metrik des SQL Anywhere-Monitors, die angibt, welcher prozentuale Anteil der Seiten im Cache nicht auf die Festplatte übertragen werden kann. Siehe [Liste von Metriken für MobiLink-Serverressourcen \[SQL Anywhere Server - Datenbankadministration\]](#).
- **Prozent verwendeter Seiten** Dies ist eine Metrik des SQL Anywhere-Monitors, die den prozentualen Anteil der Seiten angibt, die gültige Daten enthalten. Siehe [Liste von Metriken für MobiLink-Serverressourcen \[SQL Anywhere Server - Datenbankadministration\]](#).
- **PAGES\_LOCKED\_MAX** Die Option -ppv für mlsrv16 kann die Anzahl der Seiten im Cachespeicher ausgeben. Siehe „mlsrv16-Option -ppv“ auf Seite 70.
- **PAGES\_LOCKED** Die Option -ppv für mlsrv16 kann die Anzahl der Cacheseiten ausgeben, die in den Speicher geladen wurden. Siehe „mlsrv16-Option -ppv“ auf Seite 70.
- **PAGES\_USED** Die Option -ppv für mlsrv16 kann die Anzahl der verwendeten Cacheseiten ausgeben. Siehe „mlsrv16-Option -ppv“ auf Seite 70.
- **Option -vk für mlsrv16** Bewirkt, dass der Server eine Zeile in das Log schreibt, wenn der Cache vergrößert oder verkleinert wird. Siehe „mlsrv16-Option -ppv“ auf Seite 70.

## VMs

Ein weiterer Bereich innerhalb des MobiLink-Servers, der viel Speicherplatz in Anspruch nimmt, sind die VMs für Embedded Java und .NET. In Deployments mit intensiver Nutzung der API für die direkte Zeilenbehandlung oder mit eingebetteten Anwendungsservern wie JBoss oder Tomcat, kann der von den VMs belegte Speicherplatz die Größe des Cachespeichers übersteigen.

Wenn Sie eine Java VM verwenden, haben Sie normalerweise eine gewisse Kontrolle über den von dieser belegten Speicherplatz. Die meisten Java VMs stellen die Optionen -Xms und -Xmx bereit, mit deren Hilfe Sie die maximale bzw. anfängliche VM-Heapgröße festlegen können.

Um den von den eingebetteten VMs belegten Speicherplatz zu ermitteln, können Sie entweder die Metrik **Speichernutzung der VM** des SQL Anywhere-Monitors oder den Wert **VM\_MEM\_USE** für die Option -ppv verwenden.

Siehe:

- [Liste von Metriken für MobiLink-Serverressourcen \[SQL Anywhere Server - Datenbankadministration\]](#)
- „mlsrv16-Option -ppv“ auf Seite 70

## Sonstige Speichernutzung

Der verbleibende Speicherplatz dient zum Speichern von Statusinformationen über jede Synchronisation, Thread-Stacks, Thread-übergreifender Kommunikation und ODBC-Treiber. In der Regel macht dies ein paar Dutzend Megabyte Speicherplatz aus und dieser wird durch die Optionen -nc, -sm und -w für mlsrv16 begrenzt. Es ist erwähnenswert, dass nicht auf SQL Anywhere basierende ODBC-Treiber sehr viel Speicherplatz belegen können, vor allem bei der Verarbeitung von BLOB-Spalten.

Siehe:

- „mlsrv16-Option -nc“ auf Seite 65
- „mlsrv16-Option -sm“ auf Seite 81
- „mlsrv16-Option -w“ auf Seite 90

## Fehlerbehandlung beim MobiLink-Serverstart

In diesem Abschnitt werden einige bekannte Probleme beschrieben, die beim Starten des MobiLink-Servers auftreten können.

## Achten Sie darauf, dass die Kommunikationssoftware läuft

Geeignete Netzwerk-Kommunikationssoftware muss installiert und ausgeführt werden, bevor Sie den MobiLink-Server starten. Wenn Sie zuverlässige Netzwerksoftware mit nur einem installierten Netzwerk betreiben, sollten dabei keine Probleme auftreten. Vergewissern Sie sich, ob andere Software, die Netzwerkkommunikation benötigt, einwandfrei arbeitet, bevor Sie den MobiLink-Server aufrufen.

Es kann sinnvoll sein, zu überprüfen, ob ping und telnet ordnungsgemäß funktionieren. Die Anwendungen ping und telnet werden mit den meisten TCP/IP-Protokoll-Stacks mitgeliefert.

## Fehlersuche bei Startproblemen mit der Netzwerkkommunikation

Wenn Sie bei der Einrichtung einer Verbindung über ein Netzwerk auf Probleme stoßen, können Sie Fehlersuchoptionen auf Client und Server verwenden, um die Probleme zu diagnostizieren. Die Startinformationen erscheinen im Meldungsfenster des MobiLink-Servers. Sie können die Option `-o` verwenden, wenn die Ergebnisse in einer Datei protokolliert werden sollen.

Siehe „[Protokollierung für den MobiLink-Server](#)“ auf Seite 33.

## MobiLink-Serverkonnektivität überprüfen

Wenn Sie HTTP oder HTTPS verwenden, mit oder ohne Relay Server, können Sie mithilfe eines Webbrowsers überprüfen, ob der MobiLink-Server auf Anforderungen wartet. Wenn beispielsweise Ihre MobiLink-Server-Befehlszeile wie folgt lautet:

```
mlsrv16 ... -x http(port=8080)
```

und der Computer `ml1.mycorp.com` heißt, können Sie einen Webbrowser öffnen und darin auf **http://ml1.mycorp.com:8080** zeigen.

Der MobiLink-Server antwortet mit einem Fehler (**404 Not Found**), der auch die Hauptversion des MobiLink-Servers enthält.

Siehe „[Einführung in den Relay Server](#)“ [*Relay Server*].

## MobiLink-Serveroptionen

### mlsrv16-Syntax

Startet einen MobiLink-Server.

#### Syntax

**mlsrv16 -c "connection-string" [ options ]**

| Option             | Beschreibung                                                                                                                                                                  |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>@data</code> | Liest Optionen aus der angegebenen Umgebungsvariablen oder Konfigurationsdatei ein. Siehe „ <a href="#">mlsrv16-Option @data</a> “ auf Seite 50.                              |
| <code>-a</code>    | Behält die Verwendung zur konsolidierten Datenbank nach einem Synchronisationsfehler während dieser Verbindung bei. Siehe „ <a href="#">mlsrv16-Option -a</a> “ auf Seite 51. |
| <code>-b</code>    | Füllt Zeichenfolgen mit Leerzeichen auf. Siehe „ <a href="#">mlsrv16-Option -b</a> “ auf Seite 51.                                                                            |

| Option                                   | Beschreibung                                                                                                                                                                      |
|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-bn</b> <i>size</i>                   | Legt die maximale Anzahl von Byte fest, die beim Vergleich von BLOBs zur Erkennung von Konflikten berücksichtigt wird. Siehe „ <a href="#">mlsrv16-Option -bn</a> “ auf Seite 52. |
| <b>-c</b> " <i>keyword=value; ...</i> "  | Ermöglicht die Eingabe der DBC-Verbindungsparameter für Ihre konsolidierte Datenbank. Diese Option ist erforderlich. Siehe „ <a href="#">mlsrv16-Option -c</a> “ auf Seite 53.    |
| <b>-ca</b> <i>host_or_ip</i>             | Legt den Hostnamen oder die IP-Adresse für den MobiLink-Arbiterserver fest. Siehe „ <a href="#">mlsrv16-Option -ca</a> “ auf Seite 53.                                            |
| <b>-cinit</b> <i>size</i>                | Legt die Ausgangsgröße des Servercache fest. Siehe „ <a href="#">mlsrv16-Option -cinit</a> “ auf Seite 54.                                                                        |
| <b>-cm</b> <i>size</i>                   | Legt die Größe des Servercache fest. Siehe „ <a href="#">mlsrv16-Option -cm</a> “ auf Seite 55.                                                                                   |
| <b>-cmax</b> <i>size</i>                 | Legt die maximale Größe für den Hauptspeichercache des Servers fest. Siehe „ <a href="#">mlsrv16-Option -cmax</a> “ auf Seite 55.                                                 |
| <b>-cmin</b> <i>size</i>                 | Legt die minimale Größe für den Hauptspeichercache des Servers fest. Siehe „ <a href="#">mlsrv16-Option -cmin</a> “ auf Seite 56.                                                 |
| <b>-cn</b> <i>connections</i>            | Legt die maximale Anzahl der gleichzeitigen Verbindungen mit dem konsolidierten Datenbankserver fest. Siehe „ <a href="#">mlsrv16-Option -cn</a> “ auf Seite 56.                  |
| <b>-cr</b> <i>count</i>                  | Legt die maximale Anzahl der Versuche für Datenbankverbindungen fest. Siehe „ <a href="#">mlsrv16-Option -cr</a> “ auf Seite 57.                                                  |
| <b>-cs</b> " <i>keyword=value; ...</i> " | Liefert die Systemdatenbank-Verbindungsparameter für die MobiLink-Systemdatenbank (MLSD). Siehe „ <a href="#">mlsrv16-Option -cs</a> “ auf Seite 57.                              |
| <b>-ct</b> <i>Verbindungs-Timeout</i>    | Legt eine Zeitdauer ohne Benutzung fest, nach der die Verbindung getrennt wird. Siehe „ <a href="#">mlsrv16-Option -ct</a> “ auf Seite 58.                                        |
| <b>-dl</b>                               | Zeigt alle Lognachrichten im Meldungsfenster des MobiLink-Servers an. Siehe „ <a href="#">mlsrv16-Option -dl</a> “ auf Seite 58.                                                  |

| Option                                        | Beschreibung                                                                                                                                                                                                      |
|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-dr</b>                                    | Nur für Adaptive Server Enterprise. Stellt sicher, dass an der Synchronisation beteiligte Tabellen das DataRow-Sperrschema nicht verwenden. Siehe „ <a href="#">mlsrv16-Option -dr</a> “ auf Seite 59.            |
| <b>-ds</b> <i>size</i>                        | Legt die maximale Menge an Daten fest, die zur Verwendung in allen neu startbaren Downloads gespeichert werden kann. Siehe „ <a href="#">mlsrv16-Option -ds</a> “ auf Seite 59.                                   |
| <b>-dsd</b>                                   | Deaktiviert die Snapshot-Isolation, die standardmäßige Download-Isolationsstufe für konsolidierte SQL Anywhere- und Microsoft SQL Server-Datenbanken. Siehe „ <a href="#">mlsrv16-Option -dsd</a> “ auf Seite 60. |
| <b>-dt</b>                                    | Ermittelt Transaktionen nur innerhalb der aktuellen Datenbank. Siehe „ <a href="#">mlsrv16-Option -dt</a> “ auf Seite 60.                                                                                         |
| <b>-e</b> <i>filename</i>                     | Speichert das Fehlerlog einer entfernten Datenbank in der benannten Datei. Siehe „ <a href="#">mlsrv16-Option -e</a> “ auf Seite 61.                                                                              |
| <b>-esu</b>                                   | Verwendet die Snapshot-Isolation für Uploads. Siehe „ <a href="#">mlsrv16-Option -esu</a> “ auf Seite 62.                                                                                                         |
| <b>-et</b> <i>filename</i>                    | Speichert gesendete Fehlerlogs von entfernten Datenbanken in der benannten Datei, löscht aber zunächst den Inhalt der Datei, falls sie schon besteht. Siehe „ <a href="#">mlsrv16-Option -et</a> “ auf Seite 62.  |
| <b>-fips</b>                                  | Erzwingt FIPS-zertifizierte Verschlüsselung für alle sicheren MobiLink-Datenströme. Siehe „ <a href="#">mlsrv16-Option -fips</a> “ auf Seite 63.                                                                  |
| <b>-ftr</b> <i>path</i>                       | Gibt einen Speicherort für Dateien an, die vom MobiLink-Dienstprogramm für Dateiübertragungen (mlfiletransfer) heruntergeladen werden sollen. Siehe „ <a href="#">mlsrv16-Option -ftr</a> “ auf Seite 63.         |
| <b>-ftru</b>                                  | Gibt einen Speicherort für mit dem MobiLink-Dienstprogramm für Dateiübertragungen (mlfiletransfer) hochgeladene Dateien an. Siehe „ <a href="#">mlsrv16-Option -ftru</a> “ auf Seite 64.                          |
| <b>-lsc</b> <i>protocol[protocol-options]</i> | Diese Option legt die Informationen für die lokale Serververbindung fest. Siehe „ <a href="#">mlsrv16-Option -lsc</a> “ auf Seite 65.                                                                             |

| Option                        | Beschreibung                                                                                                                                                                          |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-nc</b> <i>connections</i> | Legt die maximale Anzahl gleichzeitiger Netzwerkverbindungen fest. Siehe „ <a href="#">mlsrv16-Option -nc</a> “ auf Seite 65.                                                         |
| <b>-notifier</b>              | Startet einen Notifier für die serverinitiierte Synchronisation. Siehe „ <a href="#">mlsrv16-Option -notifier</a> “ auf Seite 66.                                                     |
| <b>-o</b> <i>logfile</i>      | Protokolliert Nachrichten in einer Datei. Siehe „ <a href="#">mlsrv16-Option -o</a> “ auf Seite 67.                                                                                   |
| <b>-on</b> <i>size</i>        | Legt die maximale Größe für die Logdatei fest. Siehe „ <a href="#">mlsrv16-Option -on</a> “ auf Seite 68.                                                                             |
| <b>-oq</b>                    | Verhindert, dass das Einblendfenster bei einem Startfehler erscheint. Siehe „ <a href="#">mlsrv16-Option -oq</a> “ auf Seite 68.                                                      |
| <b>-os</b> <i>size</i>        | Maximale Größe der alten Logdateien. Siehe „ <a href="#">mlsrv16-Option -os</a> “ auf Seite 68.                                                                                       |
| <b>-ot</b> <i>logfile</i>     | Löscht zuerst den Inhalt der Datei und protokolliert anschließend Meldungen darin. Siehe „ <a href="#">mlsrv16-Option -ot</a> “ auf Seite 69.                                         |
| <b>-ppv</b> <i>period</i>     | Bewirkt, dass MobiLink neue periodische Überwachungswerte entsprechend der festgelegten Dauer ausgibt. Siehe „ <a href="#">mlsrv16-Option -ppv</a> “ auf Seite 70.                    |
| <b>-q</b>                     | Minimiert das Meldungsfenster des MobiLink-Servers. Siehe „ <a href="#">mlsrv16-Option -q</a> “ auf Seite 75.                                                                         |
| <b>-r</b> <i>retries</i>      | Legt die maximale Anzahl der erneuten Versuche bei Upload-Deadlocks fest. Siehe „ <a href="#">mlsrv16-Option -r</a> “ auf Seite 75.                                                   |
| <b>-rd</b> <i>delay</i>       | Legt die maximale Verzögerung in Sekunden fest, bevor bei Transaktions-Deadlocks ein neuer Versuch unternommen wird. Siehe „ <a href="#">mlsrv16-Option -rd</a> “ auf Seite 76.       |
| <b>-rp</b> <i>directory</i>   | Gibt das Verzeichnis an, in dem Synchronisationen für die Wiedergabe mit dem Dienstprogramm mlreplay aufgezeichnet werden. Siehe „ <a href="#">mlsrv16-Option -rp</a> “ auf Seite 77. |

| Option                                                            | Beschreibung                                                                                                                                                                                     |
|-------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-rrp</b> <i>directory</i>                                      | Die Option bewirkt, dass der MobiLink-Server das mlreplay-Dienstprogramm beim Serverstart im angegebenen Verzeichnis ausführt. Siehe „ <a href="#">mlsrv16-Option -rrp</a> “ auf Seite 76.       |
| <b>-s</b> <i>count</i>                                            | Gibt die maximale Anzahl der Zeilen an, die zugleich in die konsolidierte Datenbank hochgeladen oder von dort abgerufen werden können. Siehe „ <a href="#">mlsrv16-Option -s</a> “ auf Seite 77. |
| <b>-sl dnet</b> <i>script-options</i>                             | Legt die .NET CLR-Optionen (Common Language Runtime) fest und erzwingt, dass CLR beim Hochfahren geladen wird. Siehe „ <a href="#">mlsrv16-Option -sl dnet</a> “ auf Seite 78.                   |
| <b>-sl java</b> <i>script-options</i>                             | Legt die Java VM-Optionen fest und erzwingt das Laden der Java VM beim Starten. Siehe „ <a href="#">mlsrv16-Option -sl java</a> “ auf Seite 79.                                                  |
| <b>-sm</b> <i>number</i>                                          | Legt die maximale Anzahl von Synchronisationen fest, die zugleich bearbeitet werden können. Siehe „ <a href="#">mlsrv16-Option -sm</a> “ auf Seite 81.                                           |
| <b>-tc</b> <i>minutes</i>                                         | Legt den Schwellenwert für den Timeout bei der Ausführung von SQL-Skripten fest. Siehe „ <a href="#">mlsrv16-Option -tc</a> “ auf Seite 82.                                                      |
| <b>-tf</b>                                                        | Legt fest, dass die Ausführung des SQL-Skripts fehlschlägt, wenn der Schwellenwert für den Timeout erreicht wird (nicht für Oracle). Siehe „ <a href="#">mlsrv16-Option -tf</a> “ auf Seite 83.  |
| <b>-ts</b> <i>session-name(session-option=[option-value;...])</i> | Richtet eine Protokollierungssitzung für den MobiLink-Server ein. Siehe „ <a href="#">mlsrv16-Option -ts</a> “ auf Seite 84.                                                                     |
| <b>-tx</b> <i>count</i>                                           | Gruppieren Gruppen von Transaktionen für transaktionale Uploads und schreibt sie gemeinsam fest. Siehe „ <a href="#">mlsrv16-Option -tx</a> “ auf Seite 85.                                      |
| <b>-ud</b>                                                        | Legt auf UNIX-Plattformen die Ausführung als Daemon fest. Siehe „ <a href="#">mlsrv16-Option -ud</a> “ auf Seite 85.                                                                             |
| <b>-ui</b>                                                        | Startet den MobiLink-Server unter Linux mit X-Windows im Shell-Modus, wenn keine nutzbare Anzeige verfügbar ist. Siehe „ <a href="#">mlsrv16-Option -ui</a> “ auf Seite 86.                      |

| Option                           | Beschreibung                                                                                                                                                                                                                                                 |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-ux</b>                       | Öffnet unter Linux mit X-Windows das Meldungsfenster des MobiLink-Servers. Siehe „ <a href="#">mlsrv16-Option -ux</a> “ auf Seite 86.                                                                                                                        |
| <b>-v[levels]</b>                | Steuert den Typ der Meldungen, die in die Logdatei geschrieben werden. Siehe „ <a href="#">mlsrv16-Option -v</a> “ auf Seite 86.                                                                                                                             |
| <b>-w count</b>                  | Legt die anfängliche Anzahl von Datenbank-Worker-Threads fest. Siehe „ <a href="#">mlsrv16-Option -w</a> “ auf Seite 90.                                                                                                                                     |
| <b>-wm count</b>                 | Legt die maximal zulässige Anzahl von Datenbank-Worker-Threads fest. Siehe „ <a href="#">mlsrv16-Option -wm</a> “ auf Seite 91.                                                                                                                              |
| <b>-wn count</b>                 | Legen Sie die Anzahl von Netzwerk-Worker-Threads für die gleichzeitige Verarbeitung von Netzwerkdatenströmen fest. Siehe „ <a href="#">mlsrv16-Option -wn</a> “ auf Seite 82.                                                                                |
| <b>-wu count</b>                 | Legt die maximale Anzahl der Datenbank-Worker-Threads fest, die gleichzeitig Uploads durchführen dürfen. Siehe „ <a href="#">mlsrv16-Option -wu</a> “ auf Seite 92.                                                                                          |
| <b>-x protocol"options; ..."</b> | Legt das Kommunikationsprotokoll fest. Optional können Sie Netzwerkparameter in der Form <i>Parameter=Wert</i> festlegen, wobei mehrere Parameter durch Semikola getrennt werden. Siehe „ <a href="#">mlsrv16-Option -x</a> “ auf Seite 93.                  |
| <b>-zf</b>                       | Gibt an, dass der MobiLink-Server zu Beginn jeder Synchronisation eine Prüfung auf Skriptänderungen vornehmen soll. Siehe „ <a href="#">mlsrv16-Option -zf</a> “ auf Seite 102.                                                                              |
| <b>-zp</b>                       | Einige scheinbare Unterschiede zwischen den TIME-STAMP-Werten werden ignoriert, wenn die Werte in der entfernten und in der konsolidierten Datenbank unterschiedliche Gesamtstellenzahlen haben. Siehe „ <a href="#">mlsrv16-Option -zp</a> “ auf Seite 103. |
| <b>-zs name</b>                  | Legt einen Servernamen fest. Siehe „ <a href="#">mlsrv16-Option -zs</a> “ auf Seite 103.                                                                                                                                                                     |
| <b>-zt number</b>                | Gibt die maximale Anzahl von Prozessoren an, die für den MobiLink-Server verwendet werden. Siehe „ <a href="#">mlsrv16-Option -zt</a> “ auf Seite 104.                                                                                                       |

| Option                  | Beschreibung                                                                                                                                                                       |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-zu</b> { +   - }    | Steuert das automatische Hinzufügen von Benutzern, wenn das Skript <code>authenticate_user</code> nicht definiert ist. Siehe „ <a href="#">mlsrv16-Option -zu</a> “ auf Seite 104. |
| <b>-zus</b>             | Bewirkt, dass MobiLink Uploadskripten für Tabellen aufruft, für die keine Uploaddaten vorhanden sind. Siehe „ <a href="#">mlsrv16-Option -zus</a> “ auf Seite 105.                 |
| <b>-zw</b> 1,...5       | Steuert, welche Stufe der Warnmeldungen angezeigt werden soll.. Siehe „ <a href="#">mlsrv16-Option -zw</a> “ auf Seite 105.                                                        |
| <b>-zwd</b> <i>code</i> | Deaktiviert spezifische Warncodes. Siehe „ <a href="#">mlsrv16-Option -zwd</a> “ auf Seite 106.                                                                                    |
| <b>-zwe</b> <i>code</i> | Aktiviert spezifische Warncodes. Siehe „ <a href="#">mlsrv16-Option -zwe</a> “ auf Seite 106.                                                                                      |

### Bemerkungen

Der MobiLink-Server öffnet Verbindungen zum Server Ihrer konsolidierten Datenbank über ODBC. Danach akzeptiert er Verbindungen von Clientanwendungen und steuert den Synchronisationsprozess.

Sie müssen die Verbindungsparameter für die konsolidierte Datenbank mit der Option `-c` übergeben. Die Befehlszeilenoptionen können in beliebiger Reihenfolge angeführt werden. Die Option `-c` wird hier lediglich der Übersicht halber an erster Stelle angeführt. In der Liste der Optionen kann sie an beliebiger Stelle stehen, muss jedoch vor der Verbindungszeichenfolge angegeben werden.

Wenn Ihre ODBC-Datenquelle nicht für den automatischen Start der konsolidierten Datenbank konfiguriert wurde, muss die Datenbank vor dem MobiLink-Server gestartet werden.

### Siehe auch

- „[MobiLink-Server](#)“ auf Seite 30

## mlsrv16-Option @data

Liest Optionen aus der angegebenen Umgebungsvariablen oder Konfigurationsdatei ein

### Syntax

```
mlsrv16 -c "connection-string" @data ...
```

### Bemerkungen

Benutzen Sie diese Option, um mlsrv16-Befehlszeilenoptionen aus der angegebenen Umgebungsvariablen oder Konfigurationsdatei einzulesen. Wenn beide mit demselben Namen vorhanden sind, wird die Umgebungsvariable verwendet.

Wenn Sie Kennwörter oder andere Informationen in einer Konfigurationsdatei schützen möchten, können Sie das Dienstprogramm zum Ausblenden von Dateien zum Verschleiern des Inhalts von Konfigurationsdateien verwenden.

#### Siehe auch

- „Konfigurationsdateien“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „Dienstprogramm zum Verschleiern von Dateien (dbfhide)“ [[SQL Anywhere Server - Datenbankadministration](#)]

## mlsrv16-Option -a

Weist den MobiLink-Server an, nach einem Synchronisationsfehler auf einer konsolidierten Datenbankverbindung diese Verbindung weiterhin zu verwenden.

#### Syntax

```
mlsrv16 -c "connection-string" -a ...
```

#### Bemerkungen

Wenn während der Synchronisation ein Fehler auftritt, trennt der MobiLink-Server standardmäßig automatisch die Verbindung mit der konsolidierten Datenbank und baut die Verbindung anschließend wieder auf. Das erneute Verbinden sichert die folgenden Synchronisationsstarts gegenüber einem bekannten Status ab. Ist dieses Verhalten nicht erforderlich, können Sie es mit dieser Option deaktivieren. Die Aufrechterhaltung von Statusinformationen hängt von Ihren Anforderungen ab und kann je nach der Art variieren, wie Sie die Funktionsweise der MobiLink-Skripten mit dem RDBMS konfiguriert haben. Dies trifft auch dann zu, wenn es sich bei der Datenbank um Oracle, SQL Anywhere oder um ein anderes unterstütztes Produkt handelt. Einige Statusinformationen müssen je nach der Clientanwendung eventuell reinitialisiert werden.

## mlsrv16-Option -b

Entfernt bei Spalten des Typs VARCHAR, CHAR, LONG VARCHAR oder LONG CHAR nachgestellte Leerzeichen aus den Zeichenfolgen während der Synchronisation.

#### Syntax

```
mlsrv16 -c "connection-string" -b ...
```

#### Bemerkungen

##### Hinweis

Verwenden Sie in der konsolidierten Datenbank unbedingt VARCHAR anstelle von CHAR, um dieses Problem zu vermeiden.

Mit dieser Option sollen Unterschiede zwischen dem SQL Anywhere CHAR-Datentyp und dem CHAR- oder VARCHAR-Datentyp aufgelöst werden, die von der konsolidierten Datenbank verwendet werden. Der SQL Anywhere CHAR-Datentyp entspricht VARCHAR. In den meisten konsolidierten Datenbanken,

die nicht unter SQL Anywhere laufen, ist der CHAR(n)-Datentyp mit Leerzeichen auf n Zeichen aufgefüllt.

Wenn -b angegeben wird, entfernt der MobiLink-Server nachgestellte Leerzeichen aus Zeichenfolgen für Spalten des Typs CHAR, VARCHAR, LONG CHAR oder VARCHAR, wenn die Spalte in der entfernten Datenbank eine Zeichenfolge ist. Die gekürzten Daten werden dann in einem Download in die entfernten Datenbanken eingelesen.

Diese Option kann auch verwendet werden, um Konfliktaktualisierungen richtig zu erkennen, wenn das upload\_fetch-Skript oder upload\_fetch\_column\_conflict-Skript verwendet wird. Für jede aktualisierte Uploadzeile holt der MobiLink-Server die Zeile aus der konsolidierten Datenbank für den angegebenen Primärschlüssel, vergleicht die Zeile mit dem Pre-Image der Aktualisierung und bestimmt dann, ob die Aktualisierung einen Konflikt bewirkt. Wenn -b verwendet wird, kürzt MobiLink nachgestellte Leerzeichen aus Spalten des Typs CHAR, VARCHAR, LONG CHAR oder LONG VARCHAR, bevor der Vergleich vorgenommen wird.

### Siehe auch

- [CHAR-Spalten auf Seite 6](#)
- [„NVARCHAR-Datentyp“ \[SQL Anywhere Server - SQL-Referenzhandbuch\]](#)
- [„upload\\_fetch \(Tabellenereignis\)“ auf Seite 437](#)
- [„upload\\_fetch\\_column\\_conflict \(Tabellenereignis\)“ auf Seite 439](#)

### Beispiel

Wenn die Option -b nicht verwendet wird, wird ein Primärschlüsselwert 'abc' in einem Upload von einer entfernten SQL Anywhere- oder UltraLite-Datenbank in eine CHAR(10)-Spalte in der konsolidierten Datenbank zu 'abc' gefolgt von sieben Leerstellen umgewandelt. Wenn für dieselbe Zeile ein Download erfolgt, erscheint sie in der entfernten Datenbank als 'abc' gefolgt von sieben Leerstellen. Wenn die entfernte Datenbank keine Leerzeichenfüllung hat, enthält die entfernte Datenbank jetzt zwei Zeilen: 'abc' und 'abc' mit sieben Leerstellen. In der entfernten Datenbank gibt es jetzt eine Duplikatzeile.

Wenn die Option -b verwendet wird, wird der Primärschlüsselwert 'abc' in einem Upload von einer entfernten SQL Anywhere- oder UltraLite-Datenbank in eine CHAR(10)-Spalte in der konsolidierten Datenbank zu 'abc', gefolgt von sieben Leerzeichen, umgewandelt. Mit den sieben Leerstellen wird der Wert immer noch auf zehn Zeichen aufgefüllt, aber wenn dieselbe Zeile in einem Download übertragen wird, kürzt der MobiLink-Server die nachgestellten Leerzeichen und der Wert erscheint in der entfernten Datenbank als 'abc'. Die Option -b behebt das Problem der Duplikatzeile.

## mlsrv16-Option -bn

Legt die maximale Anzahl von BLOB-Byte für den Vergleich während der Konfliktermittlung fest.

### Syntax

**mlsrv16 -c "connection-string" -bn size ...**

### Bemerkungen

Wenn zwei BLOBS ähnliche oder identische Werte enthalten, kann der Vergleich der beiden zur Filterung bzw. zur Konfliktsuche je nach der betroffenen Datenmenge aufwändig sein. Diese Option weist

den MobiLink-Server an, nur die ersten *size* Byte von zwei BLOBs beim Vergleich zu berücksichtigen. Standardmäßig werden die beiden BLOBs in ihrer Gesamtheit verglichen.

In einigen Fällen kann die Begrenzung der maximal zu vergleichenden Datengröße die Synchronisation erheblich beschleunigen; sie kann jedoch auch zu Fehlern führen. Wenn sich z.B. zwei große BLOBs nur in den letzten paar Byte unterscheiden, könnte der MobiLink-Server entscheiden, dass sie identisch sind, obwohl sie es in Wahrheit nicht sind.

## mhsrv16-Option -c

Legt die Parameter für die konsolidierte Datenbank fest.

### Syntax

```
mhsrv16 -c "connection-string" ...
```

### Bemerkungen

Die Verbindungszeichenfolge muss dem MobiLink-Server ausreichend Daten zur Verfügung stellen, damit die Verbindung zur konsolidierten Datenbank hergestellt werden kann. Die Verbindungszeichenfolge ist erforderlich.

Die Verbindungszeichenfolge muss Verbindungsparameter in der Form *keyword=value*, getrennt durch Semikola, ohne Leerstellen zwischen den Parametern angeben.

Verbindungsparameter müssen in der Spezifikation der ODBC-Datenquelle enthalten sein, falls sie nicht in der Befehlszeile angegeben werden. Die erforderlichen Verbindungsdaten finden Sie in Ihrem RDBMS und in der ODBC-Datenquelle.

Eine vollständige Liste der SQL Anywhere-Verbindungsparameter finden Sie im Abschnitt „Verbindungsparameter“ [[SQL Anywhere Server - Datenbankadministration](#)].

Weitere Hinweise zur Verschleierung des Kennworts finden Sie unter „Dienstprogramm zum Verschleiern von Dateien (dbfhide)“ [[SQL Anywhere Server - Datenbankadministration](#)].

### Beispiel

```
mhsrv16 -c "DSN=odbcname;UID=DBA;PWD=sql"
```

## mhsrv16-Option -ca

Legt den Hostnamen oder die IP-Adresse des MobiLink-Arbiterservers fest, um den MobiLink-Server darüber zu informieren, wo der MobiLink-Arbitrer ausgeführt wird.

### Syntax

```
mhsrv16 -c "connection-string" -ca host_or_ip ...
```

### Bemerkungen

Alle MobiLink-Server einer Serverfarm müssen dieselbe Einstellung für die Option -ca enthalten.

Verwenden Sie neben der Option `-ca` auch die Option `-lsc`, um die Verbindungszeichenfolge für den lokalen MobiLink-Server anzugeben.

Die Befehlszeilenoptionen `-ca` und `-lsc` werden vom MobiLink-Server ignoriert, wenn seine Befehlszeile nicht die Option `-notifier` enthält.

**Hinweis**

Port 4953 wurde dem MobiLink-Arbitrator zugewiesen. Daher kann die Portnummer auf dem Computer, auf dem der MobiLink-Arbitratorserver ausgeführt wird, nicht von anderen Anwendungen verwendet werden.

**Siehe auch**

- „`mlsrv16`-Option `-lsc`“ auf Seite 65
- „`mlsrv16`-Option `-notifier`“ auf Seite 66

## mlsrv16-Option `-cinit`

Legt die Ausgangscachegröße des Serverspeichers fest.

**Syntax**

`mlsrv16 -c "connection-string" -cinit size[ k | m | g | p ] ...`

**Bemerkungen**

Die anfängliche Speichermenge, die der Server zum Speichern von Tabellendaten, Netzwerkpuffern, im Cache gespeicherten Downloaddaten und anderen Strukturen verwendet, die zur Synchronisation verwendet werden. Wenn sich mehr Daten auf dem Server befinden als für diesen Speicherpool zugelassen, werden die Daten auf die Festplatte ausgelagert. Das Auslagern der Daten auf die Festplatte führt zu einer deutlichen Leistungseinbuße.

*size* ist die zu reservierende Speichermenge in Byte. Verwenden Sie **k**, **m** oder **g**, um die Einheit in kB, MB bzw. GB anzugeben. Wenn der Zahl kein Buchstabe folgt, ist die Größe in Byte angegeben.

Die Einheit **p** ist ein Prozentsatz des physischen Systemspeichers oder des vom Prozess adressierbaren Speicherplatzes, wobei der niedrigere Wert genommen wird. Die maximale Größe des vom Prozess adressierbaren Speicherplatzes hängt vom Betriebssystem ab. Beispiel:

- 2,5 GB für Windows Advanced Server, Enterprise Server und Datacenter Server (32-Bit-Version).
- 3,5 GB für den 32-Bit-Datenbankserver, der unter Windows x64 Edition läuft.
- 1,5 GB auf allen anderen 32-Bit-Systemen.
- Auf 64-Bit-Datenbankservern kann die Cachegröße als unbegrenzt betrachtet werden

Der Standardwert ist **50m**.

**Siehe auch**

- „mlsrv16-Option -cmax“ auf Seite 55
- „mlsrv16-Option -cmin “ auf Seite 56
- „Speichernutzung im MobiLink-Server“ auf Seite 41

## mlsrv16-Option -cm

Legt die maximale Cachegröße des Serverspeichers fest. Diese Option ist ein Alias für die Option -cmax. Siehe „mlsrv16-Option -cmax“ auf Seite 55.

## mlsrv16-Option -cmax

Legt die maximale Cachegröße des Serverspeichers fest.

**Syntax**

**mlsrv16 -c "connection-string" -cmax size[ k | m | g | p ] ...**

**Bemerkungen**

Die maximale Speichermenge, die der Server zum Speichern von Tabellendaten, Netzwerkpuffern, im Cache gespeicherten Downloaddaten und anderen Strukturen verwendet, die zur Synchronisation verwendet werden. Wenn sich mehr Daten auf dem Server befinden als für diesen Speicherpool zugelassen, werden die Daten auf die Festplatte ausgelagert. Das Auslagern der Daten auf die Festplatte führt zu einer deutlichen Leistungseinbuße.

*size* ist die zu reservierende Speichermenge in Byte. Verwenden Sie **k**, **m** oder **g**, um die Einheit in kB, MB bzw. GB anzugeben. Wenn der Zahl kein Buchstabe folgt, ist die Größe in Byte angegeben.

Die Einheit **p** ist ein Prozentsatz des physischen Systemspeichers oder des vom Prozess adressierbaren Speicherplatzes, wobei der niedrigere Wert genommen wird. Die maximale Größe des vom Prozess adressierbaren Speicherplatzes hängt vom Betriebssystem ab. Beispiel:

- 2,5 GB für Windows Advanced Server, Enterprise Server und Datacenter Server (32-Bit-Version).
- 3,5 GB für den 32-Bit-Datenbankserver, der unter Windows x64 Edition läuft.
- 1,5 GB auf allen anderen 32-Bit-Systemen.
- Auf 64-Bit-Datenbankservern kann die Cachegröße als unbegrenzt betrachtet werden

Der Standardwert ist **70p**.

**Siehe auch**

- „mlsrv16-Option -cinit “ auf Seite 54
- „mlsrv16-Option -cmin “ auf Seite 56
- „Speichernutzung im MobiLink-Server“ auf Seite 41

## mlsrv16-Option -cmin

Legt die Mindestcachegröße des Serverspeichers fest.

### Syntax

**mlsrv16 -c** "*connection-string*" **-cmin** *size* [ **k** | **m** | **g** | **p** ] ...

### Bemerkungen

Die minimale Speichermenge, mit der der Server Tabellendaten, Netzwerkpuffer, im Cache gespeicherte Downloaddaten und andere Strukturen speichert, die zur Synchronisation verwendet werden. Wenn sich mehr Daten auf dem Server befinden als für diesen Speicherpool zugelassen, werden die Daten auf die Festplatte ausgelagert. Das Auslagern der Daten auf die Festplatte führt zu einer deutlichen Leistungseinbuße.

*size* ist die zu reservierende Speichermenge in Byte. Verwenden Sie **k**, **m** oder **g**, um die Einheit in kB, MB bzw. GB anzugeben. Wenn der Zahl kein Buchstabe folgt, ist die Größe in Byte angegeben.

Die Einheit **p** ist ein Prozentsatz des physischen Systemspeichers oder des vom Prozess adressierbaren Speicherplatzes, wobei der niedrigere Wert genommen wird. Die maximale Größe des vom Prozess adressierbaren Speicherplatzes hängt vom Betriebssystem ab. Beispiel:

- 2,5 GB für Windows Advanced Server, Enterprise Server und Datacenter Server (32-Bit-Version).
- 3,5 GB für den 32-Bit-Datenbankserver, der unter Windows x64 Edition läuft.
- 1,5 GB auf allen anderen 32-Bit-Systemen.
- Auf 64-Bit-Datenbankservern kann die Cachegröße als unbegrenzt betrachtet werden

Der Standardwert ist **50m**.

### Siehe auch

- „mlsrv16-Option -cmax“ auf Seite 55
- „mlsrv16-Option -cinit“ auf Seite 54
- „Speichernutzung im MobiLink-Server“ auf Seite 41

## mlsrv16-Option -cn

Legt die maximale Anzahl von gleichzeitigen Verbindungen zu konsolidierten Datenbanken für Datenbank-Worker-Threads fest.

### Syntax

**mlsrv16 -c** "*connection-string*" **-cn** *value* ...

### Bemerkungen

Mit dieser Option wird die Höchstzahl der gleichzeitigen Verbindungen festgelegt, die der MobiLink-Server zur konsolidierten Datenbank für Datenbank-Worker-Threads aufbauen soll. Der minimale und der Standardwert sind so groß wie die Anzahl der Datenbank-Worker-Threads. Es wird eine Warnung

ausgegeben, wenn der angegebene Wert kleiner als die Anzahl der Datenbank-Worker-Threads ist, und der Wert wird automatisch nach oben angepasst.

Dieser Typ von MobiLink-Datenbankverbindung wird nur für Synchronisationen mit einer Skriptversion verwendet. Wenn der MobiLink-Server alle Datenbankverbindungen verwendet, die von der Option `-cn` zugelassen werden, und eine Synchronisation aussteht, derzeit aber keine Datenbankverbindung für ihre Skriptversion vorhanden ist, trennt der MobiLink-Server eine Verbindung und erstellt dann eine neue Datenbankverbindung für die Skriptversion der ausstehenden Synchronisation.

Ein Wert, der größer als die Anzahl der Datenbank-Worker-Threads ist, kann die Performance insbesondere dann erhöhen, wenn die Verbindung mit der konsolidierten Datenbank langsam ist oder wenn mehrere Skriptversionen verwendet werden. Die optimale Höchstzahl der Datenbankverbindungen ist die Anzahl der Skriptversionen, multipliziert mit der Anzahl der Datenbank-Worker-Threads. Weitere, über diesen Optimalwert hinausgehende Verbindungen beschleunigen nicht unbedingt die Synchronisation und verbrauchen Ressourcen auf Seiten des MobiLink-Servers und des Servers der konsolidierten Datenbank.

#### Siehe auch

- „`mlsrv16-Option -w`“ auf Seite 90

## mlsrv16-Option -cr

Legt die maximale Anzahl der Versuche für Datenbankverbindungen fest.

#### Syntax

```
mlsrv16 -c "connection-string" -cr value ...
```

#### Bemerkungen

Mit dieser Option wird eingestellt, wie oft der MobiLink-Server höchstens versuchen soll, eine Verbindung zur Datenbank herzustellen, falls die Qualität einer Verbindung nachlässt. Standardwert ist drei Verbindungsversuche.

## mlsrv16-Option -cs

Legt Verbindungsparameter für die MobiLink-Systemdatenbank fest (MLSD).

#### Syntax

```
mlsrv16 -c "connection-string" -cs "connection-string" ...
```

#### Bemerkungen

MobiLink-Server-Systemobjekte, z.B. Systemtabellen, Prozeduren, Trigger und Ansichten, können in einer anderen als der konsolidierten Datenbank gespeichert werden. Die Datenbank, die die MobiLink-Systemobjekte speichert, wird als MLSD bezeichnet.

Wenn dieser Befehlsparameter in der Befehlszeile verwendet wird, stellt der MobiLink-Server Verbindungen zur MLSD her, um benutzerdefinierte Skripten abzurufen und den Synchronisationsstatus

zu verwalten, z.B. ML-Benutzernamen, entfernte IDs, Fortschritts-Offsets und die Zeitstempel der letzten Uploads und Downloads. Der MobiLink-Server verwendet die ursprünglichen im Befehlszeilenparameter -c angegebenen Verbindungen mit der konsolidierten Datenbank für den Upload und Download von Daten in die Client-Datenbanken. Die konsolidierte Datenbank braucht keines der MobiLink-Server-Systemobjekte zu haben. Alle benutzerdefinierten Skripten, einschließlich der Skripten für die Fehler-Berichterstellung und die Fehlerbehandlung werden von der MLSD abgerufen und in der konsolidierten Datenbank ausgeführt.

Wenn diese Option verwendet wird, benötigt der MobiLink-Server den Microsoft Distributed Transaction Coordinator (MSDTC).

Die konsolidierte Datenbank und MLSD können eine der unterstützten konsolidierten MobiLink-Datenbanken sein. Die entsprechenden ODBC-Treiber müssen jedoch Microsoft Distributed Transactions unterstützen.

Die konsolidierte Datenbank und MLSD müssen über ein Transaktionslog verfügen, um MSDTC zu verwenden.

Diese Option kann nur auf Windows-Betriebssystemen verwendet werden.

## mlsrv16-Option -ct

Legt in Minuten die Zeitdauer fest, nach der die Verbindung vom MobiLink-Server getrennt wird, wenn keine Aktivität auf der Verbindung herrscht.

### Syntax

```
mlsrv16 -c "connection-string" -ct connection-timeout ...
```

### Bemerkungen

MobiLink-Datenbankverbindungen, die eine bestimmte Zeit ungenutzt bleiben, werden vom Server getrennt. Der Timeout wird mit der Option -ct festgelegt. Der Standardwert für die Timeoutdauer ist 60 Minuten.

## mlsrv16-Option -dl

Zeigt alle MobiLink-Servermeldungen auf dem Bildschirm im Meldungsfenster des MobiLink-Servers an.

### Syntax

```
mlsrv16 -c "connection-string" -v -dl ...
```

### Bemerkungen

Zeigt alle MobiLink-Servermeldungen im Meldungsfenster des MobiLink-Servers an. Standardmäßig erscheint nur eine Teilmenge aller Meldungen im Fenster, wenn die MobiLink-Servermeldungen in eine Logdatei geleitet werden (mit -o). Wenn viele Meldungen generiert werden, kann diese Option die Performance beeinträchtigen.

**Siehe auch**

- „mlsrv16-Option -o“ auf Seite 67
- „Datenbankservermeldungen in einer Datei protokollieren“ [*SQL Anywhere Server - Datenbankadministration*]

## mlsrv16-Option -dr

Nur für Adaptive Server Enterprise. Stellt sicher, dass an der Synchronisation beteiligte Tabellen das DataRow-Sperrschema nicht verwenden.

**Syntax**

**mlsrv16 -c "connection-string" -dr ...**

**Bemerkungen**

Verwenden Sie diese Option nur, wenn keine der gerade synchronisierten konsolidierten Tabellen mit dem DataRow-Sperrschema erstellt wurde.

Diese Option verringert das Senden von redundanten Daten durch den MobiLink-Server.

**Siehe auch**

- „MobiLink-Isolationsstufen“ auf Seite 140

## mlsrv16-Option -ds

Zur Verwendung mit neu startbaren Downloads. Legt die maximale Menge an Daten auf der Festplatte fest, die der MobiLink-Server zum Speichern aller neu startbaren Downloads verwenden kann.

**Syntax**

**mlsrv16 -c "connection-string" -ds size[ **k** | **m** | **g** ] ...**

**Bemerkungen**

Der MobiLink-Server nimmt Downloaddaten auf, die vom Client nicht empfangen wurden, um sie in einem neu startbaren Download verwenden zu können. Die Option beschränkt die Datenmenge, die der Server für alle kombinierten Synchronisationen aufnimmt.

Ist *size* zu klein, kann der Server Downloaddaten freigeben, wodurch der Neustart eines Downloads unmöglich wird. Der Server gibt Downloaddaten erst frei, wenn eine der folgenden Situationen eintritt:

- Der Benutzer schließt den Download erfolgreich ab.
- Der Benutzer gibt eine neue Synchronisationsanforderung ein, ohne dass die Wiederaufnahme aktiviert wurde.
- Der Cache wird für eingehende Anforderungen benötigt. Der älteste nicht erfolgreiche Download wird zuerst gelöscht.

Verwenden Sie **k**, **m** oder **g**, um die Einheit in kB, MB bzw. GB anzugeben. Der Standardwert ist **10m**.

Solange Daten für einen neu startbaren Download gespeichert sind, betrachtet der MobiLink-Server die Synchronisation weiterhin als aktiv (in der Download\_senden-Phase des MobiLink-Profilers) und ein Überschreiten des Netzwerk-Timeouts führt nicht zum Schließen der Synchronisation.

#### Siehe auch

- „Wiederaufnahme fehlgeschlagener Downloads“ auf Seite 135
- „dbmlsync-Option -dc“ [*MobiLink - Clientadministration*]
- „Synchronisationsparameter Resume Partial Download“ [*UltraLite - Datenbankverwaltung*]

## mlsrv16-Option -dsd

Deaktiviert die Snapshot-Isolation.

#### Syntax

**mlsrv16 -c "connection-string" -dsd ...**

#### Bemerkungen

Wenn eine konsolidierte SQL Anywhere- (ab Version 10) oder Microsoft SQL Server-Datenbank (ab 2005) verwendet wird, ist die Snapshot-Isolation die Standardisolationsstufe für Downloads. Ist die konsolidierte Datenbank eine frühere Version dieser Datenbanken, lautet die Standardisolationsstufe für Downloads READ COMMITTED.

Sie können die Standardisolationsstufe auch in einem Skript ändern. Bei Datenbanken ab SQL Anywhere Version 10 und Microsoft SQL Server 2005 wird die Isolationsstufe jedoch zu Beginn der Upload- und Downloadtransaktionen festgelegt. Wenn Sie die Isolationsstufe im begin\_connection-Skript festlegen, kann sie in den Skripten begin\_upload und begin\_download aufgehoben werden.

Diese Option gilt für konsolidierte Datenbanken in SQL Anywhere Version 10 und Microsoft SQL Server 2005.

#### Siehe auch

- „MobiLink-Isolationsstufen“ auf Seite 140
- „mlsrv16-Option -dt“ auf Seite 60
- „mlsrv16-Option -esu“ auf Seite 62

## mlsrv16-Option -dt

Nur für Microsoft SQL Server- und Adaptive Server Enterprise-Datenbanken. Verursacht, dass MobiLink nur Transaktionen in der laufenden Datenbank erkennt.

#### Syntax

**mlsrv16 -c "connection-string" -dt ...**

## Bemerkungen

Diese Option bewirkt, dass MobiLink alle Transaktionen mit Ausnahme derjenigen in der aktuellen Datenbank ignoriert. Sie erhöht den Durchsatz und verringert die Duplikation von Zeilen beim Download.

Diese Option gilt nur für zeitstempelbasierte Downloads.

Verwenden Sie diese Option, wenn Folgendes zutrifft:

- Ihre konsolidierte Datenbank läuft auf Microsoft SQL Server oder Adaptive Server Enterprise, wo auch andere Datenbanken ausgeführt werden.
- Sie verwenden die Snapshot-Isolation für Uploads oder Downloads auf Microsoft SQL Server.
- Sie verwenden das DataRow-Sperrschema für Synchronisationstabellen auf Adaptive Server Enterprise.
- Ihre Upload- oder Download-Skripten greifen auf keine anderen Datenbanken auf dem Server zu.

Diese Option wird nur auf Microsoft SQL Server-Datenbanken angewendet, die die Snapshot-Isolation verwenden, und Adaptive Server Enterprise-Datenbanken, die das DataRow-Sperrschema für an der Synchronisation beteiligte Tabellen verwenden.

## Siehe auch

- „MobiLink-Isolationsstufen“ auf Seite 140
- „mlsrv16-Option -dsd“ auf Seite 60
- „mlsrv16-Option -esu“ auf Seite 62

## mlsrv16-Option -e

Speichert Fehlerlogs, die von SQL Anywhere MobiLink-Clients gesendet wurden.

## Syntax

**mlsrv16 -c "connection-string" -e filename ...**

## Bemerkungen

Ohne die Option -e werden Fehlerprotokolle von SQL Anywhere-MobiLink-Clients in einer Datei mit dem Namen *mlsrv16.mle* gespeichert. Mit der Option -e wird der MobiLink-Server angewiesen, die Fehlerprotokolle in der angegebenen Datei zu speichern. Wenn ein Fehler auftritt, sendet dbmlsync standardmäßig bis zu 32 Kilobyte Logmeldungen von der entfernten Datenbank an den MobiLink-Server.

Diese Option ermöglicht den zentralen Zugang zu entfernten Fehlerlogs, um Synchronisationsfehler diagnostizieren zu können.

Der Umfang der von einer entfernten Datenbank übertragenen Datenmenge kann über die erweiterte Option `ErrorLogSendLimit` von dbmlsync gesteuert werden.

### Siehe auch

- „mlsrv16-Option -et“ auf Seite 62
- „Erweiterte Option ErrorLogSendLimit (el)“ [*MobiLink - Clientadministration*]

## mlsrv16-Option -esu

Verwendet die Snapshot-Isolation für Uploads.

### Syntax

**mlsrv16 -c "connection-string" -esu ...**

### Bemerkungen

Standardmäßig verwendet MobiLink die Isolationsstufe 1 (festgeschriebene Daten lesen, „read committed“) für Uploads. Dies ist in der Regel die optimale Isolationsstufe.

Wenn Sie für Uploads die Snapshot-Isolation verwenden, generieren Sie unter Umständen Konflikte bei Snapshot-Transaktionen während der Aktualisierung von Uploads. Wenn dies geschieht, setzt der MobiLink-Server den gesamten Upload zurück und wiederholt den Vorgang. In diesem Fall sollten Sie Ihre Einstellungen für die MobiLink-Serveroptionen -r oder -rd anpassen, um die Zeitverzögerung zwischen den Wiederholungsversuchen und der maximalen Anzahl von Versuchen festzulegen.

Sie können die Standardisolationsstufe in einem Skript ändern. Sie ändern die Isolationsstufe für Uploads für gewöhnlich im Skript begin\_upload.

Diese Option gilt nur für konsolidierte Datenbanken in SQL Anywhere Version 10 und später und Microsoft SQL Server 2005 und später.

### Siehe auch

- „MobiLink-Isolationsstufen“ auf Seite 140
- „mlsrv16-Option -dsd “ auf Seite 60
- „mlsrv16-Option -dt “ auf Seite 60
- „mlsrv16-Option -r “ auf Seite 75
- „mlsrv16-Option -rd “ auf Seite 76

## mlsrv16-Option -et

Speichert Fehlerlogs von SQL Anywhere MobiLink-Clients in der angegebenen Datei nach dem Kürzen der vorhandenen Datei.

### Syntax

**mlsrv16 -c "connection-string" -et filename ...**

### Bemerkungen

Die Option -et hat die gleiche Wirkung wie -e, nur dass die Fehlerlogdatei gekürzt wird, bevor neue Fehler hinzugefügt werden.

Der Umfang der von einer entfernten Datenbank übertragenen Datenmenge kann über die erweiterte Option `ErrorLogSendLimit` von `dbmlsync` gesteuert werden.

#### Siehe auch

- „Erweiterte Option `ErrorLogSendLimit` (el)“ [[MobiLink - Clientadministration](#)]
- „`mlsrv16`-Option -e“ auf Seite 61

## mlsrv16-Option -fips

Erzwingt die Verwendung FIPS-zertifizierter Module für alle sicheren MobiLink-Datenströme.

#### Syntax

`mlsrv16 -c connection-string" -fips ...`

#### Bemerkungen

Die Angabe dieser Option erzwingt, dass jede MobiLink-Verschlüsselung FIPS-zertifizierte Algorithmen verwendet. Sie können unverschlüsselte Verbindungen auch verwenden, wenn die Option `-fips` angegeben wurde, eine einfache Verschlüsselung ist jedoch nicht möglich.

Mit dieser Option werden FIPS-zertifizierte Algorithmen für Verbindungen verwendet, und zwar unabhängig davon, ob Sie sie angeben. Wenn Sie beispielsweise den MobiLink-Server mit den Optionen `-fips` und `-x tls(...;fips=no;...)` starten, wird die Einstellung `fips=no` ignoriert, und der Server startet mit der Einstellung `fips=yes`.

#### Hinweis

Erforderliche getrennt lizenzierte Komponenten.

FIPS-zertifizierte Verschlüsselung erfordert eine separate Lizenz. Alle Technologien für starke Verschlüsselungen unterliegen Exportbestimmungen.

Siehe „Getrennt lizenzierte Komponenten“ [[SQL Anywhere 16 - Einführung](#)].

Bei der MobiLink-Transportschichtsicherheit bewirkt die Option `-fips`, dass der Server die FIPS-zertifizierten RSA-Verschlüsselungsalgorithmen verwendet, auch wenn RSA ohne die Option `-fips` angegeben ist.

#### Siehe auch

- „Verschlüsselung der MobiLink-Client/Server-Kommunikation“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „FIPS-zertifizierte Verschlüsselungstechnologie“ [[SQL Anywhere Server - Datenbankadministration](#)]

## mlsrv16-Option -ftr

Gibt einen Speicherort für vom Dienstprogramm `mlfiletransfer` oder vom MobiLink-Agent herunterzuladende Dateien an.

## Syntax

```
mlsrv16 -c "connection-string" -ftr path ...
```

## Bemerkungen

Diese Option legt das Stammverzeichnis für die Dateiübertragung fest. Dateien, die an einen Benutzer übertragen werden sollen, können im Stammverzeichnis oder in einem Unterverzeichnis mit dem Benutzernamen gespeichert werden. MobiLink sucht zuerst nach der angeforderten Datei in einem Unterverzeichnis des Stammverzeichnisses für die Dateiübertragung mit dem Benutzernamen des verbundenen Clients. Wenn die Datei sich nicht in diesem Unterverzeichnis befindet, sucht MobiLink sie im Stammverzeichnis für die Dateiübertragung.

Diese Option ist erforderlich, wenn Sie das Dienstprogramm mlfiletransfer zum Herunterladen von Dateien verwenden möchten.

## Siehe auch

- „MobiLink-Dienstprogramm für die Dateiübertragung (mlfiletransfer)“ [[MobiLink - Clientadministration](#)]
- „mlsrv16-Option -ftru“ auf Seite 64
- „authenticate\_file\_transfer (Verbindungsereignis)“ auf Seite 284

## mlsrv16-Option -ftru

Gibt einen Speicherort für vom Dienstprogramm mlfiletransfer oder vom MobiLink-Agent hochzuladende Dateien an.

## Syntax

```
mlsrv16 -c "connection-string" -ftru path ...
```

## Bemerkungen

Mit dieser Option wird das Stammverzeichnis für Dateien festgelegt, die mit dem mlfiletransfer-Dienstprogramm hochgeladen werden sollen. Dateien können nur in dieses Stammverzeichnis oder unmittelbare Unterverzeichnisse des Stammverzeichnisses hochgeladen werden.

Dateien können nur hochgeladen werden, wenn das authenticate\_file\_upload-Skript nicht vorhanden ist, oder wenn das Skript vorhanden ist und einen authentication\_code im Bereich von 1000-1999 zurückgibt. Diese Anforderung gilt nur für mlfiletransfer und nicht für den MobiLink-Agent.

Diese Option ist erforderlich, wenn Sie das Dienstprogramm mlfiletransfer zum Hochladen von Dateien verwenden möchten.

## Siehe auch

- „MobiLink-Dienstprogramm für die Dateiübertragung (mlfiletransfer)“ [[MobiLink - Clientadministration](#)]
- „mlsrv16-Option -ftr“ auf Seite 63
- „authenticate\_file\_transfer (Verbindungsereignis)“ auf Seite 284

## mlsrv16-Option -lsc

Gibt die Verbindungsinformationen für den lokalen Server an. Diese Informationen werden anderen Servern in der Serverfarm übergeben.

### Syntax

```
mlsrv16 -c "connection-string" -lsc protocol[protocol-options] ...
```

*protocol* : tcpip | tls | http | https

*protocol-options* : ( option=value; ... )

### Bemerkungen

Diese Option wird nur in der folgenden Situation verwendet:

- Wenn der Notifier in einer MobiLink-Serverfarm ausgeführt wird.
- Wenn das mlreplay-Dienstprogramm mit der Serveroption -rrp verwendet wird.

Wenn z.B. ein Server auf einem Host namens server\_rack10 ausgeführt wird, könnte die Befehlszeile folgendermaßen beginnen:

```
mlsrv16 -x tcpip(port=200) -zs server5 -lsc
tcpip(host=server_rack10;port=200) -c...
```

In diesem Beispiel würde ein anderer Server den gemeinsamen Status in der konsolidierten Datenbank verwenden, um die Verbindungszeichenfolge tcpip(host=server\_rack10;port=200) abzurufen und mit ihr eine Verbindung zu dem soeben gestarteten Server herzustellen.

### Siehe auch

- „mlsrv16-Option -rrp“ auf Seite 76
- „MobiLink-Server in einer Serverfarm“ auf Seite 41
- „mlsrv16-Option -zs“ auf Seite 103
- „mlsrv16-Option -rrp“ auf Seite 76
- „Notifier in einer MobiLink-Serverfarm“ [*MobiLink - Serverinitiierte Synchronisation*]

## mlsrv16-Option -nc

Legt die maximale Anzahl gleichzeitiger Netzwerkverbindungen fest.

### Syntax

```
mlsrv16 -c "connection-string" -nc connections ...
```

### Bemerkungen

Der MobiLink-Server weist neue Synchronisationsverbindungen zurück, wenn der Höchstwert erreicht ist. Auf dem Client wird ein Verbindungsfehler mit einem Systemfehlercode ausgegeben, der anzeigt, dass die Verbindung abgelehnt wurde.

Standardwert ist "1024".

Zur Begrenzung der Anzahl gleichzeitiger Synchronisationen für ein nichtbeständiges HTTP/HTTPS legen Sie für -nc einen Wert fest, der erheblich höher als der Wert von -sm ist. Wenn der Grenzwert von -sm erreicht wurde, wird vom MobiLink-Server der HTTP-Fehler 503 (Dienst nicht verfügbar) an den entfernten Client gesendet. Wenn der Grenzwert von -nc erreicht wurde, wird dagegen ein Socket-Fehler ausgegeben. Je größer der Unterschied zwischen -nc und -sm ist, desto wahrscheinlicher ist es, dass die abgelehnten Verbindungen den HTTP-Fehler 503 anstelle des weniger aussagekräftigen Socket-Fehlers generiert. Legen Sie z.B. -sm auf 100 und -nc auf 1000 fest.

Der Maximalwert für -nc hängt vom Betriebssystem und seiner Konfiguration ab. Sie müssen unter Umständen die Konfiguration optimieren, um eine größere Socket-Kapazität zu erzielen.

### Siehe auch

- „mlsrv16-Option -sm“ auf Seite 81

## mlsrv16-Option -notifier

Startet den Notifier für die serverinitiierte Synchronisation.

### Syntax

**mlsrv16 -c "connection-string" -notifier [ *notifier-properties-file* ] ...**

### Bemerkungen

Wenn Sie einen Namen für die Notifier-Konfigurationsdatei festlegen oder keinen Namen angeben, sondern eine Standard-Notifier-Eigenschaftsdatei namens *config.notifier* verwenden, wird der Notifier mit der betreffenden Datei konfiguriert. Hierbei werden jeweils die Konfigurationsinformationen aufgehoben, die in der Tabelle ml\_properties in der konsolidierten Datenbank gespeichert sind.

Andernfalls verwendet MobiLink die Konfigurationsinformationen aus der Tabelle ml\_properties in der konsolidierten Datenbank.

Wenn Sie die Option -notifier verwenden, starten Sie alle aktivierten Notifier.

Weitere Hinweise zum Aktivieren von Notifiern finden Sie unter „Notifier-Eigenschaften“ [*MobiLink - Serverinitiierte Synchronisation*].

### Siehe auch

- „MobiLink-Server-Einstellungen für serverinitiierte Synchronisation“ [*MobiLink - Serverinitiierte Synchronisation*]
- „Serverseitige Einstellungen, die mit der Notifier-Konfigurationsdatei konfiguriert wurden“ [*MobiLink - Serverinitiierte Synchronisation*]
- „Notifier“ [*MobiLink - Serverinitiierte Synchronisation*]
- „Notifier in einer MobiLink-Serverfarm“ [*MobiLink - Serverinitiierte Synchronisation*]

## mlsrv16-Option -o

Protokolliert Ausgabemeldungen in einer Logdatei des MobiLink-Servers und begrenzt die im Meldungsfenster des MobiLink-Servers protokollierten Daten.

### Syntax

**mlsrv16 -c "connection-string" -o logfile ...**

### Bemerkungen

Alle Logmeldungen in die angegebene Datei schreiben. Im Meldungsfenster des MobiLink-Servers wird, sofern es vorhanden ist, normalerweise nur eine Teilmenge der protokollierten Meldungen angezeigt.

Wenn bei der Synchronisation Fehler auftreten, gibt der MobiLink-Server den vollständigen Fehlerkontext in seiner Ausgabedatei an. Folgende Informationen können im Fehlerkontext enthalten sein:

- **Entfernte ID** Dies ist die entfernte ID der entfernten Datenbank, die synchronisiert wird.
- **Benutzername** Der tatsächliche Benutzername, der den MobiLink-Clients bei der Synchronisation zur Verfügung gestellt wurde.
- **Geänderter Benutzername** Dies ist der Benutzername, der über das modify\_user-Skript geändert wurde.
- **Transaktion** Hier wird die Transaktion angegeben, in der der Fehler aufgetreten ist. Die Transaktion kann authenticate\_user, begin\_synchronization, upload, prepare\_for\_download, download oder end\_synchronization sein.
- **Tabellenname** Zeigt den Tabellennamen an, sofern er verfügbar ist, sonst Null.
- **Zeilenoperation** Mögliche Vorgänge sind INSERT, UPDATE, DELETE oder FETCH.
- **Zeilendaten** Zeigt alle Spaltenwerte der Zeile, die den Fehler verursachte.
- **Skriptversion** Gegenwärtig für die Synchronisation verwendete Skriptversion.
- **Skript** Dies ist das Skript, das den Fehler verursacht hat.

Ungeachtet der gewählten Ausführlichkeitsebene erscheinen im Log Fehlerkontextinformationen.

### Siehe auch

- „mlsrv16-Option -os“ auf Seite 68
- „mlsrv16-Option -dl“ auf Seite 58
- „mlsrv16-Option -ot“ auf Seite 69
- „mlsrv16-Option -on“ auf Seite 68
- „mlsrv16-Option -v“ auf Seite 86

## mlsrv16-Option -on

Gibt eine Maximalgröße für die Logdatei des MobiLink-Servers an, bei deren Erreichen die Logdatei mit der Erweiterung .old umbenannt und eine neue Datei begonnen wird.

### Syntax

```
mlsrv16 -c "connection-string" -on size [ k | m ]...
```

### Bemerkungen

*size* ist die maximale Dateigröße für das Meldungslog, angegeben in Byte. Stellen Sie ein "k" oder "m" dahinter, um die Einheit in kB oder MB anzugeben. Die minimale Größenbegrenzung beträgt 10 kB.

Wenn die Logdatei die angegebene Größe erreicht, benennt der MobiLink-Server die Ausgabedatei mit der Erweiterung .old um und legt eine neue mit demselben Namen an.

#### Hinweis

Wenn die \*.old-Datei bereits existiert, wird sie überschrieben. Es werden höchstens zwei Dateien verwendet. Um den Verlust alter Logdateien zu verhindern, benutzen Sie die Option -os.

Diese Option kann nicht mit der Option -os verwendet werden.

### Siehe auch

- „mlsrv16-Option -o“ auf Seite 67
- „mlsrv16-Option -ot“ auf Seite 69
- „mlsrv16-Option -on“ auf Seite 68
- „mlsrv16-Option -os“ auf Seite 68
- „mlsrv16-Option -v“ auf Seite 86

## mlsrv16 Option -oq

Verhindert unter Windows, dass das Fehlerfenster angezeigt wird, falls beim Start ein Fehler auftritt.

### Syntax

```
mlsrv16 -c "connection-string" -oq ...
```

### Bemerkungen

Der MobiLink-Server zeigt standardmäßig ein Fenster an, wenn beim Start ein Fehler auftritt. Die Option -oq verhindert, dass dieses Fenster angezeigt wird.

## mlsrv16-Option -os

Legt die Maximalgröße von aktuellen und alten MobiLink-Server-Meldungslogdateien fest.

### Syntax

```
mlsrv16 -c "connection-string" -os size [ k | m ] ...
```

## Bemerkungen

*size* ist die maximale Dateigröße zum Protokollieren von Ausgabemeldungen. Die Standardeinheit ist Byte. Stellen Sie ein "k" oder "m" dahinter, um die Einheit in kB oder MB anzugeben. Die minimale Größenbegrenzung beträgt 10 kB.

Bevor der MobiLink-Server Ausgabemeldungen in eine Datei protokolliert, prüft er die aktuelle Dateigröße. Wenn aufgrund der Lognachricht die Dateigröße den festgelegten Wert übersteigt, benennt der MobiLink-Server die Meldungslogdatei in *jmmmtt<sub>xx</sub>.mls* um, wobei *xx* eine Zahl zwischen 00 und 99 ist und *jmmmtt* das aktuelle Datum (Jahr, Monat und Tag) darstellt.

Die letzte Ausgabe wird immer an die Datei angefügt, die mit -o oder -ot angegeben wurde.

Sie können diese Option nicht gemeinsam mit der Option -on verwenden.

### Hinweis

Diese Option erstellt eine unbegrenzte Anzahl von Logdateien. Um dies zu vermeiden, verwenden Sie -o oder -on.

## Siehe auch

- „mlsrv16-Option -o“ auf Seite 67
- „mlsrv16-Option -on“ auf Seite 68
- „mlsrv16-Option -ot“ auf Seite 69
- „mlsrv16-Option -v“ auf Seite 86

## mlsrv16-Option -ot

Protokolliert die Ausgabemeldungen in der Logdatei des MobiLink-Servers, nachdem der Inhalt der Datei gelöscht wurde.

## Syntax

**mlsrv16 -c "connection-string" -ot logfilename ...**

## Bemerkungen

In der Standardeinstellung wird die Ausgabe an das MobiLink-Servermeldungsfenster oder den entsprechenden Bildschirm gesendet.

## Siehe auch

- „mlsrv16-Option -on“ auf Seite 68
- „mlsrv16-Option -os“ auf Seite 68
- „mlsrv16-Option -v“ auf Seite 86
- „mlsrv16-Option -o“ auf Seite 67

## mlsrv16-Option -ppv

Bewirkt, dass MobiLink neue periodische Überwachungswerte entsprechend dem festgelegten Zeitraum ausgibt. Zeiträume werden in Sekunden angegeben.

### Syntax

```
mlsrv16 -c "connection-string" -ppv period ...
```

### Bemerkungen

Diese Werte können Informationen über den Status des Servers bereitstellen und dienen dazu, den Zustand und die Performance des MobiLink-Servers zu ermitteln. Mithilfe der Werte DB\_CONNECTIONS und LONGEST\_DB\_WAIT können z.B. potenzielle Probleme der Option -w oder in den Synchronisationsskripten festgestellt werden. Die Werte bieten auch eine einfache Möglichkeit, Datendurchsatzraten systemweit zu protokollieren, z.B. die Anzahl von Upload- oder Downloadzeilen pro Sekunde und die Anzahl der erfolgreichen Synchronisationen pro Sekunde.

Der empfohlene Zeitraum ist 60 Sekunden.

Wenn der Zeitraum zu klein ist, wächst das Log sehr schnell an.

Jeder Ausgabezeile wird **PERIODIC:** vorangestellt, um die Suche und das Filtern von Werten zu erleichtern.

Die ausgegebenen Werte können folgende Informationen enthalten:

| Ausgegebener Wert       | Beschreibung                                                                                                     |
|-------------------------|------------------------------------------------------------------------------------------------------------------|
| CMD_PROCESSOR_STAGE_LEN | Die Länge der Warteschlange für Synchronisationsaufgaben.                                                        |
| CPU_USAGE               | Der Umfang der vom MobiLink-Server verwendeten CPU-Zeit in Mikrosekunden.                                        |
| DB_CONNECTIONS          | Die Anzahl der aktuell verwendeten Datenbankverbindungen.                                                        |
| FREE_DISK_SPACE         | Der auf dem temporären Plattenspeicher verfügbare Speicherplatz in Byte.                                         |
| HEARTBEAT_STAGE_LEN     | Die Länge der Warteschlange für periodische Aufgaben, die nicht im Rahmen der Synchronisation ausgeführt werden. |
| LONGEST_DB_WAIT         | Der längste Zeitraum, den eine aktive Synchronisation auf die Datenbank warten musste.                           |
| LONGEST_SYNC            | Das Alter der ältesten Synchronisation in Mikrosekunden.                                                         |

| Ausgegebenener Wert      | Beschreibung                                                                                            |
|--------------------------|---------------------------------------------------------------------------------------------------------|
| MEMORY_USED              | Die Anzahl der vom RAM-Speicher verwendeten Byte (nur für Windows).                                     |
| ML_NUM_CONNECTED_CLIENTS | Die Anzahl von verbundenen Synchronisationsclients.                                                     |
| NOTIFIER_STAGE_LEN       | Die Länge der Notifier-Arbeitswarteschlange.                                                            |
| NUM_COMMITS              | Die Gesamtzahl der Festschreibungen.                                                                    |
| NUM_CONNECTED_FILE_XFERS | Die Anzahl der derzeitigen MobiLink-Dateiübertragungen.                                                 |
| NUM_CONNECTED_LISTENERS  | Die Anzahl der derzeit verbundenen Listener.                                                            |
| NUM_CONNECTED_MONITORS   | Die Anzahl der derzeit verbundenen Monitore.                                                            |
| NUM_CONNECTED_PINGS      | Die Anzahl der derzeit verbundenen Clients, die einen Ping senden.                                      |
| NUM_CONNECTED_SYNCs      | Die Anzahl der derzeit verbundenen Datensynchronisationen.                                              |
| NUM_ERRORS               | Die Gesamtzahl der Fehler.                                                                              |
| NUM_FAILED_SYNCs         | Die Gesamtzahl der fehlgeschlagenen Synchronisationen.                                                  |
| NUM_IN_APPLY_UPLOAD      | Die Anzahl der Synchronisationen, die derzeit in der Phase der Upload-Übernahme sind.                   |
| NUM_IN_AUTH_USER         | Die Anzahl der Synchronisationen, die derzeit in der Phase der Benutzerauthentifizierung sind.          |
| NUM_IN_BEGIN_SYNC        | Die Anzahl der Synchronisationen, die derzeit in der Phase des Synchronisationsbeginns sind.            |
| NUM_IN_CONNECT           | Die Anzahl der Synchronisationen, die derzeit in der Verbindungsphase sind.                             |
| NUM_IN_CONNECT_FOR_ACK   | Die Anzahl der Synchronisationen, die derzeit in der Verbindungsphase für die Downloadquittierung sind. |
| NUM_IN_END_SYNC          | Die Anzahl der Synchronisationen, die derzeit in der Synchronisationsende-Phase sind.                   |
| NUM_IN_FETCH_DNLD        | Die Anzahl der Synchronisationen, die derzeit in der Phase des Download-Abrufs sind.                    |

| Ausgegebener Wert            | Beschreibung                                                                                                                                          |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| NUM_IN_GET_DB_WORKER_FOR_ACK | Die Anzahl der Synchronisationen, die derzeit auf eine Datenbankverbindung für die Verarbeitung einer nicht blockierenden Downloadbestätigung warten. |
| NUM_IN_NON_BLOCKING_ACK      | Die Anzahl von Synchronisationen, die derzeit in der Phase der nicht blockierenden Downloadquittierung sind.                                          |
| NUM_IN_PREP_FOR_DNLD         | Die Anzahl der Synchronisationen, die derzeit in der Phase der Downloadvorbereitung sind.                                                             |
| NUM_IN_RECVING_UPLOAD        | Die Anzahl der Synchronisationen, die derzeit in der Phase des Uploadempfangs sind.                                                                   |
| NUM_IN_SEND_DNLD             | Die Anzahl der Synchronisationen, die derzeit in der Downloadsendephase sind.                                                                         |
| NUM_IN_SYNC_REQUEST          | Die Anzahl der Synchronisationen, die derzeit in der Phase der Synchronisationsanforderung sind.                                                      |
| NUM_IN_WAIT_FOR_DNLD_ACK     | Die Anzahl der Synchronisationen, die derzeit in der Wartephase für die Downloadquittierung sind.                                                     |
| NUM_ROLLBACKS                | Die Gesamtzahl der Rollbacks.                                                                                                                         |
| NUM_ROWS_DOWNLOADED          | Die Gesamtzahl der an die entfernten Datenbanken gesendeten Zeilen.                                                                                   |
| NUM_ROWS_UPLOADED            | Die Gesamtzahl der von den entfernten Datenbanken empfangenen Zeilen.                                                                                 |
| NUM_SUCCESS_SYNCs            | Die Gesamtzahl der erfolgreichen Synchronisationen.                                                                                                   |
| NUM_UPLOAD_CONNS_IN_USE      | Die Anzahl der derzeit verwendeten Uploadverbindungen.                                                                                                |
| NUM_WAITING_CONS             | Die Anzahl der Synchronisationen, die derzeit auf die konsolidierte Datenbank warten.                                                                 |
| NUM_WARNINGS                 | Die Gesamtzahl der Warnungen.                                                                                                                         |
| OE_STAGE_LEN                 | Die Länge der Arbeitswarteschlange für den integrierten Outbound Enabler.                                                                             |
| PAGES_IN_STREAMSTACK         | Die Anzahl der Seiten, die in den Netzwerkdatenströmen enthalten sind.                                                                                |

| Ausgegebenener Wert    | Beschreibung                                                                                                                                                                                                                                                                    |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PAGES_LOCKED           | Die Anzahl der Cacheseiten, die in den Speicher geladen wurden.                                                                                                                                                                                                                 |
| PAGES_LOCKED_MAX       | Die Anzahl der Seiten im Cachespeicher.                                                                                                                                                                                                                                         |
| PAGES_SWAPPED_IN       | Die Gesamtzahl der vom Plattenspeicher gelesenen Seiten.                                                                                                                                                                                                                        |
| PAGES_SWAPPED_OUT      | Die Gesamtzahl der auf den Plattenspeicher ausgelagerten Seiten.                                                                                                                                                                                                                |
| PAGES_USED             | Die Anzahl der verwendeten Cacheseiten. Diese umfasst auch die Seiten, die auf den Plattenspeicher ausgelagert werden, sodass sie die Cachegröße überschreiten kann.                                                                                                            |
| PRIMARY_IS_KNOWN       | Gibt an, ob der Primärserver bekannt ist oder nicht. Der Wert 0 gibt an, dass es für den Server gleichgültig ist, welcher der primäre Server ist. Der Wert 1 gibt an, dass der Server den Primärserver kennt. Der Wert 2 gibt an, dass der Server den Primärserver nicht kennt. |
| RAW_TCP_STAGE_LEN      | Die Länge der Warteschlange für Netzwerkaufgaben.                                                                                                                                                                                                                               |
| SERVER_IS_PRIMARY      | Gibt an, ob der Server ein Primär- oder Sekundärserver ist. Der Wert 1 gibt an, dass der Server ein Primärserver ist. Andernfalls ist der Wert 0.                                                                                                                               |
| SIRT_NUM_LWP_HITS      | Die Anzahl der einfachen Abrufe von Agenten für entfernte Aufgaben, die auf eine Benachrichtigung hinweisen.                                                                                                                                                                    |
| SIRT_NUM_LWPS          | Die Anzahl der einfachen Abrufe von Agenten für entfernte Aufgaben.                                                                                                                                                                                                             |
| SIRT_NUM_REQUESTS      | Die Anzahl der derzeit noch ausstehenden Benachrichtigungen zu entfernten Aufgaben.                                                                                                                                                                                             |
| STREAM_STAGE_LEN       | Die Länge der Warteschlange für Netzwerkaufgaben einer hohen Ebene.                                                                                                                                                                                                             |
| TCP_BYTES_READ         | Die Gesamtzahl der jemals gelesenen Byte.                                                                                                                                                                                                                                       |
| TCP_BYTES_WRITTEN      | Die Gesamtzahl der jemals geschriebenen Byte.                                                                                                                                                                                                                                   |
| TCP_CONNECTIONS        | Die Anzahl der derzeit geöffneten TCP-Verbindungen.                                                                                                                                                                                                                             |
| TCP_CONNECTIONS_CLOSED | Die Gesamtzahl der jemals geschlossenen Verbindungen.                                                                                                                                                                                                                           |

| Ausgegebener Wert        | Beschreibung                                                                                                                                                                                                                                                  |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TCP_CONNECTIONS_OPENED   | Die Gesamtzahl der jemals geöffneten Verbindungen.                                                                                                                                                                                                            |
| TCP_CONNECTIONS_REJECTED | Die Gesamtzahl der jemals abgelehnten Verbindungen.                                                                                                                                                                                                           |
| TIMED_WORK_STAGE_LEN     | Die Länge der Arbeitswarteschlange für das dynamische Caching.                                                                                                                                                                                                |
| TRACKED_MEMORY           | Der vom Server belegte Speicherplatz. Verwenden Sie diese Metrik für Nicht-Windows-Systeme, in denen die MEMORY_USED-Metrik nicht zur Verfügung steht. Auf Microsoft Windows-Systemen können Sie mit der MEMORY_USED-Metrik eine höhere Genauigkeit erzielen. |
| VM_MEM_USE               | Der von zugeordneten VMs genutzte Speicherplatz.                                                                                                                                                                                                              |

### Beispiel

Die folgende Beispielausgabe enthält die periodischen Überwachungswerte.

```

I. 2009-10-28 11:46:29. <Main> PERIODIC: TCP_CONNECTIONS: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: PAGES_USED: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: PAGES_LOCKED: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: PAGES_LOCKED_MAX: 13243
I. 2009-10-28 11:46:29. <Main> PERIODIC: TCP_CONNECTIONS_OPENED: 2
I. 2009-10-28 11:46:29. <Main> PERIODIC: TCP_CONNECTIONS_CLOSED: 2
I. 2009-10-28 11:46:29. <Main> PERIODIC: TCP_CONNECTIONS_REJECTED: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: TCP_BYTES_READ: 5137
I. 2009-10-28 11:46:29. <Main> PERIODIC: TCP_BYTES_WRITTEN: 4549
I. 2009-10-28 11:46:29. <Main> PERIODIC: ML_NUM_CONNECTED_CLIENTS: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: PAGES_SWAPPED_OUT: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: PAGES_SWAPPED_IN: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: PAGES_IN_STREAMSTACK: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: CPU_USAGE: 3359375
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_COMMITS: 7
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_ROLLBACKS: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_SUCCESS_SYNCNS: 1
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_FAILED_SYNCNS: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_ERRORS: 2
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_WARNINGS: 3
I. 2009-10-28 11:46:29. <Main> PERIODIC: DB_CONNECTIONS: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: RAW_TCP_STAGE_LEN: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: STREAM_STAGE_LEN: 5
I. 2009-10-28 11:46:29. <Main> PERIODIC: HEARTBEAT_STAGE_LEN: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: CMD_PROCESSOR_STAGE_LEN: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_ROWS_DOWNLOADED: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_ROWS_UPLOADED: 7
I. 2009-10-28 11:46:29. <Main> PERIODIC: FREE_DISK_SPACE: 124154904576
I. 2009-10-28 11:46:29. <Main> PERIODIC: LONGEST_DB_WAIT: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: LONGEST_SYNC: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: MEMORY_USED: 140275712
I. 2009-10-28 11:46:29. <Main> PERIODIC: SERVER_IS_PRIMARY: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_CONNECTED_SYNCNS: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_CONNECTED_PINGS: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_CONNECTED_FILE_XFERS: 0

```

```

I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_CONNECTED_MONITORS: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_CONNECTED_LISTENERS: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_WAITING_CONS: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_IN_SYNC_REQUEST: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_IN_RECVING_UPLOAD: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_IN_CONNECT: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_IN_AUTH_USER: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_IN_BEGIN_SYNC: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_IN_APPLY_UPLOAD: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_IN_PREP_FOR_DNLD: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_IN_FETCH_DNLD: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_IN_END_SYNC: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_IN_SEND_DNLD: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_IN_WAIT_FOR_DNLD_ACK: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_IN_GET_DB_WORKER_FOR_ACK: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_IN_CONNECT_FOR_ACK: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_IN_NON_BLOCKING_ACK: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NUM_UPLOAD_CONNS_IN_USE: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: TRACKED_MEMORY: 56269577
I. 2009-10-28 11:46:29. <Main> PERIODIC: VM_MEM_USE: 517013504
I. 2009-10-28 11:46:29. <Main> PERIODIC: TIMED_WORK_STAGE_LEN: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: NOTIFIER_STAGE_LEN: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: SIRT_NUM_REQUESTS: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: SIRT_NUM_LWPS: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: SIRT_NUM_LWP_HITS: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: OE_STAGE_LEN: 0
I. 2009-10-28 11:46:29. <Main> PERIODIC: PRIMARY_IS_KNOWN: 0

```

## mlsrv16-Option -q

Weist MobiLink an, mit einem minimierten Meldungsfenster zu starten.

### Syntax

```
mlsrv16 -c "connection-string" -q ...
```

### Bemerkungen

Minimiert das Meldungsfenster des MobiLink-Servers.

## mlsrv16-Option -r

Legt die maximale Anzahl von Deadlock-Neuversuchen fest.

### Syntax

```
mlsrv16 -c "connection-string" -r retries ...
```

### Bemerkungen

Standardmäßig unternimmt der MobiLink-Server bei Uploads, die durch einen Deadlock in der konsolidierten Datenbank blockiert sind, bis zu 10 Versuche. Wenn der Deadlock nicht unterbrochen wird, schlägt die Synchronisation fehl, da nicht garantiert ist, dass der Deadlock überwunden werden kann. Mit dieser Option können Sie eine beliebige Wiederholungsgrenze festlegen. Wenn der Server bei Transaktions-Deadlocks nicht versuchen soll, die Transaktion auszuführen, geben Sie **-r 0** ein. Die Obergrenze für diese Einstellung ist 2 hoch 32 minus eins.

**Hinweis**

Deadlocks dürfen nicht Teil eines normalen Synchronisationssystems sein. Wenn sie auftreten, entfernen Sie sie, indem Sie die Synchronisationsskripten korrigieren.

## mlsrv16-Option -rd

Legt die maximale Zeitverzögerung zwischen Deadlock-Neuversuchen fest.

**Syntax**

```
mlsrv16 -c "connection-string" -rd delay ...
```

**Bemerkungen**

Wenn Uploadtransaktionen durch einen Deadlock in der konsolidierten Datenbank blockiert sind, wartet der MobiLink-Server eine zufällig gewählte Zeitspanne ab, bevor er die betreffende Transaktion wiederholt. Durch die zufällige Verzögerung steigt die Wahrscheinlichkeit, dass zukünftige Versuche erfolgreich sind. Mit dieser Option können Sie die maximale Verzögerung in Sekunden angeben. Der Wert 0 (Null) hat sofortige Wiederholungen zur Folge, höhere Werte sind aber empfehlenswert, da sie erfolgreichere Wiederholungsversuche bewirken. Der Standardwert und Maximalverzögerungswert ist **30**.

**Hinweis**

Deadlocks dürfen nicht Teil eines normalen Synchronisationssystems sein. Wenn sie auftreten, entfernen Sie sie, indem Sie die Synchronisationsskripten korrigieren.

## mlsrv16-Option -rrp

Bewirkt, dass der MobiLink-Server beim Serverstart das mlreplay-Dienstprogramm ausführt und alle aufgezeichneten Sitzungen (Dateien mit der Erweiterung *mlr*) im angegebenen Verzeichnis wiedergibt.

Verwenden Sie diese Option, um entfernte Schemata vorab in den MobiLink-Server zu laden. Dies erspart die Zeit und den Aufwand für das Senden des entfernten Schemas bei der anfänglichen Synchronisation entfernter Datenbanken.

**Syntax**

```
mlsrv16 -c "connection-string" -rrp directory ...
```

**Bemerkungen**

Um die Option -rrp verwenden zu können, muss eine lokale Server-Verbindungszeichenfolge mithilfe der Option -lsc angegeben werden, damit das mlreplay-Dienstprogramm eine Verbindung zum Server herstellen kann.

Die Optionen -rrp und -rp verwenden

- Zeichnen Sie Synchronisationen mit der Option -rp auf.

- Ermitteln Sie, welche vorher aufgezeichneten Synchronisationen für das Vorabladen des Schemas zu verwenden sind. Für jedes Schema und/oder jede Gruppe von Publikationen muss eine vorhanden sein.
- Kopieren Sie die vorher aufgezeichneten Synchronisationen in ein neues Verzeichnis.
- Führen Sie sie in einer Produktionsumgebung ohne die Option -rp und mit der Option-rrp aus.

**Siehe auch**

- „mlsrv16-Option -rp“ auf Seite 77
- „mlsrv16-Option -lsc“ auf Seite 65
- „Dienstprogramm MobiLink Replay (mlreplay)“ auf Seite 702

## mlsrv16-Option -rp

Gibt das Verzeichnis an, in dem Synchronisationen für die Wiedergabe mit dem Dienstprogramm mlreplay aufgezeichnet werden.

**Syntax**

**mlsrv16 -c** "*connection-string*" **-rp** *directory* ...

**Bemerkungen**

Um eine optimale Performance zu erzielen, verwenden Sie diese Option zum Aufzeichnen von Synchronisationen, die mit der Option -rrp benutzt werden sollen. Die Option -rrp ermöglicht, dass alle Synchronisationen, einschließlich der ersten Synchronisation eines jeden eindeutigen Schemas, den Schema-Cache nutzen.

- Zeichnen Sie Synchronisationen mit der Option -rp auf.
- Ermitteln Sie, welche vorher aufgezeichneten Synchronisationen für das Vorabladen des Schemas zu verwenden sind. Für jedes Schema und/oder jede Gruppe von Publikationen muss eine vorhanden sein.
- Kopieren Sie die vorher aufgezeichneten Synchronisationen in ein neues Verzeichnis.
- Führen Sie sie in einer Produktionsumgebung ohne die Option -rp und mit der Option-rrp aus.

**Siehe auch**

- „mlsrv16-Option -rrp“ auf Seite 76
- „Dienstprogramm MobiLink Replay (mlreplay)“ auf Seite 702

## mlsrv16-Option -s

Legt die maximale Anzahl von Zeilen fest, die gleichzeitig gesendet werden können.

**Syntax**

**mlsrv16 -c** "*connection-string*" **-s** *count* ...

**Bemerkungen**

Legt die maximale Anzahl der Zeilen fest, die gleichzeitig abgerufen, eingefügt, aktualisiert oder gelöscht werden können.

Der MobiLink-Server sendet Uploadzeilen über den ODBC-Treiber an die konsolidierte Datenbank. Diese Option steuert die Anzahl der Zeilen, die pro Batch an den Datenbankserver gesendet werden. Durch Erhöhen dieses Wertes wird die Verarbeitung des Uploaddatenstroms beschleunigt und die Netzwerkzeit verkürzt. Mit einer höheren Einstellung erfordert der MobiLink-Server jedoch unter Umständen mehr Ressourcen, um den Uploaddatenstrom anzuwenden.

Die Anzahl gleichzeitiger Uploadzeilen kann in der Logdatei als Zeilenmengengröße (**rowset size**) angezeigt werden.

Standardwert ist "10".

**mlsrv16-Option -sl dnet**

Legt die .NET CLR-Optionen (Common Language Runtime) fest und erzwingt, dass CLR beim Hochfahren geladen wird. Diese Option wird bei der Verwendung von .NET-Skriptenlogik empfohlen.

**Syntax**

**mlsrv16 -c "connection-string" -sl dnet options ...**

**Bemerkungen**

Legt Optionen fest, die direkt an NET CLR übergeben werden sollen.

Die folgenden Optionen stehen zur Verfügung:

| Option                       | Beschreibung                                                                                                                                                                                                                                                                                                                                               |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-Dname=value</b>          | Umgebungsvariable festlegen. Beispiel:<br><br><code>-Dsynchtype=far -Dextra_rows=yes</code><br><br>Weitere Hinweise finden Sie unter der .NET-Frameworkklasse System.Environment.                                                                                                                                                                          |
| <b>-MLAutoLoadPath= path</b> | Legen Sie den Speicherort der Basis-Assemblys fest. Funktionierte nur mit privaten Assemblys. Verwenden Sie diese Option oder -MLDomConfigFile, aber nicht beide, um MobiLink anzugeben, wo sich Assemblys befinden. Wenn Sie -MLAutoLoadPath verwenden, können Sie im Ereignisskript keine Domäne angeben. Der Standardwert ist das aktuelle Verzeichnis. |

| Option                                       | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-MLDomConfigFile=</b> <i>file</i>         | <p>Legen Sie den Speicherort der Basis-Assemblys fest. Verwenden Sie diese Option, wenn Sie gemeinsam genutzte Assemblys haben und nicht alle Assemblys in dem Verzeichnis laden möchten oder wenn MLAutoLoadPath aus einem anderen Grund nicht verwendet werden kann. Verwenden Sie -MLDomConfigFile oder -MLAutoLoadPath, aber nicht beide, um MobiLink anzugeben, wo sich Assemblys befinden.</p> <p>Wenn sich der in der Option -MLDomConfigFile referenzierte Dateipfad in der Verwendung auf eine Datei in einem Ordner bezieht, dessen Name eine Leerstelle enthält (z.B. "C:\Programme\Meine Firma\SyncServer\MLDomConfig.xml"), setzen Sie die gesamte Option in Anführungszeichen: "-MLDomConfigFile=C:\Programme\Meine Firma\SyncServer\MLDomConfig.xml".</p> |
| <b>-MLStartClasses=</b><br><i>classnames</i> | Lädt und instanziiert beim Start des Servers die benutzerdefinierten Startklassen in der angegebenen Reihenfolge.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>-clrConGC</b>                             | Gleichzeitige Garbage Collection in CLR aktivieren.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>-clrFlavor=( wks   svr )</b>              | Variante der zu ladenden .NET CLR. Die Subvariante ist <b>svr</b> für Server und <b>wks</b> für Workstations. Standardmäßig wird <b>svr</b> geladen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>-clrVersion=</b> <i>version</i>           | Version der zu ladenden .NET CLR. Das Präfix muss mit <b>v</b> gesetzt werden. Beispiel: <b>v1.0.3705</b> lädt das Verzeichnis <code>\Microsoft.NET\Framework\v1.0.3705</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

Um v4.0-Assemblys verwenden zu können, müssen Sie explizit die Option -clrVersion hinzufügen, damit der MobiLink-Server eine v4.0-Laufzeitumgebung lädt. Zum Beispiel, -clrVersion=v4.0.30319.

Diese Optionsliste kann mit folgendem Befehl aufgerufen werden:

```
mldrsv16 -sl dnet (?)
```

#### Siehe auch

- „Synchronisationsskripten in .NET“ auf Seite 536

## mldrsv16-Option -sl java

Legt die Optionen der Java VM fest und erzwingt das Laden der Java VM beim Start. Diese Option wird bei der Verwendung von Java-Skriptenlogik empfohlen. Unter Unix müssen die Optionen -cp durch Doppelpunkte voneinander abgetrennt sein.

**Syntax**

```
mllsrv16 -c "connection-string" -sl java ( options ) ...
```

**Bemerkungen**

Setzt -jrepath und andere Optionen, die direkt an die Java VM übergeben werden sollen. Folgende Optionen sind verfügbar:

| Option                                                       | Beschreibung                                                                                                                                                                                                                |
|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-classic</b>                                              | Verwenden Sie die klassische Java VM.                                                                                                                                                                                       |
| <b>-client</b>                                               | Verwenden Sie den Java VM-Client.                                                                                                                                                                                           |
| <b>-hotspot</b>                                              | Verwenden Sie die Java Hotspot VM.                                                                                                                                                                                          |
| <b>-server</b>                                               | Verwenden Sie die Java Server VM. Dies ist die Standardeinstellung.                                                                                                                                                         |
| <b>-cp</b> <i>location</i> ; <i>...</i>                      | Gibt eine Reihe von Verzeichnissen oder JAR-Dateien an, in denen nach Klassen gesucht werden soll. Sie können auch -classpath verwenden.                                                                                    |
| <b>-D</b> <i>name=value</i>                                  | Legt eine Systemeigenschaft fest. Beispiel:<br><br><i>-Dsynchtype=far -Dextra_rows=yes</i>                                                                                                                                  |
| <b>-DMLStartClasses</b> = <i>classname</i> , ...             | Lädt und instanziert beim Start des Servers die benutzerdefinierten Startklassen in der angegebenen Reihenfolge.                                                                                                            |
| <b>-jrepath</b> <i>path</i>                                  | Heben Sie den JRE-Standardpfad auf, bei dem es sich um eines der folgenden Verzeichnisse handelt:<br><i>%SQLANY16%\Bin32\jre170</i> oder <i>%SQLANY16%\Bin64\jre170</i> (für 32-Bit- bzw. 64-Bit-Plattformen).              |
| <b>-verbose</b> [ <i>:class</i>   <i>:gc</i>   <i>:jni</i> ] | Aktiviert ausführliche Ausgabe.                                                                                                                                                                                             |
| <b>-X</b> <i>vm-option</i>                                   | Geben Sie eine VM-spezifische Option an, wie in der Datei <i>%SQLANY16%\Bin32\jre170\bin\client\Xusage.txt</i> oder <i>%SQLANY16%\Bin64\jre170\bin\server\Xusage.txt</i> (für 32-Bit- bzw. 64-Bit-Plattformen) beschrieben. |

Um eine Liste mit möglichen Java-Optionen anzuzeigen, geben Sie Folgendes ein:

```
java
```

## Unix-Hinweise

Optionen müssen in Klammern gesetzt werden. Hierfür können Sie runde, wie in der Syntax oben, oder geschweifte { } Klammern verwenden.

Die Option -jrepath ist nur unter Windows verfügbar. Wenn Sie unter Unix eine bestimmte JRE laden möchten, legen Sie LD\_LIBRARY\_PATH (LIBPATH unter IBM AIX, SHLIB\_PATH unter HP-UX) so fest, dass das Verzeichnis mit der JRE einbezogen wird. Das Verzeichnis muss vor allen SQL Anywhere-Installationsverzeichnissen aufgeführt sein.

## Siehe auch

- „Schreiben eines Synchronisationsskripts in Java“ auf Seite 463

## Beispiele

Mit dem folgenden Ausschnitt aus einer mlsrv16-Befehlszeile wird unter Windows beispielsweise die Option der Java VM festgelegt, die Systemassertierungen aktiviert:

```
mlsrv16 -sl java (-cp ;\myclasses; -esa) ...
```

Der folgende Ausschnitt aus einer mlsrv16-Befehlszeile legt unter Windows die Systemeigenschaft LDAP\_SERVER fest:

```
mlsrv16 -sl java ( -cp ;\myclasses; -DLdap_SERVER=mycorp-ldap ) ...
```

Der folgende Teil einer mlsrv16-Befehlszeile gilt für Unix:

```
mlsrv16 -sl java { -cp .:$CLASSPATH:/opt/myclasses:/opt/my.jar: }
```

## mlsrv16-Option -sm

Legt durch Beschränkung der maximalen Anzahl von Netzwerkverbindungen die maximale Anzahl von Synchronisationen fest, die aktiv bearbeitet werden können.

## Syntax

```
mlsrv16 -c "connection-string" -sm number ...
```

## Bemerkungen

Der MobiLink-Server führt die nachfolgenden Synchronisationsaufgaben gleichzeitig aus:

1. Lesen der Uploaddaten aus dem Netzwerk und Entpacken der Daten.
2. Anwenden der Uploads auf die konsolidierte Datenbank.
3. Abrufen der Daten für den Download aus der konsolidierten Datenbank.
4. Packen der Downloaddaten und Senden an entfernte Datenbanken.

Die Anzahl der Synchronisationen für jede Aufgabe beschränkt sich wie folgt:

- Die Anzahl der Synchronisationen, die die Aufgaben 2 und 3 ausführen, ist kleiner/gleich der Einstellung für die mlsrv16-Option -w.

- Die Anzahl der Synchronisationen, die die Aufgabe 2 ausführen, ist kleiner/gleich der Einstellung für die `mlsrv16-Option -wu`.
- Die Anzahl der Synchronisationen, die alle vier Aufgaben ausführen, ist kleiner als oder gleich der Einstellung für die Option `-sm`.

Höhere Werte für `-sm`, besonders, wenn sie viel größer sind als `-w`, ermöglichen es dem MobiLink-Server, mehr Netzwerkaufgaben (1 und 4) als Datenbankaufgaben (2 und 3) durchzuführen. Dies hilft sicherzustellen, dass ein Datenbank-Worker nicht auf Aufgaben warten muss, wenn es sonst zu Engpässen bei der Netzwerk-Performance kommen würde. Dies kann den Durchsatz verbessern. Wenn der Wert für `-sm` jedoch zu hoch angesetzt wird und genügend gleichzeitige Verbindungen vorhanden sind, kann der MobiLink-Server mehr Speicher zuweisen als direkt verfügbar ist. Dies kann das Virtual Memory Paging des Betriebssystems aktivieren, was wiederum zur Folge hat, dass Speicherkapazität auf die Festplatte ausgelagert wird, wodurch sich der Durchsatz merklich reduziert.

### Siehe auch

- „`mlsrv16-Option -w`“ auf Seite 90
- „`mlsrv16-Option -wu`“ auf Seite 92
- „`mlsrv16-Option -nc`“ auf Seite 65

## mlsrv16-Option -wn

Legt die Anzahl von Netzwerk-Worker-Threads fest, die der MobiLink-Server für die gleichzeitige Verarbeitung von Netzwerkdatenströmen verwendet.

### Syntax

```
mlsrv16 -c "connection-string" -wn count ...
```

### Bemerkungen

Der Standardwert ist 1.

Der Einsatz mehrerer Netzwerk-Worker-Threads kann die Performance verbessern, besonders wenn CPU-intensive Netzwerk-Datenstromoptionen wie Verschlüsselung oder Komprimierung zusammen mit entweder großen Synchronisationen oder zahlreichen kleinen Synchronisationen verwendet werden. Jede Anforderung im System kann auf maximal einem Netzwerkdatenstrom-Thread aktiv sein.

## mlsrv16-Option -tc

Setzt einen Schwellenwert für den Timeout bei lang laufenden SQL-Skripten.

### Syntax

```
mlsrv16 -c "connection string" -tc minutes ...
```

### Bemerkungen

Standardmäßig beobachtet der MobiLink-Server die Ausführungszeit jedes SQL-Skripts und gibt eine Warnung aus, wenn die Ausführungszeit des Skripts 10 Minuten erreicht. Lang laufende Skripten

verursachen mit höherer Wahrscheinlichkeit Konflikte und Blockierungen in der konsolidierten Datenbank. Dadurch kann sich der allgemeine Durchsatz erheblich verringern.

Sie können mit der Option `-tf` Anweisungen stornieren, die größer sind als der Schwellenwert.

Der Standardwert kann auf Null oder eine positive Ganzzahl zurückgesetzt werden, deren Einheit Minuten sind. Wenn der Wert auf Null gesetzt wird, wird die Option `-tc` deaktiviert und der MobiLink-Server überwacht die Skriptaussführung nicht.

Wenn der Schwellenwert für den Timeout ein anderer Wert als Null ist, zeigt der MobiLink-Server die Warnung exponentiell an. Die Warnung wird angezeigt, wenn die Ausführungszeit die angegebene Zeit erstmals überschreitet. Sie wird erneut angezeigt, wenn die Ausführungszeit die angegebene Zeit 2 Mal überschreitet, dann bei 4 Mal usw.

Die Warnung enthält die für die laufende Synchronisation verwendete Verbindungs-ID und den Kontext, der, sofern verfügbar, Folgendes enthält: entfernte ID, ML-Benutzername, geänderter Benutzername, Transaktion, Tabellename, Zeilenwert und Skriptversion. Der Kontext der Timeoutwarnung wird unabhängig von den Einstellungen zur Ausführlichkeit des MobiLink-Servers angezeigt.

Wenn die konsolidierte Datenbank auf einem Oracle-Datenbankserver ausgeführt wird und die Timeoutwarnung angezeigt wird, muss möglicherweise ein Datenbankbenutzer mit DBA-Berechtigung die konsolidierte Datenbank überprüfen, um die Ursache des Problems zu ermitteln. Der von der Synchronisation verwendete ServiceName und SERIAL# sind in der Warnmeldung enthalten. Wenn die Synchronisationsverbindung gestoppt wird, beendet der MobiLink-Server die laufende Synchronisation.

#### Siehe auch

- „[mlsrv16-Option -tf](#)“ auf Seite 83

## mlsrv16-Option -tf

Diese Option wird verwendet, damit das SQL-Skript auf dem MobiLink-Server fehlschlägt, wenn die Ausführungszeit den mit `-tc` festgelegten Timeoutwert überschreitet. Diese Option ist nicht verfügbar, wenn die konsolidierte Datenbank auf einem Oracle-Server ausgeführt wird.

#### Syntax

**mlsrv16 -c "connection string" -tf ...**

#### Bemerkungen

Wenn das SQL-Skript fehlschlägt, bricht der MobiLink-Server die Synchronisation ab, oder er überspringt die Zeile (wenn das Skript ein Uploadskript ist und das Skript `handle_error 1000` zurückgibt) und setzt die Synchronisation fort.

Der MobiLink-Server zeigt eine Warnung an, wenn diese Option angegeben wird und die Datenbank auf einem Oracle-Server ausgeführt wird.

Diese Option wird ignoriert, wenn **-tc 0** festgelegt wurde.

## mlsrv16-Option -ts

Richtet eine Protokollierungssitzung für den MobiLink-Server ein.

### Syntax

**mlsrv16 -c "connection-string" -ts session-name(session-option=[option-value;...])**

Der Sitzungsname muss **logging** lauten.

| Sitzungsoption | Optionswert                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------|
| <b>events</b>  | Kommagetrennte Liste der System-Trace-Ereignisse. Die unterstützten Ereignisse sind "Info", "Warning" und "Error". |
| <b>targets</b> | <i>target-type(target-option=value[;...])</i> wobei <i>target-type</i> nur <b>file</b> sein kann.                  |

Die Zieloptionen werden als Name-Wert-Paare angegeben. Die Zieldatei kann die folgenden Optionen aufweisen:

| Zieloptionsname        | Erwarteter Wert      | Beschreibung                                                                                                                                                                                                                                                                                                          |
|------------------------|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>filename_prefix</b> | Zeichenfolge         | Ein ETD-Dateinamenpräfix mit oder ohne Pfad. Alle ETD-Dateien haben die Erweiterung <i>.etd</i> . Dieser Parameter ist erforderlich.                                                                                                                                                                                  |
| <b>max_size</b>        | Ganzzahl             | Die maximale Größe der Datei in Byte. Der Standardwert ist 0, was bedeutet, dass es keine Grenze für die Dateigröße gibt und die Datei größer wird, solange Speicherplatz verfügbar ist. Sobald die angegebene Größe erreicht ist, wird eine neue Datei gestartet.                                                    |
| <b>num_files</b>       | Ganzzahl             | Die Anzahl der Dateien, in die Informationen zur Ereignisprotokollierung geschrieben werden. Diese Option wird nur verwendet, wenn <i>max_size</i> eingestellt ist. Wenn alle Dateien die maximale angegebene Größe erreichen, beginnt der MobiLink-Server, die älteste Datei zu überschreiben.                       |
| <b>flush_on_write</b>  | YES, TRUE, NO, FALSE | Ein Wert, der steuert, ob Festplattenpuffer für jedes protokollierte Ereignis geleert werden. Die Werte YES, TRUE, NO und FALSE werden akzeptiert. Der Standardwert ist FALSE. Bei aktiviertem Parameter kann die Performance des MobiLink-Servers vermindert sein, wenn viele Trace-Ereignisse protokolliert werden. |
| <b>compressed</b>      | YES, TRUE, NO, FALSE | Ein Wert, der die Komprimierung der ETD-Datei zum Einsparen von Speicherplatz steuert. Der Standardwert ist FALSE.                                                                                                                                                                                                    |

## Bemerkungen

Alle Daten, die nach dem Teil `-ts logging` der Option angegeben werden, müssen ohne Leerzeichen angegeben werden.

## Siehe auch

- „Dienstprogramm für die Verwaltung von ETD-Dateien (dbmanageetd)“ [[SQL Anywhere Server - Datenbankadministration](#)]

## Beispiel

Im Folgenden finden Sie ein Beispiel der Option `-ts`:

```
-ts  
logging{events=Info,warning,Error;targets=file{filename_prefix=mls_etd;max_size=10000000;num_files=10;flush_on_write=true}}
```

## mlsrv16-Option -tx

Bei der Verwendung von transaktionalen Uploads gruppiert diese Option Transaktionen und schreibt sie zusammen fest.

## Syntax

```
mlsrv16 -c "connection-string" -tx count ...
```

## Bemerkungen

Verwenden Sie diese Option, um die Performance zu steigern, wenn Sie transaktionale Uploads ausführen.

*count* kann ein beliebiger nicht-negativer Wert sein. Der Standardwert ist 1. Das bedeutet, dass alle Transaktionen separat festgeschrieben werden. Verwenden Sie den Wert Null, um nach dem Upload aller Transaktionen eine Festschreibung durchzuführen.

Der ideale Wert für *count* kann nur durch Testen der Performance ermittelt werden.

## Siehe auch

- „dbmlsync-Option -tu“ [[MobiLink - Clientadministration](#)]

## mlsrv16-Option -ud

Weist MobiLink an, als Daemon abzulaufen.

## Syntax

```
mlsrv16 -c "connection-string" -ud ...
```

## Bemerkungen

Diese Option gilt nur für UNIX-Plattformen.

### Siehe auch

- [„Verwendung des MobiLink-Servers außerhalb der aktuellen Sitzung“ auf Seite 34](#)

## mlsrv16-Option -ui

Startet den MobiLink-Server unter Linux mit X Window-Serverunterstützung im Shell-Modus, wenn keine nutzbare Anzeige verfügbar ist.

### Syntax

```
mlsrv16 -c "connection-string" -ui ...
```

### Bemerkungen

Wenn "-ui" angegeben ist, versucht der Server, eine verwendbare Anzeige zu finden. Wenn keine gefunden wird, z.B. weil der X Window-Server nicht läuft, startet der MobiLink-Server im Shell-Modus.

## mlsrv16-Option -ux

Unter Linux wird das Meldungsfenster von MobiLink-Server geöffnet, in dem Meldungen angezeigt werden.

### Syntax

```
mlsrv16 -c "connection-string" -ux ...
```

### Bemerkungen

Wenn die Option -ux angegeben ist, muss der MobiLink-Server in der Lage sein, eine verwendbare Anzeige zu finden. Wenn er keine findet, weil z.B. die DISPLAY-Umgebungsvariable nicht eingestellt ist oder der X Window-Server nicht läuft, schlägt der Start des MobiLink-Servers fehl.

Um das Meldungsfenster des MobiLink-Servers im stillen Modus auszuführen, verwenden Sie -q.

Unter Windows erscheint das Meldungsfenster des MobiLink-Servers automatisch.

### Siehe auch

- [„mlsrv16-Option -q“ auf Seite 75](#)

## mlsrv16-Option -v

Ermöglicht anzugeben, welche Informationen in der Meldungslogdatei protokolliert sind.

### Syntax

```
mlsrv16 -c "connection-string" -v[ levels ] ...
```

### Bemerkungen

Diese Option steuert den Typ der Meldungen, die in die Meldungslogdatei geschrieben werden.

Wenn Sie nur -v angeben, schreibt der MobiLink-Server minimale Informationen über jede Synchronisation. Je mehr Ebenen angegeben werden, umso ausführlicher wird die Ausgabe in der Meldungslogdatei.

Eine hohe Ausführlichkeitsstufe kann die Performance negativ beeinflussen und sollte nur während der Entwicklung verwendet werden.

Der MobiLink-Server kann so eingerichtet sein, dass für einen Ziel-MobiLink-Benutzer oder eine entfernte ID eine andere Log-Ausführlichkeitsstufe verwendet wird. Der MobiLink-Server prüft die Tabelle ml\_property alle fünf Minuten und sucht die Ausführlichkeitseinstellungen für einen MobiLink-Benutzer oder eine entfernte ID. Siehe [Ausführlichkeitsstufe des Logs für bestimmte MobiLink-Benutzer und entfernte IDs auf Seite 666](#).

Wenn eine CHAR-, VARCHAR-, NCHAR- oder NVARCHAR-Spalte mit einer Byte-Länge von mehr als 32767 Byte synchronisiert wird, zeigt der MobiLink-Server bei hoher Ausführlichkeitsstufe nicht den gesamten Inhalt der Spaltenwerte an. Stattdessen wird der erste Abschnitt der Daten in einer Länge von bis zu 100 Byte angezeigt. Dies gilt für die Ebenen i, q und r.

Die folgenden Ebenen stehen zur Verfügung. Sie können eine oder mehrere dieser Optionen gleichzeitig verwenden, z.B. -vnrsu.

- **+** Aktiviert alle mit Kleinbuchstaben bezeichneten Ausführlichkeitsebenen.
- **c** Zeigt den Inhalt der einzelnen Synchronisationsskripten an, wenn sie aufgerufen werden. Diese Ebene impliziert "s".
- **e** Zeigt Systemereignisskripten an. Mit diesen Systemereignisskripten werden MobiLink-Systemtabellen abgefragt und gepflegt.
- **f** Erst-Lese-Fehler anzeigen. Damit werden Fehler protokolliert, die auftreten, wenn Systeme für die Systemlastverteilung die Serververfügbarkeit prüfen, indem sie Verbindungen herstellen, ohne Daten zu senden, und fehlgeschlagene Synchronisationen verursachen. Verwenden Sie diese Option, um zu überprüfen, ob der Lastverteiler die Verfügbarkeitsprüfungen richtig durchführt.

Siehe auch die TCP/IP-Option **ignore**. Weitere Hinweise finden Sie unter „[mlsrv16-Option -x](#)“ auf Seite 93.

- **h** Zeigt das entfernte Schema, das synchronisiert wird.
- **i** Zeigt den Spaltenwert der einzelnen Upload-Zeilen an. Verwenden Sie diese Option anstelle von -vr, womit die Spaltenwerte der einzelnen Upload- und Downloadzeilen angezeigt werden, um die Menge der protokollierten Daten zu reduzieren. Die Angabe von -vi zusammen mit -vq entspricht der Angabe von -vr.
- **k** Schreibt eine Zeile ins Log, wenn der Cache anwächst oder abnimmt. Damit kann in Belastungstests nach der optimalen Cachegröße gesucht werden. Diese kann dann beim Deployment Ihres Systems als statische Cachegröße verwendet werden.
- **m** Schreibt beim Abschluss einer Synchronisation die Dauer der Synchronisation und die Dauer der einzelnen Synchronisationsphasen ins Log. Die Synchronisationsphasen sind im Folgenden

dargestellt. Es handelt sich dabei um dieselben Phasen, die im MobiLink-Profil angezeigt werden. Alle Zeitangaben sind in Millisekunden (ms).

- **Synchronisationsanforderung** Die Zeitspanne ab der Herstellung der Netzwerkverbindung zwischen dem MobiLink-Client und dem MobiLink-Server bis zum Empfang der ersten Bytes des Upload-Datenstroms.
- **Upload empfangen** Die Zeitspanne ab dem Empfang der ersten Bytes des Uploaddatenstroms durch den MobiLink-Server bis zum vollständigen Abschluss der Übertragung des Uploaddatenstroms aus dem MobiLink-Client. Die Zeit ist möglicherweise auch bei einer nur als Download durchgeführten Synchronisation von Bedeutung. Die Zeit hängt von der Größe des Uploaddatenstroms und der Netzwerkbandbreite für die Übertragung ab.
- **DB-Worker abrufen** Die zum Abrufen eines freien Datenbank-Worker-Thread benötigte Zeit.
- **Verbinden** Die vom Datenbank-Worker-Thread benötigte Zeit, um bei Bedarf eine Datenbankverbindung herzustellen. Zum Beispiel nach einem Fehler bei der vorherigen Verbindung oder wenn die Skriptversion geändert wurde.
- **Benutzer authentifizieren** Die für die Authentifizierung des Benutzers benötigte Zeit.
- **Synchronisationsbeginn** Die für das Ereignis begin\_synchronization benötigte Zeit, sofern definiert, plus die Zeit für das Abrufen der letzten Uploadzeit last\_upload\_time für jede Subskription.
- **Upload übernehmen** Die für das Übernehmen der Uploaddaten in die konsolidierte Datenbank benötigte Zeit.
- **Download vorbereiten** Die für das prepare\_for\_download-Ereignis benötigte Zeit.
- **Download abrufen** Die Zeit, die benötigt wird, um die Zeilen abzurufen, die von der konsolidierten Datenbank heruntergeladen werden sollen, um den Downloaddatenstrom zu erstellen. Die Phase des Downloadabrufs umfasst nicht die Zeit, die benötigt wird, um den Downloaddatenstrom zu erstellen, was in der Phase des Downloadsendevorgangs geschieht. Dies kann bei großen Downloads recht lange dauern, wenn der Download nicht in den Speicher passt.
- **Synchronisationsende** Die Zeit, die für das Ereignis end\_synchronization benötigt wird, nach dem der Datenbank-Worker-Thread freigegeben wird. Diese Phase tritt ein, bevor der Downloaddatenstrom an die entfernte Datenbank gesendet wird.
- **Download senden** Die Zeit, die benötigt wird, um den Downloaddatenstrom an die entfernte Datenbank zu senden. Die Zeit hängt von der Größe des Download-Datenstroms und der Netzwerkbandbreite für die Übertragung ab. Für eine reine Upload-Synchronisation ist der Download-Datenstrom einfach eine Upload-Bestätigung.

Die Download\_senden-Phase umfasst die Zeit, die benötigt wird, um den Downloaddatenstrom zu erstellen. Dies kann bei großen Downloads recht lange dauern, wenn der Download nicht in den Speicher passt.

- **Auf Download-Quittierung warten** Die Zeit, die gewartet wird, bis der Download in die entfernte Datenbank übernommen und die Downloadbestätigung von der entfernten Datenbank

gesendet wird. Diese Phase wird nur angezeigt, wenn der MobiLink-Client die Downloadbestätigung aktiviert hat.

- **DB-Worker für Download-Quittierung abrufen** Die Zeit, die auf einen freien Datenbank-Worker-Thread gewartet wird, nachdem eine Downloadbestätigung empfangen wurde. Diese Phase wird nur angezeigt, wenn der MobiLink-Client die Downloadbestätigung aktiviert hat.
- **Für Download-Quittierung verbinden** Die Zeit, die der Datenbank-Worker-Thread benötigt, um bei Bedarf eine Datenbankverbindung herzustellen. Diese Phase wird nur angezeigt, wenn der MobiLink-Client die Downloadbestätigung aktiviert hat.
- **Nicht blockierende Download-Quittierung** Die Zeit, die für die Verbindungsereignisse `publication_nonblocking_download_ack` und `nonblocking_download_ack` benötigt wird. Diese Phase wird nur angezeigt, wenn der MobiLink-Client die Downloadbestätigung aktiviert hat.

Jeder Ausgabezeile wird PHASE: vorangestellt, um die Suche von Werten zu erleichtern.

Die folgende Beispielausgabe gibt die Dauer der verschiedenen Synchronisationsphasen an:

```
I. 2008-06-05 14:48:36. <1> PHASE: start_time: 2008-06-05 14:48:36.048
I. 2008-06-05 14:48:36. <1> PHASE: duration: 175
I. 2008-06-05 14:48:36. <1> PHASE: sync_request: 0
I. 2008-06-05 14:48:36. <1> PHASE: receive_upload: 19
I. 2008-06-05 14:48:36. <1> PHASE: get_db_worker: 0
I. 2008-06-05 14:48:36. <1> PHASE: connect: 18
I. 2008-06-05 14:48:36. <1> PHASE: authenticate_user: 51
I. 2008-06-05 14:48:36. <1> PHASE: begin_sync: 69
I. 2008-06-05 14:48:36. <1> PHASE: apply_upload: 0
I. 2008-06-05 14:48:36. <1> PHASE: prepare_for_download: 1
I. 2008-06-05 14:48:36. <1> PHASE: fetch_download: 4
I. 2008-06-05 14:48:36. <1> PHASE: wait_for_download_ack: 0
I. 2008-06-05 14:48:36. <1> PHASE: end_sync: 0
I. 2008-06-05 14:48:36. <1> PHASE: send_download: 10
I. 2008-06-05 14:48:36. <1> PHASE: get_db_worker_for_download_ack: 0
I. 2008-06-05 14:48:36. <1> PHASE: connect_for_download_ack: 0
I. 2008-06-05 14:48:36. <1> PHASE: nonblocking_download_ack: 0
```

- **n** Zeigt die Summen der Zeilenanzahl für jede Synchronisation an.
- **o** Zeigt SQL-Passthrough-Aktivität an.
- **p** Zeigt entfernte und konsolidierte Fortschritts-Offsets jeder Synchronisation an.
- **q** Zeigt die Spaltenwerte der einzelnen Download-Zeilen an. Verwenden Sie diese Option anstelle von `-vr`, womit die Spaltenwerte der einzelnen Upload- und Downloadzeilen angezeigt werden, um die Menge der protokollierten Daten zu reduzieren. Die Angabe von `-vi` zusammen mit `-vq` entspricht der Angabe von `-vr`.
- **r** Zeigt die Spaltenwerte der einzelnen Upload- bzw. Downloadzeilen an. Wenn nur die Spaltenwerte der einzelnen Uploadzeilen protokolliert werden sollen, verwenden Sie `-vi`. Wenn nur die Spaltenwerte der einzelnen Downloadzeilen protokolliert werden sollen, verwenden Sie `-vq`.
- **R** Zeigt nur bei Synchronisationen die entfernte ID in jeder Lognachricht an. Der MobiLink-Server fügt den Logeinträgen das Präfix `yyyy-mm-dd hh:mm:ss. <sync_id> (remote_id,)` hinzu.

Verwenden Sie diese Option mit der Option -vU, um in der Lognachricht auch den Benutzernamen anzuzeigen.

Diese beiden Befehlszeilenoptionen sind nicht von der Option -v+ betroffen. D.h. der MobiLink-Server fügt keine entfernte ID oder MobiLink-Benutzernamen in die Lognachrichten ein, selbst wenn die Option -v+ verwendet wird.

- **s** Zeigt den Namen jedes Synchronisationsskripts an, wenn es aufgerufen wird.
- **t** Zeigt den konvertierten SQL-Code an, der aus den Skripten stammt, die in strengem ODBC-Format geschrieben sind. Diese Stufe impliziert c. Folgendes Beispiel zeigt die automatische Konvertierung einer Anweisung für SQL Anywhere:

```
I. 2009-02-11 11:02:14. [102]: begin_upload synch2
{ call SynchLogLine( ?, ?, 'begin_upload' ) }
I. 2009-02-11 11:02:14. [102]: Translated SQL:
call SynchLogLine( ?, ?, 'begin_upload' )
```

Das nächste Beispiel zeigt die Konvertierung derselben Anweisung für den Microsoft SQL Server.

```
I. 2009-02-11 11:03:21. [102]: begin_upload synch2
{ call SynchLogLine( ?, ?, 'begin_upload' ) }
I. 2009-02-11 11:03:21. [102]: Translated SQL:
EXEC SynchLogLine ?, ?, 'begin_upload'
```

- **u** Zeigt nicht definierte Tabellenskripten an. Dies kann neuen Benutzern helfen, den Synchronisationsprozess und die Abfolge von Ereignissen zu verstehen.
- **U** Zeigt nur bei Synchronisationen den Benutzernamen in jeder Lognachricht an. Der MobiLink-Server fügt den Logeinträgen das Präfix yyyy-mm-dd hh:mm:ss. <sync\_id> (,user\_name) hinzu.

Verwenden Sie diese Option mit der Option -vR, um in der Lognachricht auch die entfernte ID anzuzeigen.

Diese beiden Befehlszeilenoptionen sind nicht von der Option -v+ betroffen. D.h. der MobiLink-Server fügt keine entfernte ID oder MobiLink-Benutzernamen in die Lognachrichten ein, selbst wenn die Option -v+ verwendet wird.

### Siehe auch

- [„Statistikeigenschaften für die MobiLink-Synchronisation“ auf Seite 209](#)

## mlsrv16-Option -w

Legt die anfängliche Anzahl von gleichzeitigen Datenbank-Worker-Threads fest, bis zu der Thread-Anzahl, die mit der Option -wm angegeben wurde.

### Syntax

```
mlsrv16 -c "connection-string" -w count ...
```

## Bemerkungen

Jeder Datenbank-Worker-Thread akzeptiert Synchronisationsanforderungen einzeln nacheinander, aber auch gleichzeitig mit allen anderen Datenbank-Worker-Threads.

Jeder Datenbank-Worker-Thread verwendet eine Verbindung zur konsolidierten Datenbank. Der MobiLink-Server öffnet eine zusätzliche Verbindung für Verwaltungszwecke. Daher beträgt die Mindestanzahl von Verbindungen des MobiLink-Servers zur konsolidierten Datenbank *count* + 1.

Die Anzahl der Datenbank-Worker-Threads hat einen starken Einfluss auf den MobiLink-Synchronisationsdurchsatz. Sie müssen Tests ausführen, um die maximale Anzahl für Ihre Synchronisationsumgebung zu bestimmen. Die Anzahl der Datenbank-Worker-Threads legt fest, wie viele Synchronisationen gleichzeitig in der konsolidierten Datenbank aktiv sein können. Der Rest wartet in einer Warteschlange, bis Datenbank-Worker-Threads verfügbar werden. Das Hinzufügen von Datenbank-Worker-Threads steigert *eventuell* den Durchsatz, erhöht aber auch die Möglichkeit eines Konflikts zwischen aktiven Synchronisationen. Ab einer bestimmten Anzahl verringert das Hinzufügen von Datenbank-Worker-Threads den Durchsatz, da die zunehmende Anzahl von Konflikten den Vorteil überlappender Synchronisationen aufwiegt.

Der Wert für diese Option ist auch die Standardeinstellung für die Option *-wu*, mit der die Anzahl der Threads für einen gleichzeitigen Upload in die konsolidierte Datenbank begrenzt werden kann. Dies ist nützlich, wenn die optimale Anzahl von Datenbank-Worker-Threads für Downloads größer ist als die optimale Anzahl für Uploads. Den besten Durchsatz erzielen Sie, wenn eine große Anzahl von Datenbank-Worker-Threads (über *-w*) mit einer kleinen zulässigen Anzahl von gleichzeitigen Uploads (über *-wu*) verwendet wird. Im Allgemeinen hängt der optimale Wert für *-wu* von der konsolidierten Datenbank ab und ist relativ unabhängig von der Verarbeitungs- oder Netzwerkgeschwindigkeit für die entfernten Datenbanken. Wenn Sie daher die Anzahl der Threads mit *-w* erhöhen, sollten Sie die Anzahl der Threads, die gleichzeitig Uploads durchführen können, mit *-wu* begrenzen.

Die Standardanzahl von Datenbank-Worker-Threads ist **5**.

## Siehe auch

- „*mlsrv16*-Option *-wm*“ auf Seite 91
- „*mlsrv16*-Option *-wu*“ auf Seite 92
- „*mlsrv16*-Option *-sm*“ auf Seite 81
- „*mlsrv16*-Option *-cn*“ auf Seite 56

## mlsrv16-Option -wm

Legt die maximale Anzahl von gleichzeitigen Datenbank-Worker-Threads fest.

## Syntax

**mlsrv16 -c "connection-string" -wm count ...**

## Bemerkungen

Der MobiLink-Server überwacht die Performance und passt nach Bedarf automatisch die Anzahl der Datenbank-Worker-Threads an. Der MobiLink-Server verwendet einen beliebigen Wert zwischen dem

anfänglichen Wert, der mit der Option -w festgelegt wurde, und dem Höchstwert, der mit der Option -wm festgelegt wurde.

Diese Funktion ermöglicht es, bei Deployments einen besseren Durchsatz mit weniger Auslastungstests zu erreichen. Wenn der Bereich zwischen -w und -wm groß genug ist, findet der MobiLink-Server automatisch die Anzahl der Datenbank-Worker-Threads, die den besten Durchsatz erzielt. Die Heuristik, die verwendet wird, um die Anzahl der Datenbank-Worker-Threads anzupassen, funktioniert jedoch möglicherweise nicht in allen Fällen gut. Außerdem ist es möglich, dass der beste Durchsatz außerhalb der Grenzen von -w und -wm liegt. Nur mit deploymentspezifischen Auslastungstests können Sie wirklich die Anzahl von Datenbank-Worker-Threads ermitteln, die den besten Durchsatz liefert.

Wenn dieser Wert nicht eingestellt wird, ist die maximale Anzahl der Datenbank-Worker-Threads standardmäßig der in der Option -w festgelegte Wert. Wenn die Option -wm nicht verwendet wird, ist die Anzahl der Datenbank-Worker-Threads fest und entspricht dem Wert der Option -w. Dann passt der MobiLink-Server sie nicht automatisch an.

### Siehe auch

- „mlsrv16-Option -w “ auf Seite 90
- „mlsrv16-Option -wu “ auf Seite 92
- „mlsrv16-Option -sm“ auf Seite 81
- „mlsrv16-Option -cn “ auf Seite 56
- Automatische Anpassung der Datenbank-Worker-Threads auf Seite 145

## mlsrv16-Option -wu

Legt die Höchstzahl der Datenbank-Worker-Threads fest, die gleichzeitig Uploads durchführen dürfen.

### Syntax

**mlsrv16 -c "connection-string" -wu count ...**

### Bemerkungen

Mit der Option -wu wird die Anzahl der Datenbank-Worker-Threads, die Uploads gleichzeitig in die konsolidierte Datenbank übernehmen können, begrenzt. Wenn die Obergrenze erreicht ist, muss ein Datenbank-Worker-Thread, der bereitsteht, um einen Upload auf die konsolidierte Datenbank auszuführen, warten, bis ein anderer seinen Upload beendet hat.

Der häufigste Grund für einen Konflikt in der konsolidierten Datenbank ist der, dass zu viele Datenbank-Worker-Threads gleichzeitig Uploads übernehmen. Downloads verursachen deutlich weniger Konflikte. Daher werden sie nur durch die mlsrv16-Option -w eingeschränkt. Aus diesem Grund muss die Einstellung -w größer oder gleich der Einstellung -wu sein.

Standardmäßig können von allen Datenbank-Worker-Threads gleichzeitig Uploads durchgeführt werden. Die Anzahl der benutzten Datenbank-Worker-Threads wird durch die Option -w festgelegt. Der Standardwert ist 5.

Wenn die Option -wu nicht angegeben wird, können Uploads gleichzeitig von einzelnen oder allen Datenbank-Worker-Threads übernommen werden. Wenn -wu angegeben ist, werden Uploads gleichzeitig

nur von der angegebenen Anzahl von Datenbank-Worker-Threads übernommen. Dies kann temporär zu stärkeren Konflikten führen, während der MobiLink-Server die Anzahl der Datenbank-Worker-Threads erhöht, um dadurch den Durchsatz zu verbessern. Wenn diese Bedingung erkannt wurde, wird die Thread-Anzahl verringert.

Die Einstellung -wu ist in Umgebungen mit hoher Auslastung empfehlenswert, in denen Uploads ständig mit reinen Download-Synchronisationen gemischt werden.

### Beispiel

In einer Pilot-Umgebung mit einem LAN und entfernten Datenbanken auf PCs können Sie feststellen, dass die optimale Zahl an Datenbank-Worker-Threads für reine Upload- und Download-Synchronisationen etwa 10 ist. Dies entspricht einer CPU-Nutzung von 100 % in der konsolidierten Datenbank. Bei weniger Datenbank-Worker-Threads können Sie feststellen, dass der Durchsatz und die CPU-Nutzung der konsolidierten Datenbank niedriger ist. Bei mehr Datenbank-Worker-Threads steigt der Durchsatz nicht an, da die konsolidierte Datenbank mit 10 Worker-Threads bereits die maximale Verarbeitungsgeschwindigkeit erreicht.

### Siehe auch

- „mlsrv16-Option -w“ auf Seite 90
- „mlsrv16-Option -wm“ auf Seite 91
- „mlsrv16-Option -sm“ auf Seite 81

## mlsrv16-Option -x

Legt Netzwerkprotokolloptionen fest, die vom MobiLink-Server verwendet werden, um Synchronisationsanforderungen zu empfangen.

### Hinweis

Erforderliche getrennt lizenzierbare Komponenten.

FIPS-zertifizierte Verschlüsselung erfordert eine separate Lizenz. Alle Technologien für starke Verschlüsselungen unterliegen Exportbestimmungen.

Siehe „Getrennt lizenzierbare Komponenten“ [*SQL Anywhere 16 - Einführung*].

### Syntax

```
mlsrv16 -c "connection-string" -x protocol[protocol-options] [ -x protocol[protocol-options] ... ] ...
```

*protocol* : tcpip | tls | http | https | oe

*protocol-options* : ( option=value; ... )

### Bemerkungen

Die Option -x muss für jedes verwendete Protokoll angegeben werden. Beispiel: Damit MobiLink sowohl TCP/IP als auch HTTP abhört, können Sie Folgendes eingeben:

```
mlsrv16 -x tcpip(port=1234) -x http(port=2222)
```

Standardwert ist tcpip mit Port 2439.

Parameter

Folgende Werte für Protokoll sind zulässig:

- **tcpip** Akzeptiert Verbindungen über TCP/IP.
- **tls** Akzeptiert Verbindungen mit TCP/IP und Transportschichtssicherheit.
- **http** Akzeptiert Verbindungen über das Standard-HTTP-Web-Protokoll.
- **https** Akzeptiert Verbindungen über eine Variante von HTTP, die sichere Transaktionen abwickelt. Das HTTPS-Protokoll implementiert HTTP über SSL/TLS unter Verwendung der RSA-Verschlüsselung.
- **oe** Verwendet einen integrierten Outbound Enabler bei der Verwendung des Relay-Servers. Die Befehlszeile des Servers kann nicht mehr als ein **-x oe**-Protokoll enthalten.

Sie können auch die folgenden Netzwerkprotokolloptionen in der Form von *Option=Wert* angeben. Mehrere Optionen müssen durch Semikola getrennt werden.

- **TCP/IP-Optionen** Wenn Sie das TCP/IP-Protokoll verwenden, können Sie optional die folgenden Protokolloptionen festlegen (diese Optionen beachten die Groß-/Kleinschreibung):

| TCP/IP-Protokolloption                                  | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>collect_network_data</b> = <code>{ yes   no }</code> | Aktiviert Synchronisationsskripten, um Netzwerkinformationen von jeder Synchronisation zu lesen. Siehe <a href="#">NetworkData-Schnittstelle [MobiLink-Server Java] auf Seite 501</a> und <a href="#">NetworkData-Schnittstelle [MobiLink-Server-API für .NET] auf Seite 597</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>host</b> = <i>hostname</i>                           | Der Hostname oder die IP-Nummer, die der MobiLink-Server überwachen soll. Der Standardwert ist localhost.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>ignore</b> = <i>hostname</i>                         | Ein Hostname oder eine IP-Nummer, der bzw. die vom MobiLink-Server bei der Herstellung einer Verbindung ignoriert wird. Mithilfe dieser Option können Sie Anforderungen von Lastverteilern auf der tiefstmöglichen Stufe ignorieren und so eine übermäßige Ausgabe im MobiLink-Serverlog und in den MobiLink-Profiler-Ausgabedateien verhindern. Sie können mehrere zu ignorierende Hosts angeben. Zum Beispiel: <code>-x tcpip(ignore=lbl;ignore=123.45.67.89)</code> . Wenn Sie mehrere Instanzen von <code>-x</code> in einer Befehlszeile angeben, wird der Host bei allen Instanzen ignoriert. Beispiel: Wenn Sie <code>-x tcpip(ignore=1.1.1.1) -x http</code> angeben, werden die Verbindungen für 1.1.1.1 sowohl für den TCP/IP- als auch für den HTTP-Datenstrom ignoriert. |

| TCP/IP-Protokolloption | Beschreibung                                                                                                                                                  |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>port=portnumber</b> | Die Socket-Portnummer, die der MobiLink-Server überwachen soll. Der Standardport ist 2439. Dies ist die IANA-registrierte Portnummer für den MobiLink-Server. |

- **Optionen für TCP/IP mit Transportschichtssicherheit** Wenn Sie das TLS-Protokoll festlegen (TLS ist TCP/IP mit Transportschichtssicherheit), können Sie wahlweise die folgenden Protokolloptionen angeben (diese Optionen beachten die Groß-/Kleinschreibung):

| TLS-Protokolloptionen                     | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>collect_network_data={ yes   no }</b>  | Aktiviert Synchronisationsskripten, um Netzwerkinformationen von jeder Synchronisation zu lesen. Siehe <a href="#">NetworkData-Schnittstelle [MobiLink-Server Java] auf Seite 501</a> und <a href="#">NetworkData-Schnittstelle [MobiLink-Server-API für .NET] auf Seite 597</a> .                                                                                                                                                                                                                                                                                                                                                                      |
| <b>e2ee_private_key=file</b>              | Die PEM- oder DER-kodierte Datei, die den privaten RSA-Schlüssel enthält. Diese Option ist erforderlich, damit die Punkt zu Punkt-Verschlüsselung wirksam werden kann.<br><br>PEM- und DER-kodierte Dateien werden mithilfe des Dienstprogramms createkey erstellt. Siehe „ <a href="#">Schlüsselpaargenerator-Dienstprogramm (createkey)</a> “ [ <a href="#">SQL Anywhere Server - Datenbankadministration</a> ].                                                                                                                                                                                                                                      |
| <b>e2ee_private_key_password=password</b> | Das Kennwort für die private Schlüsseldatei. Diese Option ist erforderlich, damit die Punkt zu Punkt-Verschlüsselung wirksam werden kann.<br><br>Wenn diese Option angegeben ist, muss der e2ee_private_key-Parameter ebenfalls angegeben werden.<br><br>Um zu verhindern, dass das Kennwort in der Befehlszeile des MobiLink-Servers angezeigt wird, verwenden Sie das Dienstprogramm dbfhide. Siehe „ <a href="#">Dienstprogramm zum Verschleiern von Dateien (dbfhide)</a> “ [ <a href="#">SQL Anywhere Server - Datenbankadministration</a> ].                                                                                                      |
| <b>fips={yes no}</b>                      | Wenn Sie das TLS-Protokoll mit RSA verwenden, können Sie "fips=yes" angeben, um Verbindungen zu akzeptieren, die das TCP/IP-Protokoll und FIPS-zertifizierte Verschlüsselungsalgorithmen einsetzen. FIPS-zertifizierte Verbindungen verwenden eine separate, FIPS 140-2-zertifizierte Software. Server, die RSA-Verschlüsselung ohne FIPS-zertifizierte Verschlüsselung verwenden, sind mit Clients kompatibel, die RSA verwenden und bei denen die fips-Option aktiviert ist. Server, die RSA verwenden und bei denen die fips-Option aktiviert ist, sind mit Clients kompatibel, die RSA verwenden und bei denen die fips-Option nicht aktiviert ist. |

| <b>TLS-<br/>Protokolloptionen</b>                     | <b>Beschreibung</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>host</b> = <i>hostname</i>                         | Der Hostname oder die IP-Nummer, die der MobiLink-Server überwachen soll. Der Standardwert ist localhost.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>identity</b> = <i>Identitätsdatei</i>              | Pfad und Dateiname der Identitätsdatei, die für die Serverauthentifizierung verwendet werden sollen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>identity_password</b> = <i>Kennwort</i>            | <p>Ein optionaler Parameter, der ein Kennwort für die Identitätsdatei festlegt.</p> <p>Wenn diese Option angegeben ist, muss die identity-Option ebenfalls angegeben werden.</p> <p>Siehe „<a href="#">Transportschichtssicherheit</a>“ [<a href="#">SQL Anywhere Server - Datenbankadministration</a>].</p> <p>Um zu verhindern, dass das Kennwort in der Befehlszeile des MobiLink-Servers angezeigt wird, verwenden Sie das Dienstprogramm dbfhide. Siehe „<a href="#">Dienstprogramm zum Verschleiern von Dateien (dbfhide)</a>“ [<a href="#">SQL Anywhere Server - Datenbankadministration</a>].</p> |
| <b>ignore</b> = <i>Hostname</i>                       | Ein Hostname oder eine IP-Nummer, der bzw. die vom MobiLink-Server bei der Herstellung einer Verbindung ignoriert wird. Mithilfe dieser Option können Sie Anforderungen von Lastverteilern auf der tiefstmöglichen Stufe ignorieren und so eine übermäßige Ausgabe im MobiLink-Serverlog und in den MobiLink-Profiler-Ausgabedateien verhindern. Sie können mehrere zu ignorierende Hosts angeben. Zum Beispiel: <code>-x tcpip(ignore=lbl;ignore=123.45.67.89)</code> .                                                                                                                                  |
| <b>port</b> = <i>Portnummer</i>                       | Die Socket-Portnummer, die der MobiLink-Server überwachen soll. Der Standardport ist 2439. Dies ist die IANA-registrierte Portnummer für den MobiLink-Server.                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>trusted_certificates</b> = <i>Zertifikatsdatei</i> | Verwenden Sie diese Option, um zu gewährleisten, dass das Clientzertifikat gültig ist, und führen Sie anschließend mithilfe der NetworkData.ClientCertificates-API die weitere Authentifizierung des Zertifikats im authenticate_user-Skript durch. Siehe <a href="#">NetworkData-Schnittstelle [MobiLink-Server Java]</a> auf Seite 501 und <a href="#">NetworkData-Schnittstelle [MobiLink-Server-API für .NET]</a> auf Seite 597.                                                                                                                                                                      |

- **HTTP-Optionen** Wenn Sie das HTTP-Protokoll verwenden, können Sie optional die folgenden Protokolloptionen festlegen (diese Optionen beachten die Groß-/Kleinschreibung):

| HTTP-Optionen                                     | Beschreibung                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>buffer_size</b> = <i>Zahl</i>                  | Die maximale Größe einer vom MobiLink-Server gesendeten HTTP-Nachricht in Byte. Eine Veränderung dieser Option führt zu einer Zu- oder Abnahme des für das Versenden von HTTP-Nachrichten zugewiesenen Speicherplatzes. Der Standardwert ist 65536 Byte.                                                                                                            |
| <b>collect_network_data</b> = <i>{ yes   no }</i> | Aktiviert Synchronisationsskripten, um Netzwerkinformationen von jeder Synchronisation zu lesen. Siehe <a href="#">NetworkData-Schnittstelle [MobiLink-Server Java] auf Seite 501</a> und <a href="#">NetworkData-Schnittstelle [MobiLink-Server-API für .NET] auf Seite 597</a> .                                                                                  |
| <b>header_limit</b> = <i>Anzahl</i>               | Die maximale Menge von Headerdaten, die in einer HTTP-Anforderung gesendet werden kann. Wenn eine Abfrage den angegebenen Wert überschreitet, gibt der Server einen HTTP-Fehlercode zurück und bricht die Anforderung ab. Mit <code>-x http(header_limit=200000)</code> wird die Grenze beispielsweise auf 200000 Byte erhöht. Der Standardwert beträgt 64000 Byte. |
| <b>host</b> = <i>Hostname</i>                     | Der Hostname oder die IP-Nummer, die der MobiLink-Server überwachen soll. Der Standardwert ist localhost.                                                                                                                                                                                                                                                           |
| <b>log_bad_request</b> = <i>{ yes   no }</i>      | Bei der Einstellung "yes" gibt der MobiLink-Server einen Fehler aus, wenn er eine unvollständige oder unerwartete HTTP-Anforderung empfängt. Diese Fehler sind analog denjenigen, die von der Option <code>-vf</code> ausgegeben werden. Der Standardwert ist "no". Siehe „ <a href="#">mlsrv16-Option -v</a> “ auf Seite 86.                                       |
| <b>port</b> = <i>Portnummer</i>                   | Die Socket-Portnummer, die der MobiLink-Server überwachen soll. Der Standardport ist 80.                                                                                                                                                                                                                                                                            |
| <b>version</b> = <i>HTTP-Version</i>              | Der MobiLink-Server erkennt automatisch die von einem Client verwendete HTTP-Version. Dieser Parameter ist eine Zeichenfolge, die angibt, welche HTTP-Version standardmäßig zu verwenden ist, wenn der Server die vom Client verwendete Version nicht erkennt. Sie können zwischen 1.0 und 1.1 wählen. Die Standardeinstellung ist 1.1.                             |

- **HTTPS-Optionen** Das HTTPS-Protokoll verwendet digitale RSA-Zertifikate für die Transportschichtsicherheit. Wenn Sie die FIPS-Verschlüsselung angeben, verwendet das Protokoll separate FIPS 140-2-zertifizierte Software, die mit HTTPS kompatibel ist.

Weitere Hinweise finden Sie unter „[Den MobiLink-Server mit Transportschichtsicherheit starten](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

Wenn Sie das https-Protokoll verwenden, können Sie optional die folgenden Protokolloptionen festlegen (bei diesen Optionen wird zwischen Groß-/Kleinschreibung unterschieden):

| HTTPS-Optionen                            | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>buffer_size=Zahl</b>                   | Die maximale Größe einer vom MobiLink-Server gesendeten HTTPS-Nachricht in Byte. Eine Veränderung dieser Option führt zu einer Zu- oder Abnahme des für das Versenden von HTTPS-Nachrichten belegten Speicherplatzes. Der Standardwert ist 65536 Byte.                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>collect_network_data={ yes   no }</b>  | Aktiviert Synchronisationsskripten, um Netzwerkinformationen von jeder Synchronisation zu lesen. Siehe <a href="#">NetworkData-Schnittstelle [MobiLink-Server Java] auf Seite 501</a> und <a href="#">NetworkData-Schnittstelle [MobiLink-Server-API für .NET] auf Seite 597</a> .                                                                                                                                                                                                                                                                                                                                                                      |
| <b>e2ee_private_key=file</b>              | Die PEM- oder DER-kodierte Datei, die den privaten RSA-Schlüssel enthält. Diese Option ist erforderlich, damit die Punkt zu Punkt-Verschlüsselung wirksam werden kann.<br><br>PEM- und DER-kodierte Dateien werden mithilfe des Dienstprogramms createkey erstellt. Siehe „ <a href="#">Schlüsselpaargenerator-Dienstprogramm (createkey)</a> “ [ <a href="#">SQL Anywhere Server - Datenbankadministration</a> ].                                                                                                                                                                                                                                      |
| <b>e2ee_private_key_password=password</b> | Das Kennwort für die private Schlüsseldatei. Diese Option ist erforderlich, damit die Punkt zu Punkt-Verschlüsselung wirksam werden kann.<br><br>Wenn diese Option angegeben ist, muss die e2ee_private_key-Option ebenfalls angegeben werden.<br><br>Um zu verhindern, dass das Kennwort in der Befehlszeile des MobiLink-Servers angezeigt wird, verwenden Sie das Dienstprogramm dbfhide. Siehe „ <a href="#">Dienstprogramm zum Verschleiern von Dateien (dbfhide)</a> “ [ <a href="#">SQL Anywhere Server - Datenbankadministration</a> ].                                                                                                         |
| <b>fips={yes no}</b>                      | Wenn Sie das TLS-Protokoll mit RSA verwenden, können Sie "fips=yes" angeben, um Verbindungen zu akzeptieren, die das TCP/IP-Protokoll und FIPS-zertifizierte Verschlüsselungsalgorithmen einsetzen. FIPS-zertifizierte Verbindungen verwenden eine separate, FIPS 140-2-zertifizierte Software. Server, die RSA-Verschlüsselung ohne FIPS-zertifizierte Verschlüsselung verwenden, sind mit Clients kompatibel, die RSA verwenden und bei denen die fips-Option aktiviert ist. Server, die RSA verwenden und bei denen die fips-Option aktiviert ist, sind mit Clients kompatibel, die RSA verwenden und bei denen die fips-Option nicht aktiviert ist. |
| <b>header_limit=num-ber</b>               | Die maximale Menge von Headerdaten, die in einer HTTPS-Anforderung gesendet werden kann. Wenn eine Abfrage den angegebenen Wert überschreitet, gibt der Server einen Fehlercode zurück und bricht die Anforderung ab. Mit <code>-x https(header_limit=200000)</code> wird die Grenze beispielsweise auf 200000 Byte erhöht. Der Standardwert beträgt 64000 Byte.                                                                                                                                                                                                                                                                                        |

| HTTPS-Optionen                                       | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>host=</b> <i>hostname</i>                         | Der Hostname oder die IP-Nummer, die der MobiLink-Server überwachen soll. Der Standardwert ist localhost.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>identity=</b> <i>server-identity</i>              | Pfad und Dateiname der Identitätsdatei, die für die Serverauthentifizierung verwendet werden sollen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>identity_password=</b> <i>password</i>            | <p>Ein optionaler Parameter, der ein Kennwort für die Identitätsdatei festlegt.</p> <p>Wenn diese Option angegeben ist, muss die identity-Option ebenfalls angegeben werden.</p> <p>Siehe „<a href="#">Transportschichtsicherheit</a>“ [<a href="#">SQL Anywhere Server - Datenbankadministration</a>].</p> <p>Um zu verhindern, dass das Kennwort in der Befehlszeile des MobiLink-Servers angezeigt wird, verwenden Sie das Dienstprogramm dbfhide. Siehe „<a href="#">Dienstprogramm zum Verschleiern von Dateien (dbfhide)</a>“ [<a href="#">SQL Anywhere Server - Datenbankadministration</a>].</p> |
| <b>log_bad_request=</b> { yes   no }                 | Bei der Einstellung "yes" gibt der MobiLink-Server einen Fehler aus, wenn er eine unvollständige oder unerwartete HTTP-Anforderung empfängt. Diese Fehler sind analog denjenigen, die von der Option -vf ausgegeben werden. Der Standardwert ist "no". Siehe „ <a href="#">mlsrv16-Option -v</a> “ auf Seite 86.                                                                                                                                                                                                                                                                                         |
| <b>port=</b> <i>portnumber</i>                       | Die Socket-Portnummer, die der MobiLink-Server überwachen soll. Die Portnummer muss mit dem Port übereinstimmen, zu dessen Überwachung der MobiLink-Server eingerichtet wurde. Der Standardport ist 443.                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>trusted_certificates=</b> <i>certificate_file</i> | Verwenden Sie diese Option, um zu gewährleisten, dass das Clientzertifikat gültig ist, und führen Sie anschließend mithilfe der NetworkData.ClientCertificates-API die weitere Authentifizierung des Zertifikats im authenticate_user-Skript durch. Siehe <a href="#">NetworkData-Schnittstelle [MobiLink-Server Java]</a> auf Seite 501 und <a href="#">NetworkData-Schnittstelle [MobiLink-Server-API für .NET]</a> auf Seite 597.                                                                                                                                                                     |
| <b>version=</b> <i>http-version</i>                  | Der MobiLink-Server erkennt automatisch die von einem Client verwendete HTTP-Version. Dieser Parameter ist eine Zeichenfolge, die angibt, welche HTTP-Version standardmäßig zu verwenden ist, wenn der Server die vom Client verwendete Version nicht erkennt. Sie können zwischen 1.0 und 1.1 wählen. Die Standardeinstellung ist 1.1.                                                                                                                                                                                                                                                                  |

- **Outbound Enabler-Optionen** Wenn Sie das OE-Protokoll verwenden, können Sie optional die folgenden Protokolloptionen festlegen (bei diesen Optionen ist die Groß-/Kleinschreibung wichtig):

| Outbound Enabler-Protokolloption | Beschreibung                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>buffer_size=number</b>        | Die maximale Größe einer vom MobiLink-Server gesendeten HTTP-Nachricht in Byte. Eine Veränderung dieser Option führt zu einer Zu- oder Abnahme des für das Versenden von HTTP-Nachrichten zugewiesenen Speicherplatzes. Der Standardwert ist 65536 Byte.                                                                                                   |
| <b>config=file</b>               | Die Outbound Enabler-Konfigurationsdatei. Sie hat das gleiche Format wie eine Konfigurationsdatei für den eigenständigen Outbound Enabler, außer dass Sie in dieser Datei nicht den Parameter -cs angeben dürfen.<br><br>Sie müssen genau eine Konfigurationsdatei angeben.                                                                                |
| <b>header_limit=number</b>       | Die maximale Menge von Headerdaten, die in einer OE-Anforderung gesendet werden kann. Wenn eine Abfrage den angegebenen Wert überschreitet, gibt der Server einen Fehlercode zurück und bricht die Anforderung ab. Mit <code>-x oe(header_limit=200000)</code> wird die Grenze beispielsweise auf 200000 Byte erhöht. Der Standardwert beträgt 64000 Byte. |
| <b>log_bad_request={yes no}</b>  | Bei der Einstellung "yes" gibt der MobiLink-Server einen Fehler aus, wenn er eine unvollständige oder unerwartete HTTP-Anforderung empfängt. Diese Fehler sind analog denjenigen, die von der Option -vf ausgegeben werden. Der Standardwert ist "no". Siehe „ <a href="#">mlsrv16-Option -v</a> “ auf Seite 86.                                           |
| <b>version=http-version</b>      | Der MobiLink-Server erkennt automatisch die von einem Client verwendete HTTP-Version. Dieser Parameter ist eine Zeichenfolge, die die Version des zu verwendenden HTTP angibt, wenn der Server die vom Client verwendete Methode nicht erkennt. Sie können zwischen 1.0 und 1.1 wählen. Die Standardeinstellung ist 1.1.                                   |

## Beispiel

Die folgende Befehlszeile setzt den TCP/IP-Port auf 12345:

```
mlsrv16 -c "DSN=SQL Anywhere 16 CustDB;UID=DBA;PWD=sql" -x tcpip(port=12345)
```

Im folgenden Beispiel werden der Typ der Sicherheit (RSA), die Serveridentitätsdatei und das Identitätskennwort, mit dem der private Schlüssel des Servers geschützt wird, angegeben:

```
mlsrv16 -c "DSN=my_cons"
-x tls(identity=c:\test\serv_rsa1.crt;identity_password=pwd)
```

Das folgende Beispiel ist ähnlich dem vorherigen, mit dem Unterschied, dass der Name der Identitätsdatei ein Leerzeichen enthält:

```
mlsrv16 -c "DSN=my_cons"
-x "tls(identity=c:\Program Files\test\serv_rsa1.crt;identity_password=pwd)"
```

Das folgende Beispiel zeigt die Verwendung der Punkt zu Punkt-Verschlüsselung über HTTPS:

```
mlsrvl6 -c "DSN=my_cons" -x https(identity=my_identity.crt;
identity_password=my_id_pwd;e2ee_private_key=my_pk.pem;
e2ee_private_key_password=my_pk_pwd)
```

Das folgende Beispiel zeigt die Verwendung des integrierten Outbound Enabler:

```
mlsrvl6 -c "DSN=my_cons" -x oe(config=oeconfig.txt)
```

Im obigen Beispiel enthält die Datei *oeconfig.txt* Folgendes:

```
-f farm.name
-id servername
-t a18b2e37dbc296322ec5a6eded6ba896
-cr "host=relayserver.sybase.com;https=0;port=80"
```

### trusted\_certificates-Beispiel für Java

Das folgende Beispiel zeigt, wie Sie die NetworkData-Schnittstelle verwenden können, um Zertifikatinformationen aus einer sicheren Synchronisation abzurufen.

```
public class OrderProcessor {
    DBConnectionContext _cc;

    public OrderProcessor( DBConnectionContext cc ) {
        _cc = cc;
    }

    // The method used for the authenticate_user event.
    public void AuthUser() {
        NetworkData nd = _cc.getNetworkData();
        if( nd != null ) {
            if( nd.isTLS() ) {
                CertPath certs = nd.getCertificateChain();
                if( certs != null ) {
                    System.out.println( "  client-side cert:" );
                    int n = 1;
                    for( Certificate c : certs.getCertificates() ) {
                        System.out.println( "    cert " + n++ );
                        X509Certificate c509 = (X509Certificate) c;
                        System.out.println( "      Subject: " +
c509.getSubjectX500Principal().getName() );
                        System.out.println( "      Issuer: " +
c509.getIssuerX500Principal().getName() );
                    }
                } else {
                    System.out.println( "  no client cert" );
                }
            }
        }
    }
}
```

Führen Sie die folgende SQL-Anweisung aus, um die Java-Methode zu registrieren.

```
ml_add_java_connection_script( <version>, 'authenticate_user',
'OrderProcessor.AuthUser' )
```

Die beiden folgenden Beispiele zeigen die Optionen, die Sie zur MobiLink-Befehlszeile hinzufügen müssen. Das erste Beispiel gilt für HTTPS, das zweite Beispiel für TLS.

```
mlsrv16 -c <connection_string> -x
https(collect_network_data=1;trusted_certificates=<certificate_file>) -sl
java

mlsrv16 -c <connection_string> -x
tls(collect_network_data=1;trusted_certificates=<certificate_file>) -sl java
```

### trusted\_certificates-Beispiel für .NET

Das folgende Beispiel zeigt, wie Sie die NetworkData-Schnittstelle verwenden können, um Zertifikatinformationen aus einer sicheren Synchronisation abzurufen.

```
public class OrderProcessor {
    DBConnectionContext _cc;

    public OrderProcessor( DBConnectionContext cc ) {
        _cc = cc;
    }

    public void AuthUser() {
        NetworkData nd = _cc.NetworkData;
        if( nd != null ) {
            if( nd.IsTLS ) {
                X509Certificate2Collection certs = nd.ClientCertificates;
                if( certs != null ) {
                    PrintLn( "    client-side cert:" );
                    int n = 1;
                    foreach( X509Certificate2 x509 in certs ) {
                        PrintLn( "        cert " + n++ );
                        PrintLn( "            Subject: " + x509.SubjectName.Name );
                        PrintLn( "            Issuer: " + x509.IssuerName.Name );
                    }
                }
            }
        }
    }
}
```

Führen Sie die folgende SQL-Anweisung aus, um die .NET-Methode zu registrieren.

```
ml_add_dnet_connection_script( <version>, 'authenticate_user',
    'OrderProcessor.AuthUser' )
```

Die beiden folgenden Beispiele zeigen die Optionen, die Sie zur MobiLink-Befehlszeile hinzufügen müssen. Das erste Beispiel gilt für HTTPS, das zweite Beispiel für TLS.

```
mlsrv16 -c <connection_string> -x
https(collect_network_data=1;trusted_certificates=<certificate_file>) -sl
dnet

mlsrv16 -c <connection_string> -x
tls(collect_network_data=1;trusted_certificates=<certificate_file>) -sl dnet
```

## mlsrv16-Option -zf

Gibt an, dass der MobiLink-Server zu Beginn jeder Synchronisation eine Prüfung auf Skriptänderungen durchführen soll.

**Vorsicht**

Das Starten des MobiLink-Server mit der Option `-zf` hat negative Auswirkungen auf die Performance des MobiLink-Servers und sollte vermieden werden, soweit möglich.

**Syntax**

**mlsrv16 -c "connection-string" -zf**

**Bemerkungen**

Der MobiLink-Server geht davon aus, dass keine Skriptänderungen vorgenommen wurden. Nur wenn die Option `-zf` verwendet wird, überprüft er nach dem Start, ob Skriptänderungen erfolgt sind.

## mlsrv16-Option -zp

Passt die Gesamtstellenzahl von Zeitstempelvergleichen zum Zweck der Konflikterkennung an.

**Syntax**

**mlsrv16 -c "connection-string" -zp**

**Bemerkungen**

Diese Option bewirkt, dass MobiLink-Server in entfernten und konsolidierten Datenbanken die höchste darstellbare Zeitstempelauflösung beim Vergleich von Zeitstempeln für die Konflikterkennung verwendet. Diese Option ist hilfreich, wenn die Zeitstempel in der konsolidierten Datenbank präziser sind als die der entfernten, da aktualisierte Zeitstempel in der entfernten Datenbank bei der nächsten Synchronisation fälschlich Konflikte verursachen können. Diese Option gestattet es MobiLink, diese Konflikte zu ignorieren. Wenn die Gesamtstellenzahl nicht übereinstimmt und die Option `-zp` nicht zum Einsatz kommt, wird je Synchronisation und je Tabelle eine schemasensitive Warnung in das Log geschrieben, um auf die Option `-zp` hinzuweisen. Es wird eine weitere Warnung je Synchronisation hinzugefügt, die den Benutzer anweist, die Gesamtstellenzahl des Zeitstempels in der entfernten Datenbank möglichst anzupassen.

## mlsrv16-Option -zs

Gibt einen MobiLink-Servernamen für `mlstop` an.

**Syntax**

**mlsrv16 -c "connection-string" -zs name**

**Bemerkungen**

Der Standardname ist `<default>`.

Der angegebene Name kann ASCII-Buchstaben und Ziffern, aber keine anderen Zeichen enthalten.

Wenn `mlstop` verwendet wird, um einen mit der Option `-zs` gestarteten MobiLink-Server herunterzufahren, muss in der `mlstop`-Befehlszeile der Servername angegeben werden. Zum Beispiel,

`mlstop myMLserver`. Das Herunterfahren kann nur dann auf dem Computer eingeleitet werden, auf dem der MobiLink-Server installiert ist.

#### Siehe auch

- „Dienstprogramm zum Stoppen von MobiLink (`mlstop`)“ auf Seite 698

## mlsrv16-Option -zt

Gibt die maximale Anzahl von Prozessoren an, die für den MobiLink-Server verwendet werden.

#### Syntax

`mlsrv16 -c "connection-string" -zt number`

#### Bemerkungen

Diese Option kann für einige ODBC-Treiber erforderlich sein. Damit haben Sie außerdem ein Hilfsmittel zur Feinabstimmung der Prozessor-Ressourcen.

Diese Option kann nur auf Windows- und Linux-Betriebssystemen verwendet werden. Standardwert ist die Anzahl der Prozessoren auf dem Computer.

## mlsrv16-Option -zu

Steuert das automatische Hinzufügen von Benutzern, wenn die `authenticate_user`- und `authenticate_user_hashed`-Skripten nicht definiert sind.

#### Syntax

`mlsrv16 -c "connection-string" -zu{ + | - } ...`

#### Bemerkungen

Wenn dieser Parameter als `-zu+` angegeben wird, werden bei der ersten Synchronisation nicht erkannte MobiLink-Benutzernamen der Tabelle `ml_user` automatisch hinzugefügt. Wenn das Argument in der Form `-zu-` oder gar nicht angegeben wird, werden nicht erkannte Benutzernamen von der Synchronisation ausgenommen.

Diese Option ist während der Entwicklungsphase nützlich, um Benutzer zu registrieren. Sie eignet sich nicht für bereitgestellte Anwendungen.

#### Siehe auch

- „Synchronisationen durch neue Benutzer“ [*MobiLink - Clientadministration*]
- „MobiLink-Benutzer“ [*MobiLink - Clientadministration*]
- „Dienstprogramm für die MobiLink-Benutzerauthentifizierung (`mluser`)“ auf Seite 699
- „`authenticate_user` (Verbindungsereignis)“ auf Seite 291

## mlsrv16-Option -zus

Bewirkt, dass der MobiLink-Server Uploadskripten für eine Tabelle aufruft, selbst wenn für die Tabelle keine Uploadzeilen vorhanden sind.

### Syntax

**mlsrv16 -c "connection-string" -zus ...**

### Bemerkungen

Standardmäßig ruft der MobiLink-Server keine Uploadskripten für eine Tabelle auf, wenn sie keine Uploadzeilen enthält, selbst wenn die Skripten definiert sind. Diese Option überschreibt das Standardverhalten und bewirkt, dass der MobiLink-Server Uploadskripten für eine Tabelle aufruft, selbst wenn für die Tabelle keine Uploadzeilen vorhanden sind.

## mlsrv16-Option -zw

Steuert, welche Stufe der Warnmeldungen angezeigt werden soll..

### Syntax

**mlsrv16 -c "connection-string" -zw levels**

### Bemerkungen

MobiLink kennt fünf Stufen von Warnmeldungen:

| Ebene | Beschreibung                                                                                                                |
|-------|-----------------------------------------------------------------------------------------------------------------------------|
| 0     | Alle Warnmeldungen unterdrücken                                                                                             |
| 1     | Server- und hohe ODBC-Stufe: Warnmeldungen beim Start des MobiLink-Servers                                                  |
| 2     | Synchronisations- und Benutzerstufe: Warnmeldungen, wenn eine Synchronisation startet                                       |
| 3     | Schema-Stufe: Warnmeldungen, wenn ein MobiLink-Server ein Client-Schema verarbeitet                                         |
| 4     | Skript- und niedrige ODBC-Stufe: Warnmeldungen, wenn ein MobiLink-Server Skripten abrufen, vorbereitet oder ausführt        |
| 5     | Tabellen- oder Zeilenstufe: Warnmeldungen, wenn ein MobiLink-Server Tabellenvorgänge in einem Upload oder Download ausführt |

Um festzulegen, welche Stufe der Warnmeldungen ausgegeben werden soll, können Sie die Stufen mit einem Komma oder einen Bereich mit zwei Punkten trennen. Beispielsweise ist: **-zw1..3,5** dasselbe wie **-zw1,2,3,5**.

Die Ausgabe von Meldungen hat Einfluss auf die Performance. Stufen mit höheren Zahlen produzieren gewöhnlich mehr Meldungen.

Wenn -zw in einer Befehlszeile mehrfach verwendet wird, erkennt MobiLink nur die letzte Angabe. Wenn die Einstellungen von -zw, -zwd und -zwe in Konflikt stehen, setzt MobiLink den Vorrang wie folgt: -zwe, dann -zwd, dann -zw.

Standardwert ist **1,2,3,4,5**, also werden alle Stufen der Warnmeldungen angezeigt.

## mlsrv16-Option -zwd

Deaktiviert spezifische Warncodes.

### Syntax

**mlsrv16 -c "connection-string" -zwd code, ...**

### Bemerkungen

Sie können spezifische Warncodes deaktivieren, sodass sie nicht gemeldet werden, selbst wenn andere Codes derselben Stufe angezeigt werden.

Eine vollständige Liste der Warnmeldungen finden Sie unter „[Warnmeldungen des MobiLink-Servers](#)“ [[Fehlermeldungen](#)].

Wenn -zwd in einer Befehlszeile mehrfach verwendet wird, sammelt MobiLink die Einstellungen. Wenn die Einstellungen von -zw, -zwd und -zwe in Konflikt stehen, setzt MobiLink den Vorrang wie folgt: -zwe, dann -zwd, dann -zw.

## mlsrv16-Option -zwe

Aktiviert spezifische Warncodes.

### Syntax

**mlsrv16 -c "connection-string" -zwe code, ...**

### Bemerkungen

Sie können spezifische Warncodes aktivieren, sodass sie angezeigt werden, obwohl Sie andere Codes derselben Stufe mit -zw deaktiviert haben.

Eine vollständige Liste der Warnmeldungen finden Sie unter „[Warnmeldungen des MobiLink-Servers](#)“ [[Fehlermeldungen](#)].

Wenn -zwe in einer Befehlszeile mehrfach verwendet wird, sammelt MobiLink die Einstellungen. Wenn die Einstellungen von -zw, -zwd und -zwe in Konflikt stehen, setzt MobiLink den Vorrang wie folgt: -zwe, dann -zwd, dann -zw.

# Synchronisationsmethoden

## MobiLink-Entwicklungstipps

Durch das Hinzufügen von Synchronisationsfunktionen zu einer Anwendung erhält diese Anwendung zusätzliche Komplexität. Obwohl die zusätzliche Komplexität fast immer überschaubar ist, müssen Sie sie berücksichtigen. Das komplette Synchronisationssystem, von den entfernten Datenbanken bis hin zur konsolidierten Datenbank, einschließlich anderer Anwendungen für die konsolidierte Datenbank, besteht aus vielen Teilen und jedes einzelne erfordert Ihre Aufmerksamkeit. Die folgenden Tipps können daher von Nutzen sein.

Wenn Sie Synchronisationen zu einem Anwendungsprototyp hinzufügen, ist es unter Umständen schwierig festzustellen, welche Komponenten Probleme bereiten. Sie sollten daher mit einem Prototyp ohne Synchronisation beginnen. Erst wenn Ihr Prototyp richtig funktioniert, sollten Sie die Synchronisation aktivieren.

Verwenden Sie zunächst nur einfache Synchronisationsverfahren. Bei Vorgängen wie dem einfachen Upload bzw. Download sind nur ein oder zwei Skripten erforderlich. Erst wenn diese Skripten fehlerfrei ausgeführt werden, können Sie die Synchronisation mit anspruchsvolleren Funktionen fortsetzen, z.B. mit Zeitstempeln, Primärschlüsselpools, Konfliktlösung und beliebiger Geschäftslogik.

### MobiLink und Primärschlüssel

In einem Synchronisationssystem können identische Zeilen in unterschiedlichen Datenbanken (entfernt und konsolidiert) nur über den Primärschlüssel ermittelt werden. Zudem bietet er die einzige Möglichkeit zur Konflikterkennung. Aus diesem Grund müssen MobiLink-Anwendungen folgende Regeln beachten:

- Jede zu synchronisierende Tabelle muss einen Primärschlüssel haben.
- Aktualisieren Sie nie die Werte von Primärschlüsseln in synchronisierten Tabellen.
- Primärschlüssel in synchronisierten Tabellen müssen für alle synchronisierten Datenbanken eindeutig sein.

### Siehe auch

- „Eindeutige Primärschlüssel“ auf Seite 117

## Implementieren zeitstempelbasierter Downloads

Die timestamp-Methode ist die nützlichste allgemeine Methode für effiziente Downloads. Dabei werden der Zeitpunkt der letzten Synchronisation der einzelnen Benutzer sowie nur die Downloadzeilen ermittelt, die seitdem geändert wurden.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

**Kontext und Bemerkungen**

MobiLink verwaltet einen TIMESTAMP-Wert, aus dem hervorgeht, wann die einzelnen MobiLink-Benutzer zuletzt Daten per Download übertragen haben. Dieser Wert wird als **Zeitpunkt des letzten Downloads** bezeichnet.

**Aufgabe**

1. Wenn Sie die zeitstempelbasierte Synchronisation für eine Tabelle implementieren möchten, fügen Sie der konsolidierten Datenbank eine last\_modified-Spalte hinzu, in die der Zeitpunkt der letzten Änderung der betreffenden Zeile gespeichert wird. Die Spalte wird normalerweise folgendermaßen deklariert:

| DBMS                       | Spalte mit Zeitpunkt der letzten Änderung                                           |
|----------------------------|-------------------------------------------------------------------------------------|
| Adaptive Server Enterprise | datetime                                                                            |
| IBM DB2 LUW                | timestamp NOT NULL GENERATED ALWAYS FOR EACH ROW ON UPDATE AS ROW CHANGE TIME-STAMP |
| Microsoft SQL Server       | datetime                                                                            |
| MySQL                      | timestamp default CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP                     |
| Oracle                     | timestamp                                                                           |
| SQL Anywhere               | timestamp DEFAULT timestamp                                                         |

2. Vergleichen Sie in Skripten für die Ereignisse download\_cursor und download\_delete\_cursor den ersten Parameter für den Wert in der Spalte TIMESPTAMP.

**Ergebnisse**

Die zeitstempelbasierte Synchronisation wird implementiert.

**Beispiel**

Das folgende, aus dem MobiLink-Beispiel Contact entnommene Beispiel veranschaulicht, wie Sie einen zeitstempelbasierten Download implementieren können.

- Tabellendefinition:

```
CREATE TABLE "DBA"."Customer"(  
  "cust_id" integer NOT NULL DEFAULT GLOBAL AUTOINCREMENT,  
  "name" char(40) NOT NULL,  
  "rep_id" integer NOT NULL,  
  "last_modified" timestamp NULL DEFAULT timestamp,  
  "active" bit NOT NULL,  
  PRIMARY KEY ("cust_id") )
```

- download\_cursor-Skript:

```
SELECT cust_id, Customer.name, Customer.rep_id
FROM Customer KEY JOIN SalesRep
WHERE Customer.last_modified >= {ml s.last_table_download}
AND SalesRep.ml_username = {ml s.username}
AND Customer.active = 1
```

#### Siehe auch

- „Synchronisieren der Ansprechpartner im Beispiel Contact“ [*MobiLink - Erste Orientierung*]
- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109
- „Quellcode der Synchronisationslogik“ [*MobiLink - Erste Orientierung*]

## Zeitpunkt des letzten Downloads in Skripten

Der Zeitstempel des letzten Downloads wird bei vielen MobiLink-Ereignissen als Parameter weitergegeben. Der Zeitstempel für den letzten Download ist der Wert, der während der letzten erfolgreichen Synchronisation unmittelbar vor der Download-Phase aus der konsolidierten Datenbank abgerufen wurde. Wenn der aktuelle MobiLink-Benutzer nicht synchronisiert hat bzw. nie erfolgreich synchronisiert hat, wird dieser Wert auf 1900-01-01 gesetzt.

Wenn Sie über mehrere Publikationen verfügen und diese zu unterschiedlichen Zeiten synchronisiert haben, können mehrere verschiedene Zeitstempel für den letzten Download vorliegen. Aus diesem Grund gibt es für die Zeitstempel des letzten Downloads zwei Skriptparameternamen:

- **last\_table\_download** ist der Zeitstempel für den letzten Download der aktuellen zu synchronisierenden Tabelle.
- **last\_download** Zeitpunkt der letzten Synchronisation aller Tabellen. Dies ist der früheste last\_table\_download-Wert aller zu synchronisierenden Tabellen.

Wenn Sie in MobiLink-Skripten Fragezeichen anstelle von benannten Parametern verwenden, wird auf der Basis des Ereignisses stets der korrekte Wert verwendet. Die Verwendung von Fragezeichen in SQL-Skripten ist veraltet und es wird empfohlen, benannte Parameter zu verwenden.

#### Vorsicht

Die Spalte mit den zuletzt geänderten Informationen sollte nicht synchronisiert werden. Falls Ihre entfernten Datenbanken eine derartige Spalte erfordern, sollten Sie einen anderen Spaltennamen verwenden. Andernfalls kann es sein, dass der TIMESTAMP-Wert durch den hochgeladenen Wert überschrieben wird und nicht den Zeitpunkt enthält, zu dem die Zeile in der konsolidierten Datenbank zuletzt aktualisiert wurde.

#### Siehe auch

- „So werden Download-Zeitstempel generiert und verwendet“ auf Seite 110
- „Skriptparameter“ auf Seite 236

#### Beispiel

```
SELECT cust_id, Customer.name, Customer.rep_id
FROM Customer KEY JOIN SalesRep
```

```
WHERE Customer.last_modified >= {ml s.last_table_download}
AND SalesRep.ml_username = {ml s.username}
AND Customer.active = 1
```

## So werden Download-Zeitstempel generiert und verwendet

MobiLink generiert und verwendet wie nachfolgend beschrieben einen Zeitstempel für zeitstempelbasierte Downloads:

- Nach der Festschreibung eines Uploads und kurz vor Aufruf des Ereignisses `prepare_for_download` ruft der MobiLink-Server die aktuelle Uhrzeit aus der konsolidierten Datenbank ab und speichert den Wert. Dieser **TIMESTAMP**-Wert stellt die Anfangszeit des aktuellen Downloads dar. Bei der nächsten Synchronisation sollten nur Daten heruntergeladen werden, die nach diesem Zeitpunkt geändert wurden.

### Hinweis

Wenn die konsolidierte Datenbank die Snapshot-Isolation unterstützt, ist der Download-Zeitstempel das Minimum der folgenden Werte:

- aktuelle Uhrzeit
- Start der ältesten offenen Transaktion

- Der MobiLink-Server sendet diesen **TIMESTAMP**-Wert als Teil des Downloads und der Client speichert ihn.
- Bei seiner nächsten Synchronisation verwendet der Client den **TIMESTAMP**-Wert für den **last\_download\_timestamp**, den er mit dem Upload sendet.
- Der MobiLink-Server übergibt den **last\_download\_timestamp**-Wert, den der Client soeben hochgeladen hat, an Ihre Download-Skripten. Ihr Cursor kann dann Änderungen mit Zeitstempeln auswählen, die aktueller als der letzte **last\_download\_timestamp**-Wert oder mit diesem Wert identisch sind, um sicherzustellen, dass nur neue Änderungen heruntergeladen werden.

## Speicherort der letzten Downloadzeit

Die letzte Downloadzeit wird in der entfernten Datenbank gespeichert. Dies ist der geeignete Speicherort, da nur die entfernte Datenbank weiß, ob der Download erfolgreich angewendet wurde.

Bei entfernten SQL Anywhere-Datenbanken wird die letzte Downloadzeit per Subskription gespeichert. Siehe „SYSSYNC-Systemansicht“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)].

Bei entfernten UltraLite-Datenbanken wird die letzte Downloadzeit per Publikation gespeichert. Siehe „syspublication-Systemtabelle“ [[UltraLite - Datenbankverwaltung](#)].

## Letzte Downloadzeit ändern

In einigen seltenen Fällen muss **last\_download\_timestamp** geändert werden. Wenn Sie beispielsweise versehentlich alle Daten in einer entfernten Datenbank gelöscht haben, können Sie sie erneut herunterladen, indem Sie ein Verbindungsskript namens `modify_last_download_timestamp` definieren, das den Wert für den Zeitstempel des letzten Downloads zurücksetzt. Andere Ereignisse

namens `generate_next_last_download_timestamp` und `modify_next_last_download_timestamp` setzen den Zeitstempel nicht für die aktuelle Synchronisation zurück, sondern für die nächste. Zum Beispiel können Sie einen UTC-TIMESTAMP-Wert mit UTC-Werten in den `last_modified`-Spalten Ihrer Tabellen vergleichen. Siehe:

- „`modify_last_download_timestamp` (Verbindungsereignis)“
- „`generate_next_last_download_timestamp` (Ereignis)“
- „`modify_next_last_download_timestamp` (Verbindungsereignis)“

UltraLite stellt auch eine Funktion zur Änderung der letzten Downloadzeit von der entfernten Datenbank bereit. Siehe:

- `ULResetLastDownloadTime`-Methode [UltraLite Embedded SQL] [*UltraLite - C- und C++-Programmierung*]
- `ULConnection.ResetLastDownloadTime`-Methode [UltraLite C++] [*UltraLite - C- und C++-Programmierung*]
- `ULConnection.ResetLastDownloadTime`-Methode [UltraLite.NET] [*UltraLite - .NET-Programmierung*]

#### Siehe auch

- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109

## Sommerzeit-Lösungen

Bei einem verteilten Datenbanksystem kann die Zeitumstellung zu Problemen führen, wenn Daten genau während der Zeitumstellung synchronisiert werden. Es kann sogar zu Datenverlusten kommen. Dieses Problem tritt nur im Herbst auf, wenn die Zeit zurückgestellt wird und es eine Stunde gibt, die nicht eindeutig definiert ist.

Um Probleme mit der Sommerzeit zu vermeiden, gibt es vier mögliche Lösungen:

- Sorgen Sie dafür, dass der konsolidierte Datenbankserver auf UTC-Zeit eingestellt ist.
- Schalten Sie die Sommerzeitumstellung auf dem konsolidierten Datenbankserver aus.
- Schalten Sie den Server eine Stunde aus, wenn die Zeit umgestellt wird.
- Verwenden Sie UTC-Zeitstempel in den `TIMESTAMP`-Spalten Ihres Downloads und verwenden Sie entweder ein `generate_next_last_download_timestamp`-Skript oder ein `modify_next_last_download_timestamp`-Skript, um einen UTC-Zeitstempel als nächsten Zeitstempel für den letzten Download zur Verfügung zu stellen.

## Snapshot-Synchronisation

Die Snapshot-Synchronisation einer Tabelle ist ein vollständiger Download aller relevanten Zeilen in der Tabelle, selbst wenn sie zuvor bereits heruntergeladen wurden. Dies ist die einfachste Synchronisationsmethode. Sie kann jedoch dazu führen, dass unnötig große Datenmengen ausgetauscht

werden, wodurch die Performance eingeschränkt werden kann und möglicherweise höhere Telefongebühren anfallen.

Sie können die Snapshot-basierte Synchronisation verwenden, um alle Zeilen der Tabelle herunterzuladen oder die Zeilen mit Partitionierung herunterzuladen. Siehe [„Partitionierte Zeilen in entfernten Datenbanken“ auf Seite 113](#).

### Einsatz der Snapshot-Synchronisation

Die Ad-Hoc-Methode ist gewöhnlich am sinnvollsten bei Tabellen, die die beiden folgenden Merkmale aufweisen:

- **Relativ wenige Zeilen** Wenn nur wenige Zeilen vorhanden sind, ist der Overhead für den Download aller Zeilen gering.
- **Zeilen ändern sich häufig** Wenn sich die meisten Zeilen in einer Tabelle häufig ändern, bringt der Ausschluss jener Zeilen, die sich seit der letzten Synchronisation geändert haben, nur geringe Vorteile.

Eine Tabelle, die eine Liste mit Wechselkursen enthält, könnte für diesen Ansatz geeignet sein, da es relativ wenige Währungen gibt, die sich jedoch häufig ändern. Abhängig von der Art des Geschäfts könnten auch eine Tabelle, die Preise enthält, eine Liste mit Zinssätzen oder aktuelle Nachrichten denkbare Einsatzmöglichkeiten hierfür darstellen.

### Implementierung der Snapshot-basierten Synchronisation

Beachten Sie die folgenden Punkte, wenn Sie die Snapshot-basierte Synchronisation implementieren.

- Lassen Sie die Uploadskripten undefiniert, außer wenn entfernte Benutzer die Werte aktualisieren.
- Wenn in der Tabelle Zeilen gelöscht werden dürfen, schreiben Sie ein `download_delete_cursor`-Skript, das alle Zeilen oder zumindest alle nicht mehr benötigten Zeilen aus der entfernten Tabelle löscht. Beim letzteren Vorgehen sollten Sie nicht die Zeilen aus der konsolidierten Datenbank löschen, sondern diese vielmehr für einen Löschvorgang kennzeichnen. Sie müssen die Zeilenwerte kennen, um diese aus der entfernten Datenbank löschen zu können.

Siehe [„download\\_delete\\_cursor-Skripten“ auf Seite 267](#).

- Schreiben Sie ein `download_cursor`-Skript, das alle Zeilen auswählt, die Sie in die entfernte Tabelle einbeziehen wollen.

### Zeilen löschen bei Verwendung der Snapshot-Synchronisation

Sie sollten aber nicht die Zeilen aus der konsolidierten Datenbank löschen, sondern diese vielmehr für einen Löschvorgang kennzeichnen. Sie müssen die Zeilenwerte kennen, um diese aus der entfernten Datenbank löschen zu können. Wählen Sie mit dem `download_cursor`-Skript nur die nicht markierten Zeilen und mit dem `download_delete_cursor`-Skript nur die markierten Zeilen aus.

Das `download_delete_cursor`-Skript wird vor dem `download_cursor`-Skript ausgeführt. Wenn eine Zeile in den Download einbezogen werden soll, brauchen Sie keine Zeile mit demselben Primärschlüssel in die Löschliste aufzunehmen. Wenn eine Downloadzeile am entfernten Standort empfangen wird, ersetzt sie eine vorhandene Zeile mit demselben Primärschlüssel.

Siehe „Skripten für den Download von Zeilen“ auf Seite 265.

### Alternatives Lösungsverfahren

Anstatt die Zeilen einer entfernten Datenbank mit einem `download_cursor`-Skript zu löschen, können Sie auch der entfernten Anwendung gestatten, den Löschvorgang durchzuführen. Beispielsweise können Sie unmittelbar nach der Synchronisation der Anwendung gestatten, SQL-Anweisungen auszuführen, mit denen die nicht mehr benötigten Zeilen gelöscht werden.

Die von der Anwendung gelöschten Zeilen werden bei der nächsten Synchronisation ganz normal an den MobiLink-Server übertragen. Sie können dies aber verhindern, indem Sie die Anweisung `STOP SYNCHRONIZATION DELETE` ausführen. Zum Beispiel:

```
STOP SYNCHRONIZATION DELETE;  
DELETE FROM table-name  
  WHERE expiry_date < CURRENT_TIMESTAMP;  
COMMIT;  
START SYNCHRONIZATION DELETE;
```

Siehe „`download_delete_cursor`-Skripten“ auf Seite 267.

### Snapshot-Beispiel

Die Tabelle `ULProduct` in der Beispielanwendung wird durch Snapshot-Synchronisationen gewartet. Die Tabelle enthält relativ wenige Zeilen und aus diesem Grund gibt es wenig Overhead bei der Snapshot-Synchronisation.

1. Es gibt kein Uploadskript. Dies spiegelt die Geschäftsentscheidung wider, dass in den entfernten Datenbanken keine Produkte hinzugefügt werden können.
2. Es gibt kein `download_delete_cursor`-Skript, da davon ausgegangen wird, dass Produkte nicht aus der Liste gelöscht werden.
3. Das `download_cursor`-Skript wählt den Produktbezeichner, den Preis und den Namen jedes aktuellen Produkts aus. Falls das Produkt bereits vorhanden ist, wird der Preis in der entfernten Tabelle aktualisiert. Ist das Produkt neu, wird eine Zeile in die entfernte Tabelle eingefügt.

```
SELECT prod_id, price, prod_name  
FROM ULProduct
```

Ein weiteres Beispiel der Snapshot-Synchronisation in einer Tabelle mit wenigen Zeilen finden Sie unter „Synchronisieren der Vertriebsmitarbeiter im Beispiel `Contact`“ [*MobiLink - Erste Orientierung*].

## Partitionierte Zeilen in entfernten Datenbanken

Jede entfernte MobiLink-Datenbank verfügt unter Umständen über eine jeweils andere Teilmenge der Daten aus der konsolidierten Datenbank. Sie können Ihre Synchronisationsskripten so schreiben, dass die Daten unter entfernten Datenbanken **partitioniert** werden.

Die Verteilung kann disjunkt sein oder Überlappungen enthalten. Wenn zum Beispiel jeder Mitarbeiter seinen eigenen Kundenkreis hat und es keine gemeinsamen Kunden gibt, ist die Verteilung **disjunkt**. Wenn es gemeinsame Kunden gibt, die in mehreren entfernten Datenbanken vorhanden sind, enthält die Verteilung **Überlappungen**.

Die Aufteilung wird in den Skripten `download_cursor` und `download_delete_cursor` für die Tabelle implementiert, die die Zeilen festlegt, die in die entfernte Datenbank übertragen werden sollen. Jedes dieser Skripte verwendet einen MobiLink-Benutzernamen als Parameter. Durch die Definition Ihrer Skripten mit diesem Parameter in der WHERE-Klausel erhält jeder Benutzer die jeweils richtigen Zeilen.

## Unabhängige Verteilung

Die Verteilung wird für jede an der Synchronisation beteiligte Tabelle durch die Skripten `download_cursor` und `download_delete_cursor` gesteuert. Diese Skripten verwenden zwei Parameter, einen Zeitstempel für den letzten Download und den MobiLink-Benutzernamen, der im zu synchronisierenden Aufruf angegeben wird.

Beachten Sie die folgenden Richtlinien, um eine Tabelle zwischen entfernten Datenbanken zu partitionieren:

- Beziehen Sie eine Spalte in die Tabellendefinition ein, die den Benutzernamen für die Synchronisation mit der konsolidierten Datenbank enthält. Diese Spalte braucht nicht in die entfernten Datenbanken übertragen zu werden.
- Fügen Sie in die WHERE-Klausel der Skripten `download_cursor` und `download_delete_cursor` eine Bedingung mit der Forderung ein, dass diese Spalte mit dem Skriptparameter übereinstimmt.

Der Skriptparameter wird durch einen benannten Parameter im Skript dargestellt. Folgendes `download_cursor`-Skript teilt zum Beispiel eine Tabelle `Contact` anhand der Mitarbeiterkennung `emp_id` auf.

```
SELECT id, contact_name
FROM Contact
WHERE last_modified >= {ml s.last_table_download}
AND emp_id = {ml s.username}
```

### Siehe auch

- „`download_cursor` (Tabellenereignis)“ auf Seite 329
- „`download_delete_cursor` (Tabellenereignis)“ auf Seite 331
- „Synchronisieren der Kunden im Beispiel `Contact`“ [*MobiLink - Erste Orientierung*]
- „Synchronisieren der Ansprechpartner im Beispiel `Contact`“ [*MobiLink - Erste Orientierung*]

### Beispiel

Die Tabellen mit dem Primärschlüsselpool in der Beispielanwendung `CustDB` werden verwendet, um jeder entfernten Datenbank ihre eigene Gruppe von Primärschlüsselwerten zu übermitteln. Diese Methode wird verwendet, um mehrfach vorhandene Primärschlüssel zu vermeiden. Siehe „[Primärschlüsselpools](#)“ auf Seite 122.

Diese Methode setzt voraus, dass die Tabellen des Primärschlüsselpools zwischen den entfernten Datenbanken unabhängig aufgeteilt werden.

Eine der Schlüsselpooltabellen ist die Tabelle `ULCustomerIDPool`, die Primärschlüsselwerte für die einzelnen Benutzer enthält, damit diese einen Kunden hinzufügen können. Die Tabelle hat folgende drei Spalten:

- **pool\_cust\_id** Ein Primärschlüsselwert, der in der Tabelle ULCustomer verwendet wird. Dies ist die einzige Spalte, die in die entfernte Datenbank übertragen wird.
- **pool\_emp\_id** Der Mitarbeiter, der Eigentümer dieses Primärschlüssels ist.
- **last\_modified** Diese Tabelle wird auf der Grundlage der Spalte last\_modified mit der Zeitstempelmethode gewartet.

Weitere Hinweise zur Zeitstempelsynchronisation finden Sie unter „[Implementieren zeitstempelbasierter Downloads](#)“ auf Seite 107.

Das download\_cursor-Skript für diese Tabelle sieht wie folgt aus:

```
SELECT pool_cust_id
FROM ULCustomerIDPool
WHERE last_modified >= {ml s.last_table_download}
AND pool_emp_id = {ml s.username}
```

## Partitionen mit Überlappungen

Einige Tabellen in Ihrer Datenbank können Zeilen enthalten, die zu mehreren entfernten Datenbanken gehören. Jede entfernte Datenbank enthält eine Teilmenge der Zeilen in der konsolidierten Datenbank und diese Teilmenge überlappt mit anderen entfernten Datenbanken. Dies ist häufig bei Kundentabellen der Fall. Im vorliegenden Beispiel besteht eine Viele-zu-Viele-Beziehung zwischen der Tabelle und den entfernten Datenbanken, und es gibt gewöhnlich eine Tabelle, die diese Beziehung repräsentiert. Die Skripten für die Ereignisse download\_cursor und download\_delete\_cursor müssen die per Download in die Beziehungstabelle zu übertragende Tabelle durch einen Join verknüpfen.

### Beispiel

Die Beispielanwendung CustDB verwendet diese Methode für die Tabelle ULOrder. Die Tabelle ULEmpCust enthält die Informationen der Viele-zu-Viele-Beziehung zwischen ULCustomer und ULEmployee.

Jede einzelne entfernte Datenbank empfängt nur die Zeilen aus der Tabelle ULOrder, für die der Wert in der Spalte emp\_id mit dem MobiLink-Benutzernamen übereinstimmt.

In der Version von SQL Anywhere sieht das download\_cursor-Skript für ULOrder der Anwendung CustDB folgendermaßen aus:

```
SELECT o.order_id, o.cust_id, o.prod_id,
       o.emp_id, o.disc, o.quant, o.notes, o.status
FROM ULOrder o , ULEmpCust ec
WHERE o.cust_id = ec.cust_id
AND ec.emp_id = {ml s.username}
AND ( o.last_modified >= {ml s.last_table_download}
      OR ec.last_modified >= {ml s.last_table_download} )
AND ( o.status IS NULL
      OR o.status != 'Approved' )
AND ( ec.action IS NULL )
```

Dieses Skript ist relativ komplex. Es veranschaulicht, dass die Abfrage, die eine Tabelle in der entfernten Datenbank definiert, mehrere Tabellen in der konsolidierten Datenbank einbeziehen kann. Das Skript liest alle Zeilen von ULOrder ein, für die Folgendes gilt:

- Die Spalte cust\_id in ULOrder stimmt mit der Spalte cust\_id in ULEmpCust überein.
- Die Spalte emp\_id in ULEmpCust stimmt mit dem Benutzernamen für die Synchronisation überein.
- Die letzte Änderung der Reihenfolge oder der Mitarbeiter-Kundenbeziehung fand später statt als zum Zeitpunkt der letzten Synchronisation des Benutzers.
- Der Status ist alles andere als **Approved** (Genehmigt).

Die Aktionsspalte von ULEmpCust dient dazu, Spalten für Löschvorgänge zu kennzeichnen. Bei NULL gilt die Zeile als vollständig aktiv (nicht gelöscht).

Das download\_delete\_cursor-Skript sieht folgendermaßen aus:

```
SELECT o.order_id, o.cust_id, o.prod_id, o.emp_id, o.disc, o.quant, o.notes,
o.status
FROM ULOrder o, dba.ULEmpCust ec
WHERE o.cust_id = ec.cust_id
AND ( ( o.status = 'Approved' AND o.last_modified >= {ml
s.last_table_download} )
OR ( ec.action = 'D' ) )
AND ec.emp_id = {ml s.username}
```

Dieses Skript löscht alle Zeilen mit dem Status Approved (Genehmigt) aus der entfernten Datenbank.

## Fremdschlüsseltabellen partitionieren

Das im vorherigen Abschnitt gegebene Beispiel veranschaulicht, wie Tabellen basierend auf Kriterien in einer anderen Tabelle verteilt werden. Siehe [„Partitionen mit Überlappungen“ auf Seite 115](#).

Einige Tabellen in der entfernten Datenbank haben möglicherweise unabhängige oder sich überschneidende Teilmengen, doch sie enthalten keine Spalte, die die Teilmenge bestimmt. Hierbei handelt es sich um Fremdschlüsseltabellen, die gewöhnlich einen Fremdschlüssel (oder eine Reihe von Fremdschlüsseln) haben, die andere Tabellen referenzieren. Die referenzierte Tabelle enthält eine Spalte, die die korrekte Teilmenge bestimmt.

In diesem Fall müssen die Skripten download\_cursor und download\_delete\_cursor die referenzierten Tabellen verknüpfen und eine WHERE-Klausel enthalten, die die Zeilen auf die korrekte Teilmenge begrenzt.

Ein Beispiel finden Sie in den Download-Skripten der der Tabelle Customer unter [„Synchronisieren der Ansprechpartner im Beispiel Contact“ \[MobiLink - Erste Orientierung\]](#).

## Reine Upload- und reine Download-Synchronisation

Standardmäßig wird eine Zwei-Wege-Synchronisation ausgeführt: die Daten werden übertragen und heruntergeladen. Sie haben jedoch die Möglichkeit, ausschließlich einen Upload oder einen Download durchzuführen.

**Hinweis**

Unter diesem Thema finden Sie Informationen über die Einrichtung der Synchronisation für reinen Upload oder reinen Download, wenn Sie das MobiLink-Synchronisationssystem in Ihrer Datenbank erstellen. Sie können reinen Upload oder reinen Download auch festlegen, wenn Sie ein Synchronisationsmodell in Sybase Central erstellen.

**Entfernte SQL Anywhere-Datenbanken**

- **Upload** Verwenden Sie für reine Upload-Synchronisationen die dbmlsync-Option -uo oder die erweiterte Option UploadOnly. Siehe:
  - „dbmlsync-Option -uo“ [[MobiLink - Clientadministration](#)]
  - „Erweiterte Option UploadOnly (uo)“ [[MobiLink - Clientadministration](#)]
- **Download** Verwenden Sie für reine Download-Synchronisationen die dbmlsync-Option -ds oder die erweiterte Option DownloadOnly. Siehe:
  - „dbmlsync-Option -ds“ [[MobiLink - Clientadministration](#)]
  - „Erweiterte Option DownloadOnly (ds)“ [[MobiLink - Clientadministration](#)]

Entfernte SQL Anywhere-Datenbanken können auch reine Download-Publikationen verwenden. Diese Download-Methode unterscheidet sich von den reinen Download-Synchronisationen. Siehe „Reine Download-Publikationen“ [[MobiLink - Clientadministration](#)].

**Entfernte UltraLite-Datenbanken**

- **Upload** Sie führen eine reine Upload-Synchronisation mithilfe des Synchronisationsparameters Upload Only aus.  
  
Siehe „Synchronisationsparameter Upload Only“ [[UltraLite - Datenbankverwaltung](#)].
- **Download** Sie führen eine reine Download-Synchronisation mithilfe des Synchronisationsparameters Download Only aus.  
  
Siehe „Synchronisationsparameter Download Only“ [[UltraLite - Datenbankverwaltung](#)].

## Eindeutige Primärschlüssel

Jede zu synchronisierende Tabelle muss einen Primärschlüssel haben und der Primärschlüssel für jede zu synchronisierende Tabelle muss über alle synchronisierten Datenbanken hinweg eindeutig sein. Die Werte von Primärschlüsseln sollten nicht aktualisiert werden.

Es ist häufig sinnvoll, eine einzelne Spalte als Primärschlüssel für Tabellen zu verwenden. Jedem Kunden sollte zum Beispiel ein eindeutiger Identifizierungswert zugeordnet sein. Wenn alle Verkäufer in einer Umgebung arbeiten, in der sie eine direkte Verbindung zur Datenbank aufrechterhalten können, können diese Zahlen einfach zugeordnet werden. Sobald ein neuer Kunde in die Kundentabelle eingefügt wird, fügen Sie automatisch einen Primärschlüsselwert hinzu, der größer ist als der vorangegangene Wert.

In einer getrennten Umgebung ist es nicht einfach, eindeutige Werte für Primärschlüssel zuzuordnen, wenn neue Zeilen eingefügt werden. Wenn ein Verkäufer einen neuen Kunden einfügt, führt er diese

Aktion in einer entfernten Kopie der Kundentabelle aus. Sie müssen verhindern, dass andere Verkäufer, die andere Kopien der Kundentabelle bearbeiten, denselben Kunden-Identifizierungswert verwenden.

In diesem Abschnitt werden die folgenden Möglichkeiten beschrieben, über alle synchronisierten Datenbanken hinweg eindeutige Primärschlüssel zu generieren.

- „Zusammengesetzte Schlüssel“
- „UUIDs“
- „GLOBAL AUTOINCREMENT“
- „Primärschlüsselpools“

## Zusammengesetzte Schlüssel

Die entfernte MobiLink-ID definiert eine entfernte Datenbank eindeutig in einem Synchronisationssystem. Daher lassen sich eindeutige Primärschlüssel einfach erstellen, indem Sie einen zusammengesetzten Primärschlüssel erstellen, der die entfernte MobiLink-ID als Teil seines Wertes einbezieht. Wenn Sie eindeutige MobiLink-Benutzernamen verwalten, können Sie den Benutzernamen anstelle der entfernten ID verwenden.

### Siehe auch

- „Entfernte IDs“ [*MobiLink - Clientadministration*]

## UUIDs

Sie können sicherstellen, dass Primärschlüssel in SQL Anywhere-Datenbanken eindeutig sind, indem Sie mit der Funktion `newid()` universell eindeutige Werte für Ihren Primärschlüssel erstellen. Die sich ergebenden UUIDs können mit der Funktion `uuidtostr()` in eine Zeichenfolge und mit der Funktion `strtouuid()` wieder zurück konvertiert werden.

UUIDs, auch GUIDs genannt, sind auf allen Computern eindeutig. Die Werte sind jedoch völlig willkürlich und können daher nicht verwendet werden, um festzustellen, wann ein Wert hinzugefügt wurde, oder um die Reihenfolge der Werte zu ermitteln. UUID-Werte sind auch erheblich größer als die Werte, die von anderen Methoden benötigt werden (einschließlich `global autoincrement`), und benötigen sowohl in den Primär- als auch den Fremdschlüsseltabellen mehr Tabellenplatz. Indizes für Tabellen, die UUIDs verwenden, sind auch weniger effizient.

### Siehe auch

SQL Anywhere-Datenbanken:

- „Der Standardwert `NEWID`“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „`NEWID`-Funktion [Verschiedene]“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]
- „`UNIQUEIDENTIFIER`-Datentyp“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]

UltraLite-Datenbanken:

- „Eindeutigkeit des Primärschlüssels in UltraLite“ [*UltraLite - Datenbankverwaltung*]
- „`NEWID`-Funktion [Verschiedene]“ [*UltraLite - Datenbankverwaltung*]

## Beispiel

Die folgende SQL Anywhere-Anweisung CREATE TABLE erstellt einen Primärschlüssel, der universell eindeutig ist:

```
CREATE TABLE customer (
  cust_key UNIQUEIDENTIFIER NOT NULL
        DEFAULT NEWID( ),
  rep_key VARCHAR(5),
  PRIMARY KEY(cust_key))
```

## GLOBAL AUTOINCREMENT

In SQL Anywhere- und UltraLite-Datenbanken können Sie den Standardspaltenwert auf GLOBAL AUTOINCREMENT einstellen. Sie können diesen Standardwert bei jeder Spalte verwenden, in der Sie eindeutige Werte aufrechterhalten wollen, aber er ist vor allem für Primärschlüssel nützlich.

GLOBAL AUTOINCREMENT-Werte werden in zusammenhängenden Wertebereichen zwischen entfernten Datenbanken aufgeteilt. Die Menge der möglichen Werte ist endlich. Je größer also die einzelnen Bereiche sind, desto weniger Bereiche stehen zur Verfügung. Achten Sie sorgfältig darauf, die richtige Bereichsgröße für Ihre Anforderungen festzulegen. Es ist möglich, dass sich ein Bereich erschöpft, aber Sie können dies feststellen und einen neuen Bereich zuordnen. Siehe [So werden Standardwerte gewählt auf Seite 121](#).

## GLOBAL AUTOINCREMENT-Spalten verwenden

Sie können den Standardspaltenwert auf GLOBAL AUTOINCREMENT setzen, um eindeutige Werte beizubehalten.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Kontext und Bemerkungen

Achten Sie sorgfältig darauf, die richtige Bereichsgröße für Ihre Anforderungen festzulegen. Es ist möglich, dass sich ein Bereich erschöpft, aber Sie können dies feststellen und einen neuen Bereich zuordnen. Siehe [So werden Standardwerte gewählt auf Seite 121](#).

### Aufgabe

1. Deklarieren Sie die Spalte als GLOBAL AUTOINCREMENT-Spalte.

Wenn Sie GLOBAL AUTOINCREMENT definieren, wird die Domäne von Werten bei dieser Spalte geteilt. Jede Teilmenge enthält dieselbe Anzahl von Werten. Wenn Sie zum Beispiel die Partitionsgröße einer Ganzzahlspalte in einer Datenbank auf 1000 festlegen, erstreckt sich eine Partition von 1001 bis 2000, die folgende von 2001 bis 3000 usw.

Siehe „DEFAULT GLOBAL AUTOINCREMENT“ auf Seite 120.

2. Legen Sie den Wert für `global_database_id` fest.

SQL Anywhere- und UltraLite-Datenbanken liefern Standardwerte in einer Datenbank nur von der Partition, die eindeutig durch diese Datenbanknummer gekennzeichnet ist. Wenn Sie etwa der Datenbank im oben stehenden Beispiel die Kennnummer 10 zugeordnet haben und die Partitionsgröße 1000 beträgt, werden die Standardwerte in dieser Datenbank aus dem Bereich 10001–11000 ausgewählt. Eine andere Kopie der Datenbank mit der Kennnummer 11 liefert Standardwerte für dieselbe Spalte im Bereich 11001–12000.

Siehe „[Globale Datenbank-IDs](#)“ auf Seite 120.

## Ergebnisse

Die Spalte wird als GLOBAL AUTOINCREMENT-Spalte festgelegt.

## DEFAULT GLOBAL AUTOINCREMENT

Sie können Standardwerte in Ihrer Datenbank einstellen, indem Sie die Spalteneigenschaften in Sybase Central auswählen oder die Klausel `DEFAULT GLOBAL AUTOINCREMENT` in eine `CREATE TABLE`- oder `ALTER TABLE`-Anweisung aufnehmen.

Wahlweise kann die Partitionsgröße in Klammern unmittelbar nach dem `AUTOINCREMENT`-Schlüsselwort angegeben werden. Die Partitionsgröße kann jede positive Ganzzahl sein, obwohl dieser Wert normalerweise so eingeteilt wird, dass seine Größe kaum jemals überschritten werden kann.

Bei Spalten des Typs `INT` oder `UNSIGNED INT` ist die Standard-Partitionsgröße  $2^{16} = 65536$ ; bei Spalten anderen Typs ist die Standard-Partitionsgröße  $2^{32} = 4294967296$ . Da diese Standardwerte möglicherweise unpassend sind, besonders wenn Ihre Spalten nicht vom Typ `INT` oder `BIGINT` sind, wird empfohlen, dass Sie die Partitionsgröße explizit angeben.

Beispiel: Die folgende SQL-Anweisung erstellt eine einfache Tabelle mit zwei Spalten, einer Ganzzahl, die eine Kunden-Identifizierungsnummer angibt, und einer Zeichenfolge, die den Namen des Kunden enthält. Die Partitionsgröße wurde auf 5000 festgelegt und das wäre geeignet für eine Anwendungsdatenbank, bei der in den einzelnen entfernten Datenbanken nur wenige neue Zeilen eingefügt werden.

```
CREATE TABLE customer (  
  id      INT          DEFAULT GLOBAL AUTOINCREMENT (5000),  
  name    VARCHAR(128) NOT NULL,  
  PRIMARY KEY (id)  
)
```

## Siehe auch

- SQL Anywhere: „[CREATE TABLE-Anweisung](#)“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]
- UltraLite: „[CREATE TABLE-Anweisung \[UltraLite\]](#)“ [*UltraLite - Datenbankverwaltung*]

## Globale Datenbank-IDs

Wenn Sie das Deployment für eine Anwendung durchführen, müssen Sie jeder Datenbank eine andere Identifizierungsnummer zuordnen. Es gibt mehrere Möglichkeiten, die Kennnummern zu erstellen und zu

verteilen. Eine Methode besteht darin, die Werte in eine Tabelle zu platzieren und die entsprechende Zeile beim Download in die einzelnen Datenbanken anhand einer anderen eindeutigen Eigenschaft zuzuordnen, wie zum Beispiel einer entfernten ID.

**Festlegen der globalen Datenbank-Identifizierungsnummer** In SQL Anywhere legen Sie die globale ID einer Datenbank fest, indem Sie den Wert der öffentlichen Option `global_database_id` bestimmen. Die ID muss eine nicht negative Ganzzahl sein.

In UltraLite legen Sie die globale ID einer Datenbank fest, indem Sie die Option `global_id` bestimmen.

### Siehe auch

- „`global_database_id`-Option“ [[SQL Anywhere Server - Datenbankadministration](#)]
- „`UltraLite`-Option `global_database_id`“ [[UltraLite - Datenbankverwaltung](#)]

### So werden Standardwerte gewählt

Die globale Datenbank-ID wird in SQL Anywhere mit der öffentlichen Option `global_database_id` festgelegt, in UltraLite mit der Option `global_id`.

Die Option `global_database_id` muss in jeder Datenbank als eindeutige, positive Ganzzahl eingestellt sein. Der Bereich von Standardwerten für eine bestimmte Tabelle liegt zwischen  $pn + 1$  und  $p(n + 1)$ , wobei  $p$  die Partitionsgröße und  $n$  der Wert der globalen Datenbank-ID ist. Wenn zum Beispiel die Partitionsgröße 1000 ist und die globale Datenbank-ID den Wert 3 hat, liegt der Bereich zwischen 3001 und 4000.

SQL Anywhere und UltraLite wählen Standardwerte anhand folgender Regeln:

- Wenn die Spalte in der aktuellen Partition keine Werte enthält, ist der erste Standardwert  $pn + 1$ , wobei  $p$  für die Partitionsgröße und  $n$  für den Wert der globalen Datenbank-ID steht.
- Wenn die Spalte in der aktuellen Partition Werte enthält, die alle weniger als  $p(n + 1)$  betragen, wird der nächste Standardwert um eins größer als der vorherige Höchstwert in diesem Bereich sein.
- Standardspaltenwerte sind von Werten, die sich außerhalb der aktuellen Partition befinden, nicht betroffen, d.h. von Nummern kleiner als  $pn + 1$  oder größer als  $p(n + 1)$ . Solche Werte können vorhanden sein, wenn sie von einer anderen Datenbank über MobiLink-Synchronisation repliziert wurden.

Wenn die globale Datenbank-ID auf den Standardwert 2147483647 gesetzt ist, wird NULL in die Spalte eingetragen. Falls NULL nicht zulässig ist, wird durch den Einfügeversuch ein Fehler verursacht. Dies geschieht etwa, wenn die Spalte im Primärschlüssel der Tabelle enthalten ist.

Da die globale Datenbank-ID kein negativer Wert sein darf, sind die gewählten Werte immer positiv. Die maximale Identifizierungsnummer wird nur durch den Spaltendatentyp und die Partitionsgröße beschränkt.

Null-Standardwerte werden auch generiert, wenn der Vorrat von Werten innerhalb der Partition aufgebraucht ist. In diesem Fall sollte der Datenbank eine neue globale Datenbank-ID zugewiesen werden, damit Standardwerte aus einer anderen Partition gewählt werden können. Der Versuch, NULL einzufügen, bewirkt einen Fehler, wenn die Spalte NULL nicht zulässt. Um festzustellen, ob der Vorrat von ungenutzten Werten zu Ende geht, und um diese Situation in SQL Anywhere-Datenbanken zu beheben, erstellen Sie ein Ereignis vom Typ `GlobalAutoincrement`.

Sollten die Werte in einer bestimmten Partition zu Ende gehen, können Sie dieser Datenbank eine neue globale Datenbank-ID zuordnen. Sie können neue Datenbank-Identifizierungsnummern auf verschiedene Arten zuordnen. Eine Möglichkeit ist, einen Pool von nicht benutzten Datenbank-ID-Werten einzurichten. Dieser Pool wird auf dieselbe Weise wie ein Pool von Primärschlüsseln verwaltet. Siehe [„Primärschlüsselpools“ auf Seite 122](#).

Sie können einen Event-Handler einstellen, um den Datenbankadministrator automatisch zu benachrichtigen (oder um eine andere Aktion auszuführen), wenn die Partition beinahe aufgebraucht ist. Weitere Hinweise zu SQL Anywhere-Datenbanken finden Sie unter [„Triggerbedingungen für Ereignisse“ \[SQL Anywhere Server - Datenbankadministration\]](#).

### Siehe auch

- [„Globale Datenbank-IDs“ auf Seite 120](#)
- SQL Anywhere: „global\_database\_id-Option“ [[SQL Anywhere Server - Datenbankadministration](#)]
- UltraLite: „UltraLite-Option global\_database\_id“ [[UltraLite - Datenbankverwaltung](#)]

### Beispiel

In SQL Anywhere setzt die folgende Anweisung die Datenbank-ID auf 20.

```
SET OPTION PUBLIC.global_database_id = 20
```

Wenn die Partitionsgröße für eine bestimmte Spalte 5000 beträgt, werden die Standardwerte für diese Datenbank aus dem Bereich 100001-105000 ausgewählt.

## Primärschlüsselpools

Eine effiziente Methode zur Lösung des Problems mit den eindeutigen Primärschlüsseln besteht darin, jedem Datenbankbenutzer einen Pool von Primärschlüsselwerten zuzuweisen, die er bei Bedarf verwenden kann. Sie können zum Beispiel jedem Verkäufer 100 neue Identifizierungswerte zuteilen. Jeder Verkäufer kann Werte aus seinem eigenen Pool frei an neue Kunden vergeben.

Im Folgenden finden Sie einen Überblick darüber, wie Sie einen Primärschlüsselpool implementieren.

1. Fügen Sie in der konsolidierten Datenbank und in jeder entfernten Datenbank eine neue Tabelle ein, die den neuen Primärschlüsselpool enthält. Zusätzlich zu einer Spalte für den eindeutigen Wert in der konsolidierten Datenbank müssen diese Tabellen eine Spalte für den Benutzernamen enthalten, der identifiziert, wer das Recht hat, den Wert zuzuweisen.
2. Schreiben Sie in der konsolidierten Datenbank eine gespeicherte Prozedur, die sicherstellt, dass jedem Benutzer genügend neue Identifizierungswerte zugeordnet sind. Ordnen Sie entfernten Benutzern, die viele neue Einträge einfügen oder nur selten synchronisieren, mehr neue Werte zu.
3. Schreiben Sie ein download\_cursor-Skript, um die neuen Werte auszuwählen, die den einzelnen Benutzern zugeordnet wurden, und sie in die entfernte Datenbank herunterzuladen. Siehe [„Implementieren zeitstempelbasierter Downloads“ auf Seite 107](#).
4. Ändern Sie die Anwendung, die die entfernte Datenbank benutzt, sodass sie einen Wert aus dem Pool verwendet, wenn ein Benutzer eine neue Zeile einfügt. Die Anwendung muss dann den Wert aus dem Pool löschen, damit sie nicht mehrfach benutzt wird.

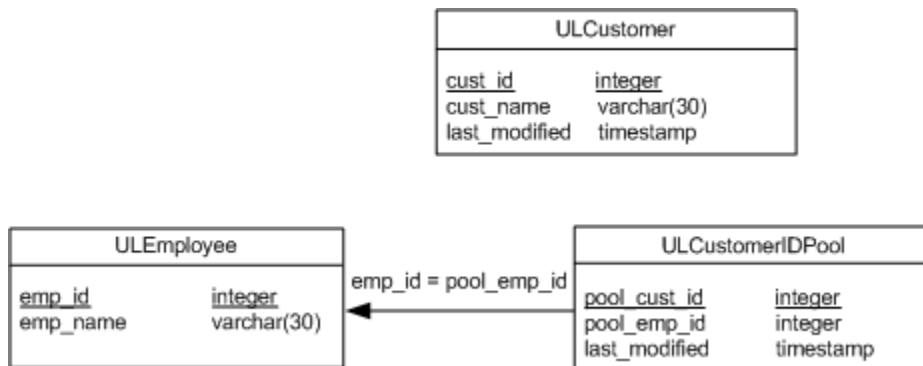
5. Schreiben Sie ein upload\_delete-Skript, um die gelöschten Schlüssel hochzuladen. Der MobiLink-Server löscht dann Zeilen aus dem konsolidierten Werte-Pool, die der Benutzer in seinem persönlichen Werte-Pool in der entfernten Datenbank gelöscht hat.
6. Schreiben Sie ein end\_upload-Skript, das die gespeicherte Prozedur aufruft, die den Werte-Pool wartet. Dies hat zur Wirkung, dass weitere Werte in den Pool des Benutzers eingefügt werden, um die beim Download gelöschten Werte zu ersetzen.

### Beispiel

Die Beispielanwendung CustDB gestattet es entfernten Benutzern, Kunden hinzuzufügen. Jede neue Zeile muss unbedingt einen eindeutigen Primärschlüsselwert haben. Dennoch ist die Verbindung jeder entfernten Datenbank zum Zeitpunkt der Dateneingabe getrennt.

Die Tabelle ULCustomerIDPool enthält eine Liste mit Primärschlüsselwerten, die von jeder entfernten Datenbank verwendet werden können. Außerdem füllt die gespeicherte Prozedur ULCustomerIDPool\_maintain den Pool wieder auf, wenn die Werte verbraucht werden. Die Wartungsprozeduren werden von einem end\_upload-Skript auf Tabellenebene aufgerufen und die Pools in jeder entfernten Datenbank werden von den Skripten download\_cursor und upload\_delete gewartet.

1. Die Tabelle ULCustomerIDPool in der konsolidierten Datenbank enthält den Pool neuer Identifizierungsnummern für Kunden. Sie hat keine direkte Verbindung mit der Tabelle ULCustomer.



2. Die Prozedur ULCustomerIDPool\_maintain aktualisiert die Tabelle ULCustomerIDPool in der konsolidierten Datenbank. Der folgende Beispielcode gilt für eine konsolidierte SQL Anywhere-Datenbank.

```

CREATE PROCEDURE ULCustomerIDPool_maintain ( IN syncuser_id INTEGER )
BEGIN
    DECLARE pool_count INTEGER;

    -- Determine how many ids to add to the pool
    SELECT COUNT(*) INTO pool_count
    FROM ULCustomerIDPool
    WHERE pool_emp_id = syncuser_id;

    -- Top up the pool with new ids
    WHILE pool_count < 20 LOOP
        INSERT INTO ULCustomerIDPool ( pool_emp_id )
        VALUES ( syncuser_id );
    END LOOP;
END
  
```

```

        SET pool_count = pool_count + 1;
    END LOOP;
END

```

Diese Prozedur zählt die Nummern, die dem aktuellen Benutzer derzeit zugeordnet sind, und fügt neue Zeilen ein, sodass dem Benutzer genügend Identifizierungsnummern für Kunden zur Verfügung stehen.

Diese Prozedur wird am Ende des Uploads vom end\_upload-Tabellenskript für die Tabelle ULCustomerIDPool aufgerufen. Das Skript sieht wie folgt aus:

```
CALL ULCustomerIDPool_maintain( {ml s.username} )
```

3. Das download\_cursor-Skript für die Tabelle ULCustomerIDPool liest die neuen Nummern in die entfernte Datenbank ein.

```

SELECT pool_cust_id
FROM ULCustomerIDPool
WHERE pool_emp_id = {ml s.username}
AND last_modified >= {ml s.last_table_download}

```

4. Um einen neuen Kunden einzufügen, muss die Anwendung, die die entfernte Datenbank verwendet, eine nicht verwendete Identifizierungsnummer auswählen, diese Nummer aus dem Pool löschen und die neuen Kundendaten mit dieser Identifizierungsnummer einfügen. Folgende Embedded SQL-Funktion für eine UltraLite-Anwendung ruft eine neue Kundennummer aus dem Pool ab:

```

bool CDemoDB::GetNextCustomerID( void )
/*****
{
    short ind;

    EXEC SQL SELECT min( pool_cust_id )
    INTO :m_CustID:ind FROM ULCustomerIDPool;
    if( ind < 0 ) {
        return false;
    }
    EXEC SQL DELETE FROM ULCustomerIDPool
    WHERE pool_cust_id = :m_CustID;
    return true;
}

```

## Konfliktverarbeitung

### Vorsicht

Primärschlüssel dürfen in synchronisierten Tabellen nie aktualisiert werden. Durch Aktualisieren von Primärschlüsseln wird ihr Zweck außer Kraft gesetzt, da der Schlüssel die einzige Möglichkeit darstellt, dieselbe Zeile in unterschiedlichen Datenbanken (entfernt und konsolidiert) zu identifizieren und Konflikte zu erkennen.

Konflikte können beim Upload von Zeilen in die konsolidierte Datenbank entstehen und sind nicht dasselbe wie Fehler. Wenn Konflikte auftreten können, sollten Sie einen Prozess definieren, um die korrekten Werte zu ermitteln oder wenigstens den Konflikt zu protokollieren. Die Konfliktverarbeitung ist ein integraler Bestandteil einer sauber geschriebenen Anwendung.

Standardmäßig gilt Folgendes:

- Wenn bei einem Einfügeversuch entdeckt wird, dass die Zeile bereits eingefügt wurde, wird ein Fehler gemeldet.
- Wenn ein Löschesuch entdeckt, dass die Zeile bereits gelöscht wurde, wird der zweite Löschesuch ignoriert.

Wenn Sie eine andere Verhaltensweise wünschen, können Sie sie ein oder mehrere Upload-Ereignisses definieren, die in diesem Abschnitt beschrieben werden.

Während der Download-Phase einer Synchronisation kommt es in der entfernten Datenbank nicht zu Konflikten. Wenn eine eingelesene Zeile einen neuen Primärschlüssel enthält, werden die Werte in eine neue Zeile eingefügt. Wenn der Primärschlüssel mit dem einer vorhandenen Zeile übereinstimmt, werden die Werte in der Zeile aktualisiert.

### Beispiel

Benutzer1 beginnt mit einem Lager von zehn Artikeln. Er verkauft drei davon und aktualisiert den Wert des Lagers von Remote1 auf sieben Artikel. Benutzer2 verkauft vier Artikel und aktualisiert das Lager von Remote2 auf sechs. Wenn Remote1 synchronisiert, wird die konsolidierte Datenbank auf sieben aktualisiert. Wenn Remote2 synchronisiert, wird ein Konflikt festgestellt, da der Wert des Lagers nicht mehr zehn ist. Um diesen Konflikt programmiertechnisch zu lösen, brauchen Sie drei Zeilenwerte:

1. Den aktuellen Wert in der konsolidierten Datenbank.
2. Den neuen Zeilenwert, den Remote2 ausgelesen hat
3. Den alten Zeilenwert, den Remote2 bei der letzten Synchronisation erhalten hat

In diesem Fall würde die Geschäftslogik folgende Berechnung verwenden, um den neuen Lagerwert zu ermitteln und den Konflikt zu lösen:

```
current consolidated - (old remote - new remote)
-> 7 - (10-6) = 3
```

Weitere Beispiele zum Umgang mit Konflikten finden Sie unter:

- „Synchronisieren der Produkte im Beispiel Contact“ [[MobiLink - Erste Orientierung](#)]

## Konflikterkennung

Wenn ein MobiLink-Client eine aktualisierte Zeile an den MobiLink-Server sendet, enthält diese Zeile nicht nur die neuen aktualisierten Werte (das Post-Image), sondern auch eine Kopie der alten Zeilenwerte (das Pre-Image), die entweder beim letzten Download abgerufen wurden oder aus den Zeilenwerten, die vor dem ersten Upload dieser Zeile existierten, bestehen. Wenn das Pre-Image nicht mit den aktuellen Werten in der konsolidierten Datenbank übereinstimmt, wird ein Konflikt festgestellt.

Der MobiLink-Server erkennt Konflikte nur dann, wenn ein Skript vom Typ `upload_fetch` oder `upload_fetch_column_conflict` angewendet wird. Wenn Sie `upload_fetch` verwenden, werden

kollidierende Aktualisierungen als Konflikt markiert. Wenn Sie `upload_fetch_column_conflict` verwenden, werden nur Konflikte bei Aktualisierungen derselben Spalte markiert.

Außerdem können Sie eine beliebige Konflikterkennung und -lösung einrichten, indem Sie eine gespeicherte Prozedur für `upload_update` verwenden. Die Konflikterkennung und -lösung wird vollständig von dem Skript gesteuert, sodass MobiLink keinen Konflikt auslöst.

## Konflikterkennung mit `upload_fetch`- oder `upload_fetch_column_conflict`-Skripten

Wenn Sie ein `upload_fetch`- oder `upload_fetch_column_conflict`-Skript für eine Tabelle definieren, vergleicht der MobiLink-Server das Pre-Image einer hochgeladenen Aktualisierung mit den Werten in der vom Skript zurückgegebenen Zeile mit denselben Primärschlüsselwerten. Der MobiLink-Server erkennt einen Konflikt, wenn Werte im Pre-Image nicht mit den aktuellen Werten in der konsolidierten Datenbank übereinstimmen. Der Server ruft die Skripten `upload_old_row_insert` und `upload_new_row_insert` auf, gefolgt vom Skript `resolve_conflict`, sobald ein Konflikt festgestellt wird.

### Hinweis

Ein Fehler tritt auf, wenn die Skripten `upload_old_row_insert` und `upload_new_row_insert` während eines Konflikts nicht definiert sind. Definieren Sie diese Skripten unter Verwendung der Anweisung `-- {ml_ignore}` als ignoriert, wenn sie für die Synchronisationstabelle nicht mehr benötigt werden.

Der Unterschied zwischen den Skripten `upload_fetch` und `upload_fetch_column_conflict` liegt in den Kriterien, die der MobiLink-Server verwendet, um einen Konflikt zu erkennen. Bei einem `upload_fetch`-Skript werden alle Unterschiede zwischen der abgerufenen Zeile und der Pre-Image-Zeile als Konflikt behandelt. Bei einem `upload_fetch_column_conflict`-Skript werden nur die von der entfernten Datenbank aktualisierten Spalten zwischen der abgerufenen Zeile und der Pre-Image-Zeile verglichen. Mit anderen Worten: `upload_fetch` bietet zeilenbasierte Konflikterkennung und `upload_fetch_column_conflict` bietet spaltenbasierte Konflikterkennung.

Das `upload_fetch`-Skript wählt eine einzelne Datenzeile aus einer Tabelle einer konsolidierten Datenbank, die der zu aktualisierenden Zeile entspricht. Sie können dieses Skript auf zweierlei Art verwenden. Die erste Möglichkeit besteht darin, die Zeile mit denselben Primärschlüsseln und denselben Spaltenwerten auszuwählen wie das hochgeladene Pre-Image. Wenn keine Zeile zurückgegeben wird, erkennt der MobiLink-Server einen Konflikt. Diese Verwendung des Skripts hat die folgende Syntax (wobei `pk1`, `pk2`, ... Primärschlüsselspalten sind und `col1`, `col2`, ... Nicht-Primärschlüsselspalten):

```
SELECT pk1, pk2, ...col1, col2, ...  
FROM table-name  
WHERE pk1 = {ml r.pk1} AND pk2 = {ml r.pk2} ...  
AND col1 = {ml o.col1} AND col2 = {ml o.col2} ...
```

### Hinweis

Diese Methode der Konflikterkennung kann nicht auf synchronisierte Tabellen angewendet werden, die große Binärspalten wie BLOB und CLOB enthalten.

Die zweite Möglichkeit besteht darin, die Zeile mit demselben Primärschlüssel auszuwählen und den MobiLink-Server die abgerufene Zeile mit dem hochgeladenen Pre-Image vergleichen zu lassen. Wenn

sich Spalten unterscheiden, erkennt der MobiLink-Server einen Konflikt. Diese Methode funktioniert mit allen synchronisierbaren Spaltentypen:

```
SELECT pk1, pk2, ...col1, col2, ...
FROM table-name
WHERE pk1 = {ml r.pk1} AND pk2 = {ml r.pk2} ...
```

Siehe „upload\_fetch (Tabellenereignis)“ auf Seite 437.

Das Ereignis upload\_fetch\_column\_conflict ist dasselbe wie upload\_fetch, außer dass bei ihm der MobiLink-Server nur dann einen Konflikt für eine Zeile erkennt, wenn seit der letzten Synchronisation dieselbe Spalte in der entfernten Datenbank und in der konsolidierten Datenbank aktualisiert wurde. Verschiedene Benutzer können dieselbe Zeile aktualisieren, ohne einen Konflikt zu generieren, sofern sie nicht dieselbe Spalte aktualisieren. Das Ereignis upload\_fetch\_column\_conflict kann nur auf Synchronisationstabellen angewendet werden, die keine BLOBs enthalten.

Wenn Sie ein upload\_fetch\_column\_conflict-Skript verwenden und kein Konflikt erkannt wird, stammen die in Ihr upload\_update-Skript übergebenen Zeilenwerte entweder aus dem Upload der entfernten Datenbank oder aus den aktuellen Werten der konsolidierten Datenbank gemäß Ihrem upload\_fetch\_column\_conflict-Skript. Der Wert der entfernten Datenbank wird für Spalten verwendet, die in der entfernten Datenbank aktualisiert wurden. Andernfalls wird der aktuelle Wert der konsolidierten Datenbank verwendet. Mit anderen Worten: Nur die Spalten, die in der entfernten Datenbank aktualisiert wurden, werden in der konsolidierten Datenbank aktualisiert.

Siehe „upload\_fetch\_column\_conflict (Tabellenereignis)“ auf Seite 439.

Für jede Tabelle in der entfernten Datenbank kann ein upload-fetch- oder ein upload\_fetch\_column\_conflict-Skript vorhanden sein.

### Die Zeile in der konsolidierten Datenbank sperren

Es ist möglich, dass eine Zeile in der konsolidierten Datenbank geändert wird, nachdem das upload\_fetch-Skript einen Konflikt erkannt hat und bevor die Konfliktlösung abgeschlossen ist. Um dieses Problem, das zu fehlerhaften Daten führen würde, zu vermeiden, können Sie die Skripten upload\_fetch oder upload\_fetch\_column\_conflicts mit einer Zeilensperre implementieren.

In konsolidierten SQL Anywhere-Datenbanken können Sie die Schlüsselwörter UPDLOCK oder HOLDLOCK verwenden, wobei UPDLOCK besser für den Mehrbenutzerbetrieb ist. Beispiel:

```
SELECT column-names from table-name WITH (UPDLOCK)
WHERE where-clause
```

Bei Microsoft SQL Server verwenden Sie HOLDLOCK. Beispiel:

```
SELECT column-names FROM table-name WITH (HOLDLOCK)
WHERE where-clause
```

Bei Adaptive Server Enterprise verwenden Sie HOLDLOCK. Beispiel:

```
SELECT column-names FROM table-name
HOLDLOCK
WHERE where-clause
```

## Beispiel

Sie definieren ein `upload_fetch`-Skript. Der MobiLink-Server ruft mit dem Skript die aktuelle Zeile in der konsolidierten Datenbank ab und vergleicht sie mit dem Pre-Image der aktualisierten Zeile. Wenn die beiden Zeilen identische Werte enthalten, liegt ein Konflikt vor. Enthalten die beiden Zeilen verschiedene Werte, wurde ein Konflikt erkannt und MobiLink ruft erst Skripten `upload_old_row_insert` und `upload_new_row_insert` und anschließend `resolve_conflict` auf.

Siehe „[Konfliktlösung mit resolve\\_conflict-Skripten](#)“ auf Seite 128.

## Konflikterkennung mit `upload_update`-Skripten

Sie definieren keine Skripten für `upload_fetch`, `upload_fetch_column_conflict`, `upload_old_row_insert`, `upload_new_row_insert` und `resolve_conflict`. Stattdessen erstellen Sie eine gespeicherte Prozedur zur Behandlung der Konflikterkennung und Konfliktlösung und rufen sie im Skript `upload_update` auf.

Siehe „[Konfliktlösung mit upload\\_update-Skripten](#)“ auf Seite 130.

## Konfliktlösung

Es gibt mehrere Möglichkeiten, Konflikte zu lösen:

- Konflikte bei ihrem Auftreten mit temporären oder permanenten Tabellen und einem `resolve_conflict`-Skript lösen.

Siehe „[Konfliktlösung mit resolve\\_conflict-Skripten](#)“ auf Seite 128.

- Konflikte bei ihrem Auftreten mit einem `upload_update`-Skript lösen.

Siehe „[Konfliktlösung mit upload\\_update-Skripten](#)“ auf Seite 130.

- Alle Konflikte gleichzeitig mit einem `end_upload`-Skript einer Tabelle lösen.

Siehe „[end\\_upload \(Tabellenereignis\)](#)“ auf Seite 361.

## Konfliktlösung mit `resolve_conflict`-Skripten

Wenn der MobiLink-Server bei einem `upload_fetch`-Skript einen Konflikt entdeckt, kommt es zu folgenden Ereignissen.

- Der MobiLink-Server fügt alte Zeilenwerte ein, die mithilfe des im `upload_old_row_insert`-Skript definierten Cursors aus der entfernten Datenbank eingelesen wurden. Gewöhnlich werden die alten Werte in eine temporäre Tabelle eingefügt.

Siehe „[upload\\_old\\_row\\_insert \(Tabellenereignis\)](#)“ auf Seite 445.

- Der MobiLink-Server fügt die neuen Zeilenwerte ein, die mithilfe des im `upload_new_row_insert`-Skript definierten Cursors aus der entfernten Datenbank eingelesen wurden. Gewöhnlich werden die neuen Werte in eine temporäre Tabelle eingefügt.

Siehe „[upload\\_new\\_row\\_insert \(Tabellenereignis\)](#)“ auf Seite 443.

- Der MobiLink-Server führt das resolve\_conflict-Skript aus. In diesem Skript können Sie eine gespeicherte Prozedur aufrufen oder eine Sequenz von Schritten festlegen, um den Konflikt mithilfe der neuen und alten Zeilenwerte zu lösen.

### Siehe auch

- „resolve\_conflict (Tabellenereignis)“ auf Seite 417

### Beispiel

Im folgenden Beispiel erstellen Sie Skripten für sechs Ereignisse und erstellen dann eine gespeicherte Prozedur.

- Im Skript begin\_synchronization erstellen Sie zwei temporäre Tabellen namens contact\_new und contact\_old. (Sie können dies auch im Skript begin\_connection vornehmen.)
- Das Skript upload\_fetch erkennt den Konflikt.
- Liegt ein Konflikt vor, füllen Sie in den Skripten upload\_old\_row\_insert und upload\_new\_row\_insert die beiden temporären Tabellen mit neuen und alten Daten, die aus der entfernten Datenbank übertragen wurden.
- Das Skript resolve\_conflict ruft die gespeicherte Prozedur MLResolveContactConflict auf, um den Konflikt zu lösen.

| Ereignis              | Skript                                                                                                                                                                            |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| begin_synchronization | <pre>CREATE TABLE #contact_new(   id INTEGER,   location CHAR(36),   contact_date DATE); CREATE TABLE #contact_old(   id INTEGER,   location CHAR(36),   contact_date DATE)</pre> |
| upload_fetch          | <pre>SELECT id, location, contact_date FROM contact WHERE id = {ml r.id}</pre>                                                                                                    |
| upload_old_row_insert | <pre>INSERT INTO #contact_new( id, location, contact_date ) VALUES ( {ml r.id}, {ml r.location}, {ml r.contact_date} )</pre>                                                      |
| upload_new_row_insert | <pre>INSERT INTO #contact_old( id, location, contact_date ) VALUES ( {ml r.id}, {ml r.location}, {ml r.contact_date} )</pre>                                                      |
| resolve_conflict      | <pre>CALL MLResolveContactConflict( )</pre>                                                                                                                                       |
| end_synchronization   | <pre>DROP TABLE #contact_new; DROP TABLE #contact_old</pre>                                                                                                                       |

Die gespeicherte Prozedur MLResolveContactConflict lautet folgendermaßen:

```

CREATE PROCEDURE MLResolveContactConflict( )
BEGIN
  --update the consolidated database only if the new contact date
  --is later than the existing contact date
  UPDATE contact c
    SET c.contact_date = cn.contact_date
    FROM #contact_new cn
    WHERE c.id = cn.id
      AND cn.contact_date > c.contact_date;
  --cleanup
  DELETE FROM #contact_new;
  DELETE FROM #contact_old;
END

```

## Konfliktlösung mit upload\_update-Skripten

Anstelle des Skripts resolve\_conflict können Sie im upload\_update-Skript eine gespeicherte Prozedur für die Konfliktlösung verwenden. Bei dieser Methode müssen Sie Konflikte programmtechnisch erkennen und lösen.

Die gespeicherte Prozedur muss alle Spalten annehmen, einschließlich der neuen (Post-Image) und alten (Pre-Image) Werte.

Das upload\_update-Skript kann wie folgt aussehen:

```

{CALL UpdateProduct(
  {ml o.id}, {ml o.name}, {ml o.desc}, {ml r.name}, {ml r.desc}
)}

```

Die gespeicherte Prozedur UpdateProduct kann wie folgt aussehen:

```

CREATE PROCEDURE UpdateProduct(
  @id INTEGER,
  @preName VARCHAR(20),
  @preDesc VARCHAR(200),
  @postName VARCHAR(20),
  @postDesc VARCHAR(200) )
BEGIN
  UPDATE product
    SET name = @postName, description = @postDesc
    WHERE id = @id
      AND name = @preName
      AND description = @preDesc
  IF @@rowcount=0 THEN
    // A conflict occurred: handle resolution here.
  END IF
END

```

Diese Methode ist häufig einfacher zu verwalten als das Lösen von Konflikten mit resolve\_conflict-Skripten, da nur ein Skript gepflegt und verwaltet werden muss und die gesamte Programmlogik in einer gespeicherten Prozedur enthalten ist. Der Code der gespeicherten Prozedur kann jedoch komplizierter sein, wenn die Tabellenspalten NULL oder BLOBs/CLOBs enthalten. Zudem gibt es bei einigen RDBMS, die von konsolidierten MobiLink-Datenbanken unterstützt werden, Einschränkungen bezüglich der Größe der Werte, die an gespeicherte Prozeduren übergeben werden können.

**Siehe auch**

- „Konflikterkennung mit upload\_update-Skripten“
- „upload\_update (Tabellenereignis)“
- „Konfliktlösung mit resolve\_conflict-Skripten“
- „COALESCE-Funktion [Verschiedene]“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]
- @@rowcount unter „Globale Variablen“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]

**Beispiel**

Die folgende gespeicherte Prozedur sp\_update\_my\_customer enthält Programmlogik für die Erkennung und Lösung von Konflikten. Sie akzeptiert alte und neue Spaltenwerte. Dieses Beispiel verwendet SQL Anywhere-Funktionen. Das Skript kann wie folgt implementiert werden:

```
{CALL sp_update_my_customer(
  {ml o.cust_1st_pk},
  {ml o.cust_2nd_pk},
  {ml o.first_name},
  {ml o.last_name},
  {ml o.nullable_col},
  {ml o.last_modified},
  {ml r.first_name},
  {ml r.last_name},
  {ml r.nullable_col},
  {ml r.last_modified}
)}
CREATE PROCEDURE sp_update_my_customer(
  @cust_1st_pk      INTEGER,
  @cust_2nd_pk      INTEGER,
  @old_first_name   VARCHAR(100),
  @old_last_name    VARCHAR(100),
  @old_nullable_col VARCHAR(20),
  @old_last_modified DATETIME,
  @new_first_name   VARCHAR(100),
  @new_last_name    VARCHAR(100),
  @new_nullable_col VARCHAR(20),
  @new_last_modified DATETIME
)
BEGIN
  DECLARE @current_last_modified DATETIME;
  // Detect a conflict by checking the number of rows that are
  // affected by the following update. The WHERE clause compares
  // old values uploaded from the remote database to current values in
  // the consolidated database. If the values match, there is
  // no conflict. The COALESCE function returns the first non-
  // NULL expression from a list, and is used in this case to
  // compare values for a nullable column.

  UPDATE my_customer
  SET first_name      = @new_first_name,
      last_name       = @new_last_name,
      nullable_col    = @new_nullable_col,
      last_modified   = @new_last_modified

  WHERE cust_1st_pk   = @cust_1st_pk
     AND cust_2nd_pk   = @cust_2nd_pk
     AND first_name    = @old_first_name
     AND last_name     = @old_last_name
     AND COALESCE(nullable_col, '') = COALESCE(@old_nullable_col, '')
     AND last_modified = @old_last_modified;
```

```

...
// Use the @@rowcount global variable to determine
// the number of rows affected by the update. If @@rowcount=0,
// a conflict has occurred. In this example, the database with
// the most recent update wins the conflict. If the consolidated
// database wins the conflict, it retains its current values
// and no action is taken.

IF( @@rowcount = 0 ) THEN
// A conflict has been detected. To resolve it, use business
// logic to determine which values to use, and update the
// consolidated database with the final values.

    SELECT last_modified INTO @current_last_modified
    FROM my_customer WITH( HOLDLOCK )
    WHERE cust_1st_pk=@cust_1st_pk
        AND cust_2nd_pk=@cust_2nd_pk;

    IF( @new_last_modified > @current_last_modified ) THEN
// The remote database has won the conflict: use the values it
// uploaded.

        UPDATE my_customer
        SET first_name      = @new_first_name,
            last_name       = @new_last_name,
            nullable_col    = @new_nullable_col,
            last_modified   = @new_last_modified
        WHERE cust_1st_pk  = @cust_1st_pk
            AND cust_2nd_pk = @cust_2nd_pk;

    END IF;
END IF;
END;

```

## Löschungen

Wenn eine Zeile aus der konsolidierten Datenbank gelöscht wird, muss dieser Vorgang aufgezeichnet werden, damit die betreffende Zeile von einem `download_delete_cursor`-Skript explizit ausgewählt und auch aus allen entfernten Datenbanken gelöscht werden kann. Dies kann durch logische Löschungen oder Schattentabellen erfolgen.

- **Logische Löschungen** Bei dieser Methode wird die Zeile nicht gelöscht. Nicht mehr benötigte Daten werden in einer Statusspalte als inaktiv markiert. Die `WHERE`-Klausel der Skripten `download_cursor` und `download_delete_cursor` und die meisten Anwendungsabfragen müssen sich auf den Status der Zeile beziehen.

Diese Methode wird in der Beispielanwendung `CustDB` eingesetzt, in der die Spalte `ULEmpCust.action` ein "D" für "Delete" (Löschen) enthält. Die Skripten verwenden diesen Wert, um den Datensatz aus der entfernten Datenbank zu löschen, und löschen am Ende der Synchronisation den Datensatz auch aus der konsolidierten Datenbank. `CustDB` verwendet diese Methode auch für die Tabelle `ULOrder`. Im Beispiel `Contact` wird die Methode auf die Tabellen `Customer`, `Contact` und `Product` angewendet.

Bei der Unterstützung des MobiLink-Synchronisationsmodells für logische Löschungen wird davon ausgegangen, dass eine logische Löschspalte sich nur in der konsolidierten Datenbank und nicht in der entfernten Datenbank befindet. Wenn ein konsolidiertes Schema in ein neues entferntes Schema

kopiert wird, lassen Sie alle Spalten aus, die mit den logischen Löschspalten in den Synchronisationseinstellungen des Modells übereinstimmen. Bei einem neuen Modell wird der Standardspaltenname *gelöscht*.

So fügen Sie den Namen der logischen Löschspalte dem entfernten Schema hinzu:

1. Im **Assistenten zum Erstellen eines Synchronisationsmodells** klicken Sie auf der Seite **Download-Löschungen** auf die Option **Logische Löschungen verwenden**.
2. Benennen Sie die logische Löschspalte um, sodass sie mit keinem Spaltennamen in der konsolidierten Datenbank übereinstimmt.
3. Nachdem der Assistent fertig ist, aktualisieren Sie das entfernte Schema und behalten die Standardtabellenauswahl bei. Der Name der logischen Löschspalte erscheint in der Schemaänderungsliste und kann dem entfernten Schema hinzugefügt werden.

#### Hinweis

Sie müssen die Spaltenzuordnung für die logische Löschspalte der entfernten Datenbank auf die logische Löschspalte der konsolidierten Datenbank festlegen.

- **Schattentabellen** Bei dieser Methode erstellen Sie eine Schattentabelle, in der die Primärschlüsselwerte von gelöschten Zeilen gespeichert werden. Wenn eine Zeile gelöscht wird, kann ein Trigger die Schattentabelle anfüllen. Das `download_delete_cursor`-Skript kann diese Schattentabelle dann verwenden, um Zeilen aus den entfernten Datenbanken zu löschen. Die Schattentabelle benötigt hierzu lediglich die Primärschlüsselspalten aus der eigentlichen Tabelle.

#### Siehe auch

- „`download_delete_cursor`-Skripten“ auf Seite 267

## Synchronisation von Löschungen vorübergehend stoppen

Verwenden Sie die STOP SYNCHRONIZATION DELETE-Anweisung, um die automatische Protokollierung von Änderungen an denjenigen Tabellen oder Spalten vorübergehend zu stoppen, die Teil einer Publikation mit einer Synchronisationssubskription sind und deshalb normalerweise während der nächsten Synchronisation in die konsolidierte Datenbank hochgeladen werden.

#### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

#### Kontext und Bemerkungen

Wenn eine STOP SYNCHRONIZATION DELETE-Anweisung abläuft, wird kein Löschvorgang synchronisiert, der anschließend auf dieser Verbindung ausgeführt wird. Die Wirkung hält so lange an, bis eine START SYNCHRONIZATION DELETE-Anweisung ausgeführt wird. Die Auswirkungen verschachteln sich nicht. Das heißt, anschließende Ausführungen von STOP SYNCHRONIZATION DELETE nach der ersten haben keine zusätzliche Auswirkung.

Diese Funktionalität kann z. B. dazu verwendet werden, unübliche Korrekturen durchzuführen. Sie sollte aber mit Vorsicht verwendet werden, da sie Teile der automatischen Synchronisationsfunktionalität deaktiviert. Diese Technik ist eine praktische Alternative zum Löschen der notwendigen Zeilen mithilfe eines `download_delete_cursor`-Skripts.

## Aufgabe

1. Führen Sie die folgende Anweisung aus, um die automatische Protokollierung der Löschvorgänge zu stoppen.

```
STOP SYNCHRONIZATION DELETE
```

2. Löschen Sie die erforderlichen Zeilen mithilfe der `DELETE`-Anweisung aus den synchronisierten Daten. Schreiben Sie diese Änderungen fest.
3. Starten Sie die Protokollierung der Löschvorgänge mithilfe der folgenden Anweisung erneut.

```
START SYNCHRONIZATION DELETE
```

## Ergebnisse

Die gelöschten Zeilen werden nicht zum MobiLink-Server gesendet und nicht aus der konsolidierten Datenbank gelöscht.

## Siehe auch

- SQL Anywhere-Clients: „[STOP SYNCHRONIZATION DELETE-Anweisung \[MobiLink\]](#)“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]
- UltraLite-Clients: „[STOP SYNCHRONIZATION DELETE-Anweisung \[UltraLite\]](#)“ [[UltraLite - Datenbankverwaltung](#)]
- SQL Anywhere-Clients: „[START SYNCHRONIZATION DELETE-Anweisung \[MobiLink\]](#)“ [[SQL Anywhere Server - SQL-Referenzhandbuch](#)]
- UltraLite-Clients: „[START SYNCHRONIZATION DELETE-Anweisung \[UltraLite\]](#)“ [[UltraLite - Datenbankverwaltung](#)]

# Fehlgeschlagene Downloads

## Nicht blockierende Downloadbestätigung verwenden

In der nicht blockierenden Downloadbestätigungs-Transaktion müssen Angaben darüber erfasst werden, welche Daten heruntergeladen werden. Diese Informationen sollten in den Skripten `publication_nonblocking_download_ack` oder `nonblocking_download_ack` aktualisiert werden, die nach der erfolgreichen Anwendung des Downloads in der entfernten Datenbank aufgerufen werden.

Wenn ein Fehler auftritt oder `SendDownloadAck` auf `OFF` gesetzt ist, werden diese nicht blockierenden Downloadbestätigungsskripten nicht aufgerufen und der Downloadzeitstempel wird nicht aktualisiert. Beim Testen der Synchronisationsskripten sollten Sie künstlich fehlgeschlagene Downloads verursachen, um sich zu vergewissern, dass Ihre Skripten mit einem fehlgeschlagenen Download umgehen können.

## Blockierende Downloadbestätigung verwenden

Die Unterstützung für die blockierende Downloadbestätigung wurde nicht mehr weiterentwickelt. Alle Downloadbestätigungen werden als nicht blockierend verarbeitet.

Siehe „Erweiterte Option SendDownloadAck (sa)“ [[MobiLink - Clientadministration](#)] und „Synchronisationsparameter Send Download Acknowledgement“ [[UltraLite - Datenbankverwaltung](#)].

## Wiederaufnahme fehlgeschlagener Downloads

Ein Download-Fehler wird durch einen Kommunikationsfehler während des Downloads verursacht oder wenn ein entfernter Benutzer versucht, den Download abzubrechen. Der MobiLink-Server nimmt Downloaddaten auf, die vom Client nicht empfangen wurden, um sie in einem neu startbaren Download verwenden zu können. Der Server gibt Downloaddaten erst frei, wenn eine der folgenden Situationen eintritt:

- Der Benutzer schließt den Download erfolgreich ab.
- Der Benutzer gibt eine neue Synchronisationsanforderung ein, ohne dass eine Wiederaufnahme versucht wurde.
- Der Cache wird für neue Downloads benötigt. Der älteste nicht erfolgreiche Download wird zuerst gelöscht.

MobiLink verfügt über Funktionen, die bei der Wiederherstellung nach einem Download-Fehler Unterstützung bieten und eine Übertragungswiederholung des gesamten Downloads verhindern. Für diese Funktion gibt es unterschiedliche Implementierungen für entfernte SQL Anywhere- und UltraLite-Datenbanken. Siehe „[mlsrv16-Option -ds](#)“ auf [Seite 59](#).

### Hinweis

Damit ein wiederaufnehmbarer Download-Versuch erfolgreich sein kann, muss sich die entfernte Datenbank mit demselben MobiLink-Server verbinden. Wenn die entfernte Datenbank sich mit einem anderen MobiLink-Server verbindet oder wenn der Download aus demselben MobiLink-Server verschoben wurde, schlägt die versuchte Wiederaufnahme fehl.

## Wann sollten Sie fehlgeschlagene Downloads wieder aufnehmen?

Wiederaufnehmbare Downloads werden häufiger benötigt, je schlechter die Netzwerkqualität wird und je größer die Downloads werden. Wenn Sie nur kleine Synchronisationen ausführen oder wenn die Synchronisation über ein LAN oder WLAN erfolgt, müssen Sie Downloads wahrscheinlich nicht wieder aufnehmen.

## Entfernte SQL Anywhere-Datenbanken

Wenn die Synchronisation während eines Downloads fehlschlägt, werden die heruntergeladenen Daten nicht in die entfernte Datenbank übernommen. Die erfolgreich übertragenen Download-Teile werden jedoch in einer temporären Datei auf dem entfernten Gerät gespeichert. Dbmlsync verwendet diese Datei, um eine langwierige Übertragungswiederholung von Daten zu vermeiden und den Systemzustand nach dem Download-Fehler wiederherzustellen.

Es gibt drei Möglichkeiten, diese Funktion zu implementieren. In allen Fällen wird dbmlsync abgebrochen und der wieder aufgenommene Download schlägt fehl, wenn neue Daten hochgeladen werden müssen.

- **-dc** Wenn ein Download fehlgeschlagen ist, können Sie den Download beim nächsten Aufruf von dbmlsync mit der Option -dc wieder aufnehmen. Falls ein Teil des fehlgeschlagenen Downloads übertragen wurde, überträgt der MobiLink-Server nur noch die restlichen Daten.

Weitere Hinweise finden Sie unter „dbmlsync-Option -dc“ [[MobiLink - Clientadministration](#)].

- **Erweiterte Option ContinueDownload (cd)** Wenn die erweiterte Option -cd in der Befehlszeile dbmlsync verwendet wird, hat sie dieselbe Funktion wie die Option -dc. Sie können diese Option auch in der Datenbank speichern oder mithilfe von sp\_hook\_dbmlsync\_set\_extended\_options in einer einzelnen Synchronisation verwenden.

Siehe „Erweiterte Option ContinueDownload (cd)“ [[MobiLink - Clientadministration](#)] und „sp\_hook\_dbmlsync\_set\_extended\_options“ [[MobiLink - Clientadministration](#)].

- **Einstiegsprozedur sp\_hook\_dbmlsync\_begin** Mit dem Parameter restart können Sie veranlassen, dass ein Download wieder aufgenommen wird. Sie erkennen, dass ein Download wieder aufgenommen werden kann, wenn der Parameter restartable download auf TRUE gesetzt ist. Wenn eine Download-Datei von einer bestimmten Größe vorhanden ist, können Sie auch Logik im Hook erstellen, um einen Download wieder aufzunehmen. Verwenden Sie dabei die Größe des neu startbaren Downloads.

Siehe „sp\_hook\_dbmlsync\_end“ [[MobiLink - Clientadministration](#)].

### Entfernte UltraLite-Datenbanken

Sie können das Verhalten von UltraLite-Anwendungen nach einem fehlgeschlagenen Download wie folgt steuern:

- Wenn Sie den Synchronisationsparameter Keep Partial Download bei der Synchronisation auf TRUE setzen und während des Downloads ein Fehler auftritt, übernimmt UltraLite den Teil der Änderungen, der übertragen wurde. UltraLite setzt den Synchronisationsparameter Partial Download Retained ebenfalls auf TRUE.

Die UltraLite-Datenbank ist zu diesem Zeitpunkt möglicherweise inkonsistent. Abhängig von Ihrer Anwendung wollen Sie möglicherweise sicherstellen, dass die Synchronisation erfolgreich beendet oder zurückgesetzt wird, bevor Sie eine Änderung der Daten zulassen. Siehe „Synchronisationsparameter Keep Partial Download“ [[UltraLite - Datenbankverwaltung](#)] und „Synchronisationsparameter Partial Download Retained“ [[UltraLite - Datenbankverwaltung](#)].

- Um den Download wieder aufzunehmen, setzen Sie den Synchronisationsparameter Resume Partial Download auf TRUE und synchronisieren erneut. Siehe „Synchronisationsparameter Resume Partial Download“ [[UltraLite - Datenbankverwaltung](#)].

Die neu gestartete Synchronisation führt keinen Upload aus und überträgt nur Änderungen, die vom fehlgeschlagenen Download übertragen worden wären. Das heißt, der fehlgeschlagene Download wird abgeschlossen, doch es werden keine Änderungen synchronisiert, die seit dem letzten Download-Versuch vorgenommen wurden. Um diese Änderungen zu erhalten, müssen Sie, nachdem der

fehlgeschlagene Download abgeschlossen ist, erneut eine Synchronisation durchführen, oder "Rollback Partial Download" aufrufen und mit "Resume Partial Download" mit dem Wert FALSE noch einmal synchronisieren.

Wenn Sie den Download erneut starten, werden automatisch zahlreiche Synchronisationsparameter der fehlgeschlagenen Synchronisation verwendet. Der Publikationsparameter wird z.B. ignoriert: Die Synchronisation überträgt die Publikationen, die vom ursprünglichen Download angefordert wurden. Lediglich die Parameter Resume Partial Download (muss auf TRUE gesetzt werden) und User Name müssen gesetzt werden. Außerdem werden Einstellungen für die folgenden Parameter berücksichtigt, falls diese gesetzt wurden:

- Keep Partial Download
- DisableConcurrency
- Observer
- User Data
- Um die Änderungen aus dem fehlgeschlagenen Download zurückzusetzen, ohne die Synchronisation wieder aufzunehmen, rufen Sie die Methode oder Funktion zum Zurücksetzen der Änderungen auf. Für Embedded SQL ist dies die Funktion ULRollbackPartialDownload. Für UltraLite-Komponenten ist dies eine Methode am Connection-Objekt.
  - **UltraLite.NET** [ULConnection.RollbackPartialDownload-Methode \[UltraLite.NET\] \[UltraLite - .NET-Programmierung\]](#)
  - **Embedded SQL** [ULRollbackPartialDownload-Methode \[UltraLite Embedded SQL\] \[UltraLite - C- und C++-Programmierung\]](#)

Sie können die Änderungen aus einem fehlgeschlagenen Download zurücksetzen, wenn die Synchronisation nicht abgeschlossen werden kann, z.B. weil der Server oder das Netzwerk nicht verfügbar ist, und Sie die Datenkonsistenz bewahren möchten, während der Endbenutzer weiter mit der Anwendung arbeiten kann.

Weitere Hinweise zu Kommunikationsfehlern finden Sie unter [Fehlermeldungen](#).

#### Hinweis

Wenn der Synchronisationsparameter send\_download\_ack auf TRUE festgelegt ist, wird die Einstellung für den wieder aufgenommenen Download ignoriert.

## Downloadbestätigung

Die Downloadbestätigung ist eine optionale Komponente der Synchronisation, bei der der Client den MobiLink-Server sofort informiert, wenn der Download erfolgreich in die entfernte Datenbank übernommen wurde. Sie wird für Deployments empfohlen, deren Geschäftslogik so schnell wie möglich agieren muss, wenn eine entfernte Quittung für einen Download empfangen wird. Es ist keine Downloadbestätigung erforderlich, um sicherzustellen, dass die Daten von der entfernten Datenbank empfangen wurden.

Es gibt zwei Modi der Downloadbestätigung: den blockierenden Modus, der nicht mehr weiterentwickelt wurde, und den nicht blockierenden Modus. Alle Downloadbestätigungen werden jetzt als nicht blockierend verarbeitet.

Zur Verwendung der Downloadbestätigung müssen Einstellungen auf dem Client und dem Server vorgenommen werden.

Auf dem Client legen Sie die Downloadbestätigung mit der erweiterten dbmlsync-Option SendDownloadACK oder dem UltraLite-Synchronisationsparameter "Send Download Acknowledgment" fest.

Auf dem Server gibt es zwei Verbindungsereignisse, mit deren Hilfe Sie die Zeit des letzten erfolgreichen Downloads in der konsolidierten Datenbank aufzeichnen können, wenn Sie die nicht blockierende Downloadbestätigung verwenden: publication\_nonblocking\_download\_ack und nonblocking\_download\_ack.

### Hinweis

Die Downloadbestätigung kann nicht zusammen mit wiederaufnehmbaren Downloads verwendet werden. Siehe „Wiederaufnahme fehlgeschlagener Downloads“ auf Seite 135.

### Siehe auch

- „publication\_nonblocking\_download\_ack (Verbindungsereignis)“ auf Seite 407
- „nonblocking\_download\_ack (Verbindungsereignis)“ auf Seite 401
- dbmlsync: „Erweiterte Option SendDownloadAck (sa)“ [*MobiLink - Clientadministration*]
- UltraLite: „Synchronisationsparameter Send Download Acknowledgement“ [*UltraLite - Datenbankverwaltung*]

## Ergebnismengen aus Aufrufen an gespeicherte Prozeduren

Sie können den Download einer Ergebnismenge aus einem Aufruf einer gespeicherten Prozedur durchführen. Beispiel: Sie benutzen ein download\_cursor-Skript für folgende Tabelle:

```
CREATE TABLE MyTable (  
    pk INTEGER PRIMARY KEY NOT NULL,  
    col1 VARCHAR(100) NOT NULL,  
    col2 VARCHAR(20) NOT NULL,  
    employee VARCHAR(100) NOT NULL,  
    last_modified TIMESTAMP NOT NULL DEFAULT TIMESTAMP  
)
```

Das Tabellenskript download\_cursor kann z.B. wie folgt aussehen:

```
SELECT pk, col1, col2  
FROM MyTable  
WHERE last_modified >= {ml s.last_table_download}  
AND employee = {ml s.username}
```

Wenn Sie Downloads in Tabelle MyTable durchführen wollen, um eine komplexere Unternehmenslogik einzusetzen, können Sie nun Ihr Skript wie folgt erstellen, wobei DownloadMyTable eine gespeicherte

Prozedur ist, die zwei Parameter übernimmt (Zeitstempel des letzten Downloads und MobiLink-Benutzername) und eine Ergebnismenge zurückgibt. (In diesem Beispiel wird aus Gründen der Portierbarkeit eine ODBC-Aufrufkonvention verwendet.):

```
{call DownloadMyTable( {ml s.last_table_download}, {ml s.username} )}
```

Im Folgenden finden Sie einige Beispiele für die verschiedenen unterstützten konsolidierten Datenbanken. Einzelheiten hierzu können Sie der Dokumentation Ihrer konsolidierten Datenbank entnehmen.

Das folgende Beispiel ist für SQL Anywhere-, Adaptive Server Enterprise- und Microsoft SQL Server geeignet.

```
CREATE PROCEDURE DownloadMyTable
    @last_dl_ts DATETIME,
    @u_name VARCHAR( 128 )
AS
BEGIN
    SELECT pk, col1, col2
    FROM MyTable
    WHERE last_modified >= @last_dl_ts
    AND employee = @u_name
END
```

Bei Oracle kann die Ergebnismenge von einem REF CURSOR-Parameter zurückgegeben werden, der in einer gespeicherten Prozedur definiert ist. Wenn Sie jedoch den SQL Anywhere 16 - Oracle-ODBC-Treiber verwenden, muss der REF CURSOR-Parameter als letzter in der Parameterliste der gespeicherten Prozedur definiert sein. Der REF CURSOR-Parameter kann als OUT oder IN OUT definiert werden. Die folgende gespeicherte Prozedur ist für Oracle geeignet.

```
create or replace procedure DownloadMyTable(
    v_last_dl_ts IN TIMESTAMP,
    v_user_name IN VARCHAR,
    v_ref_crshr OUT SYS_REF_CURSOR ) As
Begin
    Open v_ref_crshr For
        select pk, col1, col2
        from MyTable
        where last_modified >= v_last_dl_ts
        and employee = v_user_name;
End DownloadMyTable;
```

Verwenden Sie als Nächstes die gespeicherte Prozedur ml\_add\_table\_script, um einen Aufruf von DownloadMyTable als download\_cursor-Skript für die Synchronisationstabelle MyTable zu definieren:

```
CALL ml_add_table_script(
    'v1',
    'MyTable',
    'download_cursor',
    '{CALL DownloadMyTable(
        {ml s.last_table_download},{ml s.username} )}'
);
```

Bei Oracle nimmt die gespeicherte Prozedur DownloadMyTable nur zwei Parameter an, nicht drei, und der MobiLink-Server ruft die Ergebnismenge über den REF CURSOR-Parameter ab. REF CURSOR ist als letzter Parameter in der Definition der gespeicherten Prozedur definiert.

Folgendes Beispiel ist für IBM DB2 LUW geeignet.

```
CREATE PROCEDURE DownloadMyTable(  
    IN last_dl_ts TIMESTAMP,  
    IN u_name VARCHAR( 128 ) )  
    LANGUAGE SQL  
    MODIFIES SQL DATA  
    COMMIT ON RETURN NO  
    DYNAMIC RESULT SETS 1  
  
BEGIN  
    DECLARE c1, cursor WITH RETURN FOR  
        SELECT pk, col1, col2 FROM MyTable  
        WHERE last_modified >= last_dl_ts AND employee = u_name;  
    OPEN c1;  
END;
```

## Selbstreferenzierende Tabellen

Einige Tabellen sind selbstreferenzierend. Beispielsweise können in einer Mitarbeitertabelle in einer Spalte die Mitarbeiter aufgeführt werden und in einer anderen die Vorgesetzten jedes Mitarbeiters; desweiteren kann es eine Hierarchie der Vorgesetzten der Vorgesetzten geben. Diese Tabellen sind schwierig zu synchronisieren, da MobiLink standardmäßig alle Datenaktualisierungen in der entfernten Datenbank zusammenfügt. Dies ist zwar effizient, aber die Reihenfolge der Transaktionen geht verloren.

Sie können diese Situation auf zwei Arten regeln:

- Wenn Sie eine entfernte SQL Anywhere-Datenbank verwenden, können Sie mit der dbmlsync-Option -tu festlegen, dass jede Transaktion in der entfernten Datenbank als separate Transaktion gesendet werden soll.
- Fügen Sie eine Zuordnungstabelle hinzu, die Mitarbeiter Managern zuordnet, sodass die Reihenfolge der Transaktionen in der vormals selbstreferenzierenden Tabelle keine Rolle mehr spielt.

### Siehe auch

- „dbmlsync-Option -tu“ [[MobiLink - Clientadministration](#)]

## MobiLink-Isolationsstufen

MobiLink verbindet sich auf der bestmöglichen Isolationsstufe mit einer konsolidierten Datenbank, wenn die Isolationsstufen im RDBMS aktiviert sind. Die Standardisolationsstufen werden gewählt, um bei Gewährleistung der Datenkonsistenz beste Leistungsfähigkeit zu erzielen.

Im Allgemeinen verwendet MobiLink die Isolationsstufe SQL\_TXN\_READ\_COMMITTED für Uploads und, wenn möglich, Snapshot-Isolation für Downloads. Wenn die Snapshot-Isolation nicht verfügbar ist, verwendet MobiLink SQL\_TXN\_READ\_COMMITTED. Bei einem Download mit SQL\_TXN\_READ\_COMMITTED-Isolation besteht die Möglichkeit zu blockieren, bis eine andere Transaktion abgeschlossen ist. Ein solches Blockieren kann den Durchsatz von Synchronisationen deutlich verringern. Die Snapshot-Isolation verhindert, dass Downloads blockiert werden, bis Transaktionen in der konsolidierten Datenbank geschlossen wurden, unter der Annahme, dass der Download keine Aktualisierungen ausführt, was dringend empfohlen wird.

Bei der Snapshot-Isolation können Daten doppelt heruntergeladen werden (wenn z.B. bei längeren Transaktionen derselbe Snapshot längere Zeit verwendet wird). MobiLink-Clients sind jedoch in der Lage, dies automatisch zu verarbeiten, sodass der einzige Nachteil in der Übertragungsdauer und dem Verarbeitungsaufwand in der entfernten Datenbank liegt. Trotzdem wird empfohlen, Transaktionen mit langen Laufzeiten zu vermeiden.

Die Isolationsstufe 0 (READ COMMITTED) eignet sich für gewöhnlich nicht zur Synchronisation und kann Dateninkonsistenzen verursachen.

Die Isolationsstufe wird sofort eingestellt, nachdem eine Verbindung zur konsolidierten Datenbank hergestellt wurde. Es werden noch ein paar weitere Verbindungseinstellungen durchgeführt und dann wird die Transaktion festgeschrieben. Der COMMIT-Befehl wird von dem meisten RDBMS verlangt, damit die Isolationsstufe (und möglicherweise andere Einstellungen) wirksam werden können.

### **Konsolidierte SQL Anywhere-Datenbanken, Version 10 und höher**

SQL Anywhere Version 10 und höher unterstützen Snapshot-Isolation. Standardmäßig verwendet MobiLink die Isolationsstufe SQL\_TXN\_READ\_COMMITTED für Uploads und die Snapshot-Isolation für Downloads.

MobiLink kann die Snapshot-Isolation nur verwenden, wenn Sie sie in Ihrer konsolidierten SQL Anywhere-Datenbank aktivieren. Wenn die Snapshot-Isolation nicht aktiviert ist, verwendet MobiLink die Standardisolation SQL\_TXN\_READ\_COMMITTED.

Wenn es einer Datenbank ermöglicht wird, die Snapshot-Isolation zu verwenden, kann sich dies auf die Performance auswirken, da Kopien aller geänderten Zeilen aufbewahrt werden müssen, unabhängig von der Anzahl der Transaktionen, welche die Snapshot-Isolation verwenden. Siehe „[So aktivieren Sie die Snapshot-Isolation](#)“ [[SQL Anywhere Server - SQL-Benutzerhandbuch](#)].

Sie können die Snapshot-Isolation für Uploads mit der mlsrv16-Option -esu aktivieren und mit der mlsrv16-Option -dsd deaktivieren. Wenn Sie die MobiLink-Standardisolationsstufe in einem Verbindungsskript ändern müssen, sollten Sie dies in den Skripten begin\_upload oder begin\_download vornehmen. Wenn Sie die Standard-Isolationsstufe im Skript begin\_connection ändern, werden Ihre Einstellungen unter Umständen zu Beginn der Upload- und Downloadtransaktionen überschrieben.

Siehe „[mlsrv16-Option -esu](#)“ auf Seite 62 und „[mlsrv16-Option -dsd](#)“ auf Seite 60.

### **Konsolidierte SQL Anywhere-Datenbanken vor Version 10**

Wenn Sie eine frühere SQL Anywhere-Version als Version 10 verwenden, ist die MobiLink-Standardisolationsstufe SQL\_TXN\_READ\_COMMITTED. Sie können den Standardwert für die gesamte MobiLink-Sitzung im Skript begin\_connection ändern. Für den Upload und Download ändern Sie ihn in den Skripten begin\_upload bzw. begin\_download.

### **Konsolidierte Adaptive Server Enterprise-Datenbanken**

Bei Adaptive Server Enterprise ist die MobiLink-Standardisolationsstufe SQL\_TXN\_READ\_COMMITTED. Sie können den Standardwert für die gesamte MobiLink-Sitzung im Skript begin\_connection ändern. Für den Upload und Download ändern Sie ihn in den Skripten begin\_upload bzw. begin\_download.

## Konsolidierte Oracle-Datenbanken

Oracle unterstützt die Snapshot-Isolation, nennt sie jedoch READ COMMITTED. Standardmäßig verwendet MobiLink die Snapshot/READ COMMITTED-Isolationsstufe für Uploads und Downloads.

Sie können den Standardwert für die gesamte MobiLink-Sitzung im Skript `begin_connection` ändern. Für den Upload und Download ändern Sie ihn in den Skripten `begin_upload` bzw. `begin_download`.

Damit der MobiLink-Server die Snapshot-Isolation optimal nutzen kann, muss der vom MobiLink-Server verwendete Oracle-Account für die Oracle-Systemansicht `GV_$TRANSACTION` berechtigt sein. Ist dies nicht der Fall, wird eine Warnung ausgegeben und die Zeilen werden unter Umständen beim Download ausgelassen. Diese Berechtigung kann nur SYS erteilen. Die Oracle-Syntax zur Erteilung dieser Berechtigung lautet:

```
grant select on SYS.GV_$TRANSACTION to user-name;
```

## Konsolidierte Microsoft SQL Server-Datenbanken, Version 2005 und höher

Microsoft SQL Server 2005 unterstützt die Snapshot-Isolation. Standardmäßig verwendet MobiLink die Isolationsstufe `SQL_TXN_READ_COMMITTED` für Uploads und die Snapshot-Isolation für Downloads.

MobiLink kann die Snapshot-Isolation nur verwenden, wenn Sie sie in Ihrer konsolidierten SQL Server-Datenbank aktivieren. Wenn die Snapshot-Isolation nicht aktiviert ist, verwendet MobiLink die Standardisolation `SQL_TXN_READ_COMMITTED`. Ausführliche Informationen finden Sie in Ihrer SQL Server-Dokumentation.

Sie können die Snapshot-Isolation für Uploads mit der `mlsrv16`-Option `-esu` aktivieren und mit der `mlsrv16`-Option `-dsd` deaktivieren. Wenn Sie die MobiLink-Standardisolationsstufe in einem Verbindungsskript ändern müssen, sollten Sie dies in den Skripten `begin_upload` oder `begin_download` vornehmen. Wenn Sie die Standard-Isolationsstufe im Skript `begin_connection` ändern, werden Ihre Einstellungen unter Umständen zu Beginn der Upload- und Downloadtransaktionen überschrieben.

Siehe „[mlsrv16-Option -esu](#)“ auf Seite 62 und „[mlsrv16-Option -dsd](#)“ auf Seite 60.

Um die Snapshot-Isolation in SQL Server verwenden zu können, muss die Benutzer-ID, mit der Sie den MobiLink-Server mit der Datenbank verbinden, auf die SQL Server-Systemtabelle `SYS.DM_TRAN_ACTIVE_TRANSACTIONS` zugreifen dürfen. Wenn diese Berechtigung nicht erteilt wurde, verwendet MobiLink die Standardisolation `SQL_TXN_READ_COMMITTED`.

Wenn Ihre konsolidierte Datenbank auf einem Microsoft SQL Server läuft, auf dem auch andere Datenbanken ausgeführt werden, und Sie für Uploads oder Downloads die Snapshot-Isolation verwenden sowie Ihre Upload- oder Download-Skripten auf keine anderen Datenbanken auf dem Server zugreifen, sollten Sie die MobiLink-Serveroption `-dt` angeben. Diese Option bewirkt, dass MobiLink alle Transaktionen mit Ausnahme derjenigen in der aktuellen Datenbank ignoriert, wodurch potenziell der Durchsatz erhöht und die Duplikation von Zeilen beim Download verringert wird.

Siehe „[mlsrv16-Option -dt](#)“ auf Seite 60.

**Siehe auch**

- „mlsrv16-Option -dsd “ auf Seite 60
- „mlsrv16-Option -esu“ auf Seite 62
- „mlsrv16-Option -dt “ auf Seite 60
- „Der Synchronisationsprozess“ [*MobiLink - Erste Orientierung*]
- „Isolationsstufen und Konsistenz“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]
- „Snapshot-Isolation“ [*SQL Anywhere Server - SQL-Benutzerhandbuch*]

## MobiLink-Performance

Die folgende Liste enthält Vorschläge dazu, wie Sie die beste MobiLink-Performance erzielen.

**Test**

Die nachfolgenden Aspekte tragen zum Durchsatz Ihres Synchronisationssystems bei:

- der Gerätetyp, auf dem Ihre entfernten Datenbanken ausgeführt werden
- das Schema der entfernten Datenbanken
- das Datenvolumen und die Synchronisationsfrequenz Ihrer entfernten Datenbanken
- Netzwerkmerkmale (einschließlich HTTP, Proxys, Webserver und Relay Server)
- die Hardware, auf der der MobiLink-Server läuft
- Ihre Synchronisationsskripten
- das Aufkommen der gleichzeitig ablaufenden Synchronisationen
- der Typ der verwendeten konsolidierten Datenbank
- die Hardware, auf der die konsolidierte Datenbank ausgeführt wird
- die Aktivitäten der konsolidierten Datenbank, einschließlich aller Nicht-Synchronisationsaktivitäten
- das Schema der konsolidierten Datenbank

Das Durchführen von Tests ist besonders wichtig. Führen Sie vor dem Deployment Tests unter Verwendung derselben Hardware und desselben Netzwerks aus, das Sie für die Produktion verwenden wollen. Testen Sie zudem möglichst mit derselben Anzahl an entfernten Datenbanken, derselben Synchronisationsfrequenz und demselben Datenvolumen. Das MobiLink Replay Tool kann diese Tests unterstützen. Siehe „[Dienstprogramm MobiLink Replay \(mlreplay\)](#) “ auf Seite 702.

Experimentieren Sie während der Tests mit folgenden Performance-Tipps.

**Konflikte vermeiden**

Vermeiden Sie Konflikte, und maximieren Sie die Parallelität in Synchronisationsskripten.

Nehmen Sie zum Beispiel an, dass das Skript `begin_download` in einer Tabellenspalte die Download-Vorgänge mitzählt. Wenn mehrere Benutzer gleichzeitig synchronisieren, würde dieses Skript die Download-Vorgänge dieser Benutzer im Endeffekt serialisieren. Es ist besser, diesen Zähler in das Skript `begin_synchronization`, `end_synchronization` oder `prepare_for_download` zu integrieren, da diese Skripten unmittelbar vor dem Festschreiben aufgerufen werden. Datenbanken werden somit nur kurzzeitig gesperrt. Ein noch besserer Ansatz wäre, nur nach entfernter ID zu zählen und die Gesamtzahl zu einem späteren Zeitpunkt per Abfrage zu ermitteln.

Siehe [Konfliktsituationen auf Seite 149](#).

Weitere Hinweise zur Transaktionsstruktur der Synchronisation finden Sie unter „[Transaktionen im Synchronisationsprozess](#)“ [[MobiLink - Erste Orientierung](#)].

### Netzwerkaufgaben mit mehreren Threads verwenden

Der MobiLink-Server unterstützt die gleichzeitige Verarbeitung seiner Netzwerkdatenströme durch mehrere Netzwerk-Worker-Threads. Der Einsatz mehrerer Netzwerk-Worker-Threads kann die Performance verbessern, besonders wenn CPU-intensive Netzwerk-Datenstromoptionen wie Verschlüsselung oder Komprimierung zusammen mit entweder großen Synchronisationen oder zahlreichen kleinen Synchronisationen verwendet werden. Jede Anforderung im System kann auf maximal einem Netzwerkdatenstrom-Thread aktiv sein.

Legen Sie Datenstrom-Threads mithilfe der `mlsrv16`-Option `-wn` fest. Siehe „[mlsrv16-Option -wn](#)“ auf Seite 82.

### Optimale Anzahl von Datenbank-Worker-Threads verwenden

Sie können entweder eine feste Anzahl von MobiLink-Datenbank-Worker-Threads wählen oder die Anzahl vom MobiLink-Server automatisch anpassen lassen. Wenn Sie eine bestimmte Anzahl verwenden, indem Sie die Option `-wm` nicht angeben, müssen Sie unterschiedliche Werte für die Option `-w` ausprobieren, um die kleinste Zahl zu ermitteln, die Ihnen den optimalen Durchsatz ermöglicht. Bei automatischer Anpassung legen Sie die maximale Anzahl der Datenbank-Worker-Threads über die Option `-wm` fest und `-w` wird verwendet, um die minimale und anfängliche Anzahl anzugeben. Wenn die Option `-w` nicht angegeben ist, lautet der Standardwert 5. Wenn Sie z.B. mit `-wm 50` arbeiten und nicht die Option `-w` verwenden, passt der MobiLink-Server regelmäßig die Anzahl der aktiven Datenbank-Worker-Threads auf Werte im Bereich von 5 bis 50 an.

Eine große Anzahl von Datenbank-Worker-Threads kann den Durchsatz **eventuell** steigern, da zur gleichen Zeit mehr Synchronisationen auf die konsolidierte Datenbank zugreifen können. Damit geht jedoch ein erhöhtes Potenzial für Konflikte und Blockierungen einher.

Die Anzahl der Datenbank-Worker-Threads niedrig zu halten, reduziert die Konfliktwahrscheinlichkeit in der konsolidierten Datenbank, die Anzahl der Verbindungen zur konsolidierten Datenbank und den für optimales Caching erforderlichen Arbeitsspeicher.

**Siehe auch**

- [Anzahl der Datenbank-Worker-Threads auf Seite 150](#)
- [Automatische Anpassung der Datenbank-Worker-Threads auf Seite 145](#)
- [„mlsrv16-Option -wm“ auf Seite 91](#)
- [„mlsrv16-Option -w “ auf Seite 90](#)
- [„mlsrv16-Option -wu “ auf Seite 92](#)

**Automatische Anpassung der Datenbank-Worker-Threads**

Der MobiLink-Server kann die Anzahl der Datenbank-Worker-Threads automatisch basierend auf der Last anpassen, mit dem Ziel, den Durchsatz zu maximieren. Um diese automatische Anpassung zu aktivieren, legen Sie mithilfe der mlsrv16-Option -w die anfängliche Anzahl von gleichzeitigen Datenbank-Worker-Threads fest und mithilfe der mlsrv 16-Option -wm die maximale Anzahl von gleichzeitigen Datenbank-Worker-Threads. Der MobiLink-Server überwacht die Systemperformance und passt die Anzahl der Datenbank-Worker-Threads innerhalb der Parameter an, die durch die Optionen -w und -wm festgelegt sind, je nach den Anforderungen zu einem bestimmten Zeitpunkt.

Der mit der mlsrv 16-Option -wu festgelegte Wert, der für die maximale Anzahl von Datenbank-Worker-Threads steht, die gleichzeitig Uploads in die konsolidierten Datenbank übernehmen können, wird nicht automatisch angepasst. Wenn die Option -wu nicht angegeben ist, können Uploads von einzelnen oder allen Datenbank-Worker-Threads übernommen werden und die Anzahl der Datenbank-Worker-Threads ist variabel. Wenn der MobiLink-Server die Anzahl der Datenbank-Worker-Threads mit dem Ziel erhöht, den Durchsatz zu steigern, können kurzfristig Konflikte auftreten, bis dies erkannt wird und die Thread-Anzahl schließlich verringert wird.

**Kleinere Upload-Transaktionen verwenden**

Große Uploads können große Transaktionen in der konsolidierten Datenbank verursachen und große Transaktionen führen zu mehr Sperren in einer Transaktion. Dies erhöht die Anzahl der Blockierungen und Konflikte. Dies kann beträchtliche nachteilige Auswirkungen sowohl auf den Synchronisationsdurchsatz als auch auf den Gesamtdurchsatz der konsolidierten Datenbank haben. Kleinere Uploads reduzieren Blockierungen und Konflikte und können den Durchsatz signifikant verbessern.

Bei einem MobiLink-Synchronisationssystem mit entfernten SQL Anywhere-Datenbanken können kleinere Uploads über dbmlsync auf zwei Arten gesendet werden:

- Mit der dbmlsync-Option -tu für transaktionale Uploads. Jede Transaktion wird separat übertragen. Siehe [„dbmlsync-Option -tu“ \[MobiLink - Clientadministration\]](#).
- Mit der erweiterten Option für inkrementelle Uploads dbmlsync Increment (inc). Jedes Inkrement enthält gemischt zusammengefügte Transaktionen. Im Allgemeinen gilt: Je größer das Inkrement, um so mehr Transaktionen werden in einem Upload gemischt zusammengefügt. Siehe [„Erweiterte Option Increment \(inc\)“ \[MobiLink - Clientadministration\]](#).

Auf der Serverseite kann die Performance mit der Option -tx optimiert werden, um eine Reihe von Transaktionen aus dem Client zu einer einzigen konsolidierten serverseitigen Transaktion zusammenzufassen. Diese Option ist insofern praktisch, als Sie nach dem Festlegen der clientseitigen Option -tx einfach optimieren können, ohne die Clients ändern zu müssen. Siehe [„mlsrv16-Option -tx“ auf Seite 85](#).

Testen und optimieren Sie diese clientseitigen und serverseitigen Optionen im Hinblick auf maximalen Durchsatz.

### **Synchronisation unnötiger BLOBs vermeiden**

Es ist ineffizient, einen BLOB in einer Zeile einzuschließen, die häufig synchronisiert wird, solange der BLOB unverändert bleibt. Um dies zu vermeiden, können Sie eine Tabelle erstellen, die BLOBs und eine BLOB-ID enthält, und die ID in dieser Tabelle referenzieren, die synchronisiert werden soll.

### **Maximale Anzahl von Datenbankverbindungen festlegen**

Legen Sie die maximale Anzahl der MobiLink-Datenbankverbindungen so fest, dass sie der Anzahl der Synchronisationsskriptversionen multipliziert mit der Anzahl der Datenbank-Worker-Threads von MobiLink plus Eins entspricht. Damit wird verhindert, dass MobiLink zu häufig Verbindungen trennen und wiederherstellen muss. Sie legen die maximale Anzahl der Verbindungen mit der `mlsrv16`-Option `-cn` fest.

Siehe [MobiLink-Datenbankverbindungen auf Seite 152](#) und „`mlsrv16`-Option `-cn`“ auf Seite 56.

### **Genügend physischer Arbeitsspeicher**

Stellen Sie sicher, dass der Computer, auf dem der MobiLink-Server ausgeführt wird, über ausreichend Arbeitsspeicher verfügt, um den Cache sowie den sonstigen Speicherbedarf aufzunehmen. Prüfen Sie die Möglichkeit, zu einer 64-Bit-Plattform zu wechseln, wenn der Server einen Speicher-Cache von mehr als 1,5 GB benötigt.

Die Anzahl der Synchronisationen, die aktiv verarbeitet werden, wird nicht durch die Anzahl der Datenbank-Worker-Threads begrenzt. Der MobiLink-Server kann für eine große Menge an Synchronisationen gleichzeitig Uploads entpacken und Downloads senden. Wenn ein Server mit dem Paging auf die Festplatte beginnt, wird der Durchsatz deutlich geringer. Deshalb ist es sehr wichtig, dass der MobiLink-Server über ausreichenden Cache-Speicher verfügt, um diese Synchronisationen ohne Paging auf die Festplatte verarbeiten zu können. Um zu erkennen, wann der Cache zu klein ist, suchen Sie im Serverlog nach Warnung 10082 oder dem Alarm "Cache ist voll" vom SQL Anywhere-Monitor für MobiLink.

Der MobiLink-Server stellt automatisch den erforderlichen zusätzlichen oder weniger Speicher-Cache zur Verfügung. Verwenden Sie die Optionen `-cmax`, `-cmin` und `-cinit`, um den Cache-Speicher für den MobiLink-Server zu steuern.

### **Siehe auch**

- „`mlsrv16`-Option `-cinit`“ auf Seite 54
- „`mlsrv16`-Option `-cmax`“ auf Seite 55
- „`mlsrv16`-Option `-cmin`“ auf Seite 56

### **Ausreichende Rechenkapazität**

Stellen Sie MobiLink ausreichend Rechenkapazität zur Verfügung, sodass die Verarbeitung durch den MobiLink-Server keinen Engpass darstellt. In der Regel benötigt der MobiLink-Server bedeutend weniger CPU als die konsolidierte Datenbank. Wenn Sie jedoch die Java- oder .NET-Zeilenbehandlung verwenden, nimmt der Verarbeitungsbedarf des MobiLink-Servers zu. In der Praxis handelt es sich bei Netzwerkeinschränkungen und Datenbankkonflikten eher um Engpässe.

## Skriptausführung optimieren

Die Performance Ihrer Skripten in der konsolidierten Datenbank ist ein wichtiger Faktor. Es kann hilfreich sein, Indizes zu den Tabellen zu erstellen, sodass Upload- und Download-Cursor-Skripten die erforderlichen Zeilen effizient finden können. Zu viele Indizes können Uploads jedoch verlangsamen.

Wenn Sie den **Assistenten zum Erstellen eines Synchronisationsmodells** in Sybase Central benutzen, um Ihre MobiLink-Anwendungen zu erstellen, wird für jeden Download-Cursor beim Deployment des Modells automatisch ein Index definiert.

## Minimale Ausführlichkeit für die Protokollierung

Benutzen Sie die geringste, für Ihre Geschäftsanforderungen noch angemessene Ausführlichkeit für die Logausgabe. Standardmäßig ist die Logausgabe deaktiviert und MobiLink schreibt kein Log auf den Plattenspeicher. Sie können die Logausgabe mit der Option -v steuern und mit den Optionen -o oder -ot die Ausgabe in eine Datei aktivieren.

Als Alternative zu ausführlichen Logdateien können Sie Ihre Synchronisationen mit dem MobiLink-Profiler überwachen. Der MobiLink-Profiler muss sich nicht auf demselben Computer befinden wie der MobiLink-Server und die Auswirkungen einer Monitor-Verbindung auf die Performance des MobiLink-Servers können vernachlässigt werden. Siehe „[MobiLink-Profiler](#)“ auf Seite 192.

## Beschränkungen des Betriebssystems einplanen

Betriebssysteme schränken die Anzahl der gleichzeitigen Verbindungen ein, die ein Server über TCP/IP unterstützen kann. Wenn diese Grenze erreicht ist, was der Fall sein kann, wenn über 1.000 Clients gleichzeitig eine Synchronisation ausführen möchten, kann sich das Betriebssystem unter Umständen unerwartet verhalten, beispielsweise unerwartet Verbindungen beenden oder weitere Clients zurückweisen, die eine Verbindung herzustellen versuchen. Um dieses Verhalten zu verhindern, haben Sie folgende Möglichkeiten: Sie konfigurieren das Betriebssystem mithilfe der Option -nc so, dass es einen höheren Grenzwert für TCP/IP-Verbindungen hat, oder Sie geben mit der Option -sm die maximale Anzahl entfernter Verbindungen, die unter dem Grenzwert des Betriebssystems liegt, an.

Wenn ein Client versucht, mit einem MobiLink-Server zu synchronisieren, der seine mit der Option -sm festgelegte maximale Anzahl von gleichzeitigen Synchronisationen akzeptiert hat, empfängt er den Fehlercode -1305 (SQLE\_MOBILINK\_COMMUNICATIONS\_ERROR). Die Clientanwendung sollte diesen Fehler bearbeiten und nach ein paar Minuten versuchen, die Verbindung erneut herzustellen.

Siehe:

- „mlsrv16-Option -sm“ auf Seite 81
- „mlsrv16-Option -nc“ auf Seite 65
- „Verbindungsfehler“ [[Fehlermeldungen](#)]

## Java- oder .NET im Vergleich zur SQL-Synchronisationslogik

Bei der Verwendung der Java- oder .NET-Synchronisationslogik bzw. der SQL-Synchronisationslogik wurden keine bedeutenden Durchsatzunterschiede festgestellt. Die Java- und .NET-Synchronisationslogik haben jedoch etwas Overhead pro Synchronisation und benötigen mehr Speicher.

Außerdem wird die SQL-Synchronisationslogik auf dem Recher ausgeführt, auf dem die konsolidierte Datenbank läuft, während die Java- oder .NET-Synchronisationslogik auf dem Recher ausgeführt wird,

auf dem der MobiLink-Server läuft. Die Java- und .NET-Synchronisationslogik kann zu bevorzugen sein, wenn die konsolidierte Datenbank stark ausgelastet ist.

Eine Synchronisation, die die direkte Zeilenbehandlung verwendet, belastet den MobiLink-Server mit einem großen Verarbeitungsaufwand. Daher benötigen Sie je nach implementierter direkter Zeilenbehandlung unter Umständen mehr Arbeitsspeicher (RAM), mehr Festplattenspeicher und eine größere CPU.

### **Priorität der Synchronisation**

Wenn Sie bestimmte Tabellen haben, die häufiger als andere synchronisiert werden müssen, erstellen Sie eine eigene Publikation und Subskription für sie. Wenn Sie Synchronisationsmodelle in Sybase Central verwenden, können Sie dies durchführen, indem Sie mehr als ein Modell erstellen. Sie können diese Prioritätspublikation häufiger als andere Publikationen synchronisieren und andere Publikationen außerhalb der Zeiten der Spitzenbelastung synchronisieren.

### **Download auf die erforderlichen Zeilen begrenzen**

Achten Sie darauf, nur die erforderlichen Zeilen per Download abzurufen, indem Sie z.B. die Zeitstempel-Synchronisation statt eines Snapshots verwenden. Der Download nicht benötigter Zeilen verschwendet Ressourcen und wirkt sich nachteilig auf die Synchronisations-Performance aus.

### **Synchronisieren Sie nur, wenn es erforderlich ist**

Zu häufige Synchronisationen können zu unnötigen Belastungen des MobiLink-Synchronisationssystems führen. Überprüfen Sie sorgfältig, wie oft Sie synchronisieren müssen. Testen Sie gründlich, um zu gewährleisten, dass die Performance-Erwartungen in der Produktionsumgebung erfüllt werden können.

### **Anzahl der Zeilen bei großen Uploads schätzen**

Für SQL Anywhere-Clients können Sie die Geschwindigkeit bei einem Upload einer großen Zahl an Zeilen erheblich steigern, indem Sie dbmlsync eine Schätzung der einzulesenden Zeilen übergeben. Hierzu verwenden Sie die dbmlsync-Option -urc.

Siehe „dbmlsync-Option -urc“ [[MobiLink - Clientadministration](#)].

### **Verwenden Sie Hintergrundsynchroisation**

Je mehr Synchronisation im Hintergrund erfolgt, umso weniger dringend ist es vom Gesichtspunkt des entfernten Benutzers aus, dass sie schnellstmöglich geschieht. Legen Sie Ihre entfernte Anwendung so an, dass sie im Hintergrund synchronisiert, damit entfernte Benutzer auch während der Synchronisation weiterarbeiten können.

## **Schlüsselfaktoren für die MobiLink-Performance**

Die Gesamt-Performance eines beliebigen Systems, einschließlich Durchsatz für die MobiLink-Synchronisation, wird oft von einem Engpass an einem Punkt im System eingeschränkt. Bei der MobiLink-Synchronisation können Engpässe an folgenden Stellen den Durchsatz der Synchronisation beeinträchtigen:

- **Performance der konsolidierten Datenbank** Von besonderer Bedeutung für MobiLink ist die Geschwindigkeit, mit der die konsolidierte Datenbank MobiLink-Skripten ausführen kann. Da mehrere Datenbank-Worker-Threads gleichzeitig Skripten ausführen können, wird der beste Durchsatz erzielt, wenn in den Synchronisationsskripten Datenbankkonflikte vermieden werden.
- **Anzahl der MobiLink-Datenbank-Worker-Threads** Eine geringere Anzahl von Threads hat eine geringere Anzahl von Datenbankverbindungen, weniger Konflikte in der konsolidierten Datenbank und weniger Betriebssystem-Overhead zur Folge. Wenn die Anzahl jedoch zu klein ist, kann es geschehen, dass Clients auf freie Datenbank-Worker-Threads warten müssen oder weniger Verbindungen mit der konsolidierten Datenbank haben, als effizient ausgeführt werden könnten.
- **Bandbreite der Kommunikation zwischen Client und MobiLink** Bei langsamen Verbindungen, etwa über DFÜ- oder drahtlose Weitbereichsnetzwerke, kann das Netzwerk dazu führen, dass Clients und MobiLink-Server warten müssen, bis die Daten übertragen werden.
- **Verarbeitungsgeschwindigkeit beim Client** Geringe Verarbeitungsgeschwindigkeit beim Client kann eher bei Downloads als bei Uploads zu einem Engpass führen. Downloads benötigen mehr Client-Verarbeitung, da Zeilen und Indizes geschrieben werden müssen.
- **Die Geschwindigkeit des Computers, auf dem MobiLink läuft** Wenn die Rechenkapazität des Computers, auf dem MobiLink läuft, gering ist oder nicht genügend Arbeitsspeicher für die Datenbank-Worker-Threads von MobiLink und die Puffer vorhanden ist, kann die Ausführungsgeschwindigkeit von MobiLink einen Synchronisationsengpass darstellen. Die Performance des MobiLink-Servers hängt nur geringfügig von der Plattengeschwindigkeit ab, sofern die Puffer und Datenbank-Worker-Threads Platz im physischen Speicher haben.
- **Bandbreite für die Kommunikation zwischen MobiLink und konsolidierter Datenbank** Hier ist die Wahrscheinlichkeit eines Engpasses nicht sehr groß, wenn MobiLink und die konsolidierte Datenbank auf demselben System laufen oder wenn sie auf verschiedenen Systemen laufen und über ein Hochgeschwindigkeitsnetz verbunden sind.

## MobiLink-Optimierung der Performance

Der Schlüssel zum optimalen Durchsatz während der MobiLink-Synchronisation besteht darin, mehrere Synchronisationen gleichzeitig und effizient auszuführen. Um mehrere gleichzeitige Synchronisationen zu ermöglichen, verwendet MobiLink Pools von Datenbank-Worker-Threads für unterschiedliche Aufgaben. Ein Pool ist dem Lesen von Upload-Daten aus dem Netzwerk und dem Entpacken derselben gewidmet. Ein anderer Pool von Threads, die als **Datenbank-Worker-Threads** bezeichnet werden, wendet die Upload-Daten auf die konsolidierte Datenbank an und stellt Daten bereit, die per Download aus der konsolidierten Datenbank abgerufen werden sollen. Ein weiterer Pool von Datenbank-Worker-Threads dient zum Verpacken und Senden der Download-Daten an entfernte Datenbanken. Jeder einzelne Datenbank-Worker-Thread verwendet eine eigene Verbindung mit der konsolidierten Datenbank zum Übernehmen und Abrufen von Änderungen mithilfe der Synchronisationsskripten.

### Konfliktsituationen

Der wichtigste Faktor ist, in den Synchronisationsskripten Datenbankkonflikte zu vermeiden. Genau wie bei allen anderen Multi-Client-Einsätzen einer Datenbank müssen Sie die Datenbankkonflikte minimieren, wenn Clients gleichzeitig auf eine Datenbank zugreifen. Datenbankzeilen, die bei jeder

einzelnen Synchronisation geändert werden müssen, können die Anzahl der Konflikte erhöhen. Wenn Ihre Skripten z.B. einen Zähler in einer Zeile aktivieren, kann dies zu einem Engpass führen.

Synchronisationsanforderungen werden akzeptiert (bis zu der Grenze, die mit dem Option -sm festgelegt wurde) und die hochgeladenen Daten werden gelesen und entpackt, so dass sie für den Datenbank-Worker-Thread bereit sind. Falls es mehr Synchronisationen als Datenbank-Worker-Threads gibt, werden die überschüssigen Synchronisationen in die Warteschlange gestellt, wo sie auf einen freien Datenbank-Worker-Thread warten.

Sie können die Anzahl der Datenbank-Worker-Threads und Verbindungen steuern, aber MobiLink wird immer dafür Sorge tragen, dass für jeden Datenbank-Worker-Thread zumindest eine Verbindung vorhanden ist. Wenn mehr Verbindungen als Datenbank-Worker-Threads vorhanden sind, sind die überzähligen Verbindungen inaktiv. Überzählige Verbindungen sind nützlich beim Einsatz mehrerer Skriptversionen, wie unten noch beschrieben wird.

### Anzahl der Datenbank-Worker-Threads

Abgesehen von Konflikten in Synchronisationsskripten ist die Anzahl der Datenbank-Worker-Threads die wichtigste Einflussgröße für den Synchronisationsdurchsatz. Die Anzahl der Datenbank-Worker-Threads steuert, wie viele Synchronisationen gleichzeitig in der konsolidierten Datenbank ablaufen können.

Um die optimale Anzahl von Datenbank-Worker-Threads zu bestimmen, müssen Tests durchgeführt werden.

Durch die Erhöhung der Anzahl der Datenbank-Worker-Threads können mehr überlappende Synchronisationen auf die konsolidierte Datenbank zugreifen. Dadurch wird der Durchsatz **eventuell** gesteigert. Es nehmen jedoch auch die Ressourcen- und Datenbankkonflikte zwischen den überlappenden Synchronisationen zu, wodurch einzelne Synchronisationen mehr Zeit in Anspruch nehmen können. Mit steigender Anzahl der Datenbank-Worker-Threads geht der Vorteil der gleichzeitigen Synchronisationen durch die Kosten der längeren Einzel-Synchronisationen verloren, und durch das Hinzufügen weiterer Datenbank-Worker-Threads sinkt der Durchsatz. Durch Experimentieren können Sie die optimale Anzahl der Datenbank-Worker-Threads für Ihre Situation ermitteln. Folgende Hinweise mögen dabei hilfreich sein:

Performance-Tests zeigen, dass der beste Upload-Durchsatz in der Regel mit drei bis zehn Datenbank-Worker-Threads erreicht wird. Variationen hängen ab von Faktoren wie Art der konsolidierten Datenbank, Datenvolumen, Datenbankschema, Komplexität der Synchronisationsskripten und eingesetzte Hardware. Zu einem Engpass kommt es gewöhnlich aufgrund einer Konfliktsituation zwischen Datenbank-Worker-Threads, die die SQL-Anweisungen des Uploadskripts in der konsolidierten Datenbank gleichzeitig ausführen.

Legen Sie die Anzahl der Datenbank-Worker-Threads mithilfe der mlsrv16-Option -w fest. Sie können auch mit die mlsrv16-Option -wm verwenden, um die Anzahl der Datenbank-Worker-Threads vom MobiLink-Server automatisch in Bezug auf den Durchsatz optimieren zu lassen, vorausgesetzt, die Anzahl liegt zwischen den aktuellen Einstellungen der Optionen -w und -wm. Beachten Sie jedoch, dass die Option -wm zwar praktisch ist, aber möglicherweise nicht in allen Fällen funktioniert. Um die optimale Anzahl von Datenbank-Worker-Threads zu bestimmen, müssen wie immer Tests durchgeführt werden.

Wird keine Downloadbestätigung verwendet (Standardeinstellung), ist die Bandbreite zwischen Client und MobiLink weniger maßgeblich, da ein Datenbank-Worker-Thread zur Verarbeitung anderer Synchronisationen frei steht, während die anderen Threads den Download senden. Die Anzahl der Datenbank-Worker-Threads ist daher weniger entscheidend.

Es können weitaus mehr Downloads gleichzeitig gesendet werden als Datenbank-Worker-Threads. Für eine optimale Download-Performance ist es wichtig, dass dem MobiLink-Server ausreichend Arbeitsspeicher (RAM) zur Verfügung steht, um diese Downloads zu puffern. Andernfalls wird der Download seitenweise auf der Festplatte gespeichert, und die Download-Performance kann beeinträchtigt werden. Geben Sie die Größe des Cache-Speichers für den MobiLink-Server mit der Option `-cmax` an.

Siehe „[mlsrv16-Option -cmax](#)“ auf Seite 55.

Falls der MobiLink-Server mit dem Paging auf die Festplatte beginnt (möglicherweise, wenn zu viele Downloads gleichzeitig verarbeitet werden), sollten Sie die Option `-sm` benutzen, um entweder die Anzahl der Datenbank-Worker-Threads zu verringern oder die Gesamtanzahl der Synchronisationen, die aktiv verarbeitet werden, zu begrenzen.

Siehe „[mlsrv16-Option -sm](#)“ auf Seite 81.

Wenn Sie die Downloadbestätigung deaktiviert lassen (Standardeinstellung), kann dies die optimale Anzahl der Datenbank-Worker-Threads reduzieren, da die Datenbank-Worker-Threads keine Downloadbestätigungstransaktionen verarbeiten müssen.

Siehe „[Erweiterte Option SendDownloadAck \(sa\)](#)“ [*MobiLink - Clientadministration*].

Die Blockierung bei Downloadbestätigungen wurde abgeschafft, sodass alle Downloadbestätigungen im nicht blockierenden Modus verarbeitet werden. Dies führt zu einer besseren Performance. Bei der nicht blockierenden Bestätigung verwendet der Server den Datenbank-Worker-Thread erneut, während die entfernte Datenbank den Download übernimmt, sodass die Anzahl der Datenbank-Worker-Threads möglicherweise nicht erhöht werden muss, was zu einer verbesserten Performance führt.

Um den besten Download- und Upload-Durchsatz zu erzielen, bietet MobiLink zwei Möglichkeiten: Sie können eine Gesamtzahl von Datenbank-Worker-Threads angeben, damit die Downloads optimiert werden. Sie können auch die Anzahl begrenzen, die gleichzeitig Uploads übernehmen kann, um den Upload-Durchsatz zu optimieren.

Mit der Option `-w` wird die Anzahl der Datenbank-Worker-Threads gesteuert. Der Standardwert ist fünf.

Mit der Option `-wu` wird die Anzahl der Datenbank-Worker-Threads begrenzt, die gleichzeitig Uploads in die konsolidierte Datenbank übernehmen können. Standardmäßig können alle Datenbank-Worker-Threads Uploads gleichzeitig übernehmen, doch dies kann zu schweren Konfliktsituationen in der konsolidierten Datenbank führen. Mit der Option `-wu` können Sie diese Konfliktsituation eingrenzen, während Sie weiter eine große Anzahl von Datenbank-Worker-Threads haben, um das Abrufen von Download-Daten zu optimieren. Der Parameter `-wu` hat nur dann einen Einfluss, wenn der Wert unter der Gesamtzahl der Datenbank-Worker-Threads liegt.

Siehe „[mlsrv16-Option -w](#)“ auf Seite 90 und „[mlsrv16-Option -wu](#)“ auf Seite 92.

## MobiLink-Datenbankverbindungen

MobiLink erstellt für jeden Datenbank-Worker-Thread eine Datenbankverbindung. Mit der Option `-cn` können Sie angeben, dass MobiLink einen größeren Pool von Datenbankverbindungen herstellen soll. Die überzähligen Verbindungen bleiben jedoch inaktiv, es sei denn, MobiLink muss eine Verbindung trennen oder eine andere Skriptversion verwenden.

Es gibt zwei Fälle, in denen MobiLink eine Datenbankverbindung trennt und eine neue herstellt. Der erste Fall tritt ein, wenn ein Fehler auftritt. Der zweite Fall, wenn der Client eine Synchronisationsskriptversion anfordert und keine der verfügbaren Verbindungen diese Synchronisationsskriptversion bisher verwendet hat.

### Hinweis

Jeder Datenbankverbindung ist eine Skriptversion zugeordnet. Die Skriptversion kann nur geändert werden, wenn die Verbindung getrennt und wiederhergestellt wird.

Wenn Sie normalerweise mehr als eine Skriptversion benutzen, können Sie die Notwendigkeit, dass MobiLink Verbindungen trennen und wiederherstellen muss, reduzieren, indem Sie die Anzahl der Verbindungen erhöhen. Sie können die Notwendigkeit vollständig beseitigen, wenn die Verbindungsanzahl für Synchronisationen gleich der Anzahl der Datenbank-Worker-Threads multipliziert mit der Anzahl der Skriptversionen ist.

Ein Beispiel zur Optimierung von MobiLink für zwei Skriptversionen finden Sie in der folgenden Befehlszeile:

```
mllsrv16 -c "DSN=SQL Anywhere 16 Demo" -w 5 -cn 10
```

Da die maximal für Synchronisationen nutzbare Anzahl der Datenbankverbindungen gleich der Anzahl der Skriptversionen multipliziert mit der Anzahl der Datenbank-Worker-Threads ist, können Sie `-cn` auf 10 setzen, um zu gewährleisten, dass die Datenbankverbindungen nicht übermäßig getrennt und geöffnet werden.

Siehe „[mllsrv16-Option -cn](#)“ auf Seite 56.

## MobiLink-Performanceüberwachen

Es gibt eine Reihe von Tools, mit denen Sie die Performance Ihrer Synchronisationen überwachen können.

Der MobiLink-Profiler ist ein grafisches Tool zum Überwachen von Synchronisationen. Mit ihm können Sie feststellen, wie lange die einzelnen Schritte der Synchronisation dauern.

Siehe „[MobiLink-Profiler](#)“ auf Seite 192.

Außerdem sind eine Reihe von MobiLink-Skripten für die Überwachung der Synchronisationen verfügbar. Diese Skripten gestatten es Ihnen, Performance-Statistiken in Ihre Unternehmenslogik einzubeziehen. Sie können z.B. die Performance-Daten für eine spätere Analyse speichern oder einen DBA warnen, wenn eine Synchronisation zu lange dauert. Sie müssen diese Skripten mit derselben

Umsicht schreiben wie Ihre anderen Skripten und dabei Konflikte und Blockierungen so weit wie möglich vermeiden. Weitere Hinweise finden Sie unter .

- „download\_statistics (Verbindungsereignis)“ auf Seite 333
- „download\_statistics (Tabellenereignis)“ auf Seite 337
- „synchronization\_statistics (Verbindungsereignis)“ auf Seite 421
- „synchronization\_statistics (Tabellenereignis)“ auf Seite 425
- „time\_statistics (Verbindungsereignis)“ auf Seite 428
- „time\_statistics (Tabellenereignis)“ auf Seite 432
- „upload\_statistics (Verbindungsereignis)“ auf Seite 448
- „upload\_statistics (Tabellenereignis)“ auf Seite 452

## Zentrale Administration von entfernten Datenbanken

SQL Anywhere lässt zu, dass ein zentraler Administrator die an einer MobiLink-Synchronisation beteiligten entfernten Datenbanken verwaltet.

Im Rahmen der zentralen Administration von entfernten Datenbanken können Sie folgende Schritte ausführen:

- Zentrale Steuerung des Zeitpunkts, zu dem eine entfernte Datenbank mit MobiLink synchronisiert wird.
- Durchführung von Schemaänderungen in entfernten Datenbanken.
- Diagnose von Problemen mit bestimmten entfernten Datenbanken oder mit dem Synchronisationssystem allgemein.
- Upload von Logdateien.

Die zentrale Administration von entfernten Datenbanken ersetzt die SQL Passthrough-Funktion, die ursprünglich in Version 11.0.0 eingeführt wurde und es gestattete, Skripten mit SQL-Anweisungen von einer konsolidierten Datenbank auf einen SQL Anywhere- oder UltraLite-Client herunterzuladen und diese SQL-Anweisungen auf dem Client zum geeigneten Zeitpunkt auszuführen.

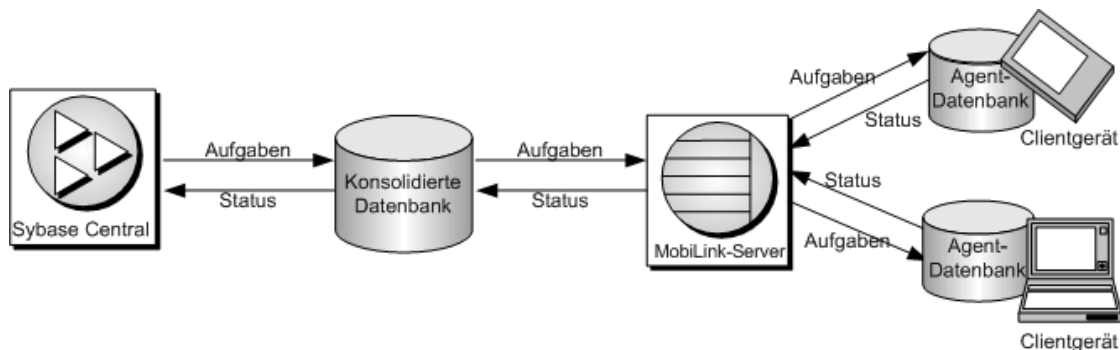
Die zentrale Administration erfolgt mithilfe von entfernten Aufgaben. Eine entfernte Aufgabe ist eine geordnete Sammlung von Befehlen (ähnlich einer Batchdatei), die Sie mit dem MobiLink 16-Plug-In für Sybase Central erstellen. Eine entfernte Aufgabe kann Bedingungen enthalten, die bestimmen, wann sie ausgeführt wird. Eine entfernte Aufgabe kann so konfiguriert werden, dass sie nur einmal ausgeführt wird, dass sie bei Bedarf ausgeführt wird oder dass sie regelmäßig wiederholt wird. Entfernte Aufgaben werden in der konsolidierten Datenbank bereitgestellt und können einem oder mehreren MobiLink-Agenten zugewiesen werden.

Der MobiLink-Agent ist eine Anwendung, die auf einem entfernten Gerät läuft. Jede Gruppe von Datenbanken, für die ein bestimmter MobiLink-Server als zentraler Administrator fungiert, muss eine Instanz des MobiLink-Agenten ausführen. Der Agent ist dafür zuständig, die ihm zugewiesenen entfernten Aufgaben zu den entsprechenden Zeiten auszuführen. Der MobiLink-Agent verwendet eine UltraLite-Datenbank, die als "Agenten-Datenbank" bezeichnet wird, zum Speichern der ihm

zugewiesenen Aufgaben, der Ergebnisse von ausgeführten Aufgaben und anderer Konfigurationsinformationen.

In regelmäßigen Abständen synchronisiert der MobiLink-Agent seine Agenten-Datenbank mithilfe einer normalen MobiLink-Synchronisation. Während dieser Synchronisationen empfängt der Agent alle neuen Aufgaben, die ihm zugeordnet wurden, und lädt optional Ergebnisse von Aufgaben hoch, die er ausgeführt hat, sodass diese vom Systemadministrator unter Verwendung des MobiLink 16-Plug-Ins für Sybase Central überprüft werden können.

Die folgende Abbildung zeigt den Informationsfluss zwischen Sybase Central, der konsolidierten Datenbank und der Clientgeräte, wenn die zentrale Administration von entfernten Datenbanken angewendet wird.



## Konzepte zur zentralen Administration

Die folgenden Konzepte sind wichtig für das Verständnis bei der Arbeit mit der zentralen Administration von entfernten Datenbanken.

### MobiLink-Projekt

Ein MobiLink-Projekt wird von einem Administrator unter Verwendung des MobiLink 16-Plug-Ins für Sybase Central erstellt. Sie müssen zuerst ein MobiLink-Projekt definieren, bevor Sie mit entfernten Aufgaben arbeiten können.

Ein MobiLink-Projekt ist eine Sammlung, die Folgendes enthält:

- Keine oder mehrere entfernte Aufgaben.
- Mindestens eine Verbindung zu einer konsolidierten Datenbank.
- Keine oder mehrere Synchronisationsmodelle.

Ein Beispiel für ein MobiLink-Projekt finden Sie unter `%%\MobiLink\CustDB\project.mlp`.

## MobiLink-Agent

Ein MobiLink-Agent ist eine Anwendung, die auf dem Clientgerät läuft. Der Agent dient dazu, Aufgaben vom MobiLink-Server zu empfangen und auszuführen, sowie den Status dieser Aufgaben an den MobiLink-Server zurückzumelden.

Der MobiLink-Agent kann mehrere entfernte Datenbanken auf dem Clientgerät verwalten. Wenn entfernte Datenbanken auf dem Clientgerät mit verschiedenen konsolidierten Datenbanken synchronisiert werden müssen, erfordert das Gerät für jede konsolidierte Datenbank, mit der die Anwendung synchronisiert werden muss, einen eigenen MobiLink-Agenten.

## MobiLink-Agenten-ID

Die MobiLink-Agenten-ID ist eine Zeichenfolge, die einen auf einem Clientgerät laufenden Agenten für den MobiLink-Server identifiziert. Der Administrator, der mit dem MobiLink-Projekt arbeitet, kann sie anzeigen. Da jeder Agent auf einem einzigen Clientgerät läuft, identifiziert diese ID für den Administrator auch das Gerät.

Jeder MobiLink-Agent muss eine eindeutige ID haben. Die Agenten-ID kann entweder beim Start mit dem `mlagent`-Befehl angegeben werden oder es wird ein Standardwert in der Form `Agent_computername_UUID` zugewiesen, wobei *computername* für den Hostnamen des Computers steht, auf dem der Agent ausgeführt wird, und *UUID* für einen universell eindeutigen Bezeichner (Universally Unique Identifier = UUID).

Es wird dringend empfohlen, nicht die Groß- und Kleinschreibung für die Unterscheidung zwischen Agenten-IDs zu benutzen. Sie sollten beispielsweise nicht die Agenten-IDs **Agent\_XYZ** und **agent\_xyz** als verschiedene Agenten-IDs erstellen. Wenn die konsolidierte Datenbank die Groß- und Kleinschreibung nicht berücksichtigt, muss diese Empfehlung beachtet werden. Wenn die konsolidierte Datenbank die Groß- und Kleinschreibung berücksichtigt, wird diese Empfehlung nicht erzwungen.

## Entfernte Aufgabe

Eine Arbeitseinheit wird als "entfernte Aufgabe" bezeichnet. Eine entfernte Aufgabe ist eine Sammlung von Befehlen. Der MobiLink-Agent empfängt zu erledigende Arbeiten in Form von entfernten Aufgaben und meldet den Status der durchgeführten Arbeiten an den Administrator zurück.

## Befehl

Ein Befehl ist eine Anweisung in einer Aufgabe, die eine bestimmte Aktion ausführt. Eine Aufgabe kann mehrere Befehle enthalten und für die Befehle gilt eine festgelegte Reihenfolge. Ein Befehl enthält eine auszuführende Aktion, Eingabeparameter und Anweisungen für den Fall, dass der Befehl fehlschlägt.

## Bereitgestellte entfernte Aufgabe

Eine bereitgestellte entfernte Aufgabe ist eine Aufgabe, die in die konsolidierte Datenbank kopiert wurde. Nur bereitgestellte Aufgaben können einem Agenten zur Ausführung zugewiesen werden.

## Statusinformationen

Statusinformationen sind Informationen über entfernte Aufgaben, z.B. ob die Aufgaben erfolgreich abgeschlossen wurden. Diese Informationen werden auf dem Client in der Agenten-Datenbank gespeichert, wenn Aufgaben ausgeführt werden, und zu unterschiedlichen Zeiten an den Server gesendet, damit der Administrator den Status der entfernten Aufgaben im System anzeigen kann.

## Agenten-Datenbank

Eine Agenten-Datenbank ist eine UltraLite-Datenbank auf dem entfernten Gerät, die vom MobiLink-Agenten verwendet wird, um Informationen zu Aufgaben und Konfiguration zu speichern.

Der Standardspeicherort der Agentendatenbank ist `%ALLUSERSPROFILE%\Application Data\SQL Anywhere 16\MobiLink Agent` unter Windows und `Mein Gerät\Anwendungsdaten\SQLAny16\MLAgent` unter Windows Mobile.

Der Dateiname der Agentendatenbank besteht aus dem mit der Option `-n` für `mlagent.exe` angegebenen Wert und der Erweiterung `.udb`. Wenn die Option `-n` nicht angegeben ist, lautet der Standardname `mlagent.udb`.

## Entfernte Datenbank

Eine entfernte Datenbank ist eine UltraLite- oder SQL Anywhere-Datenbank auf einem entfernten Gerät, die Ihre Anwendungsdaten enthält, an der MobiLink-Synchronisation beteiligt ist und von einem MobiLink-Agenten verwaltet wird. Jede entfernte Datenbank enthält einen Schemanamen, der ihr Schema identifiziert.

## Name des entfernten Schemas

Der Name eines entfernten Schemas identifiziert eine Gruppe von Datenbanken, die dasselbe Schema aufweisen. In der Regel sind alle Datenbanken, deren entferntes Schema denselben Namen aufweist, Datenbanken für dieselbe Version einer Anwendung. Ein Schema enthält beispielsweise: Tabellendefinitionen, gespeicherte Prozeduren, Trigger, Publikationen und Synchronisationsprofile. Ein Schema enthält nicht die Elemente, die sich normalerweise von einer Instanz einer Datenbank zur anderen unterscheiden, z.B. Synchronisations- und Datenbankbenutzer.

Eine entfernte Datenbank kann nicht entfernt verwaltet werden, wenn sie keinen Namen für ein entferntes Schema hat, sodass zumindest ein Name für ein entferntes Schema definiert werden muss bevor in Sybase Central ein Agent erstellt werden kann.

Wenn Sie einem Projekt entweder mit dem **Assistenten zum Erstellen eines Projekts** oder dem **Assistenten zum Hinzufügen einer konsolidierten Datenbank** eine konsolidierte Datenbank hinzufügen, prüft der Assistent automatisch, ob Namen für entfernte Schemata in der konsolidierten Datenbank definiert sind, die noch nicht im Projekt enthalten sind. Wenn dies der Fall ist, werden Sie aufgefordert, diese zu importieren.

## Serverinitiierte entfernte Aufgabe (SIRT)

Eine serverinitiierte entfernte Aufgabe ist eine entfernte Aufgabe, die ausgeführt wird, wenn der Agent eine entsprechende Benachrichtigung vom Server empfängt. Eine Aufgabe kann einen Zeitplan aufweisen, aber trotzdem vom Server initiiert werden.

## Setup-Überblick

Die folgenden Schritte zeigen die Vorgehensweise zum Einrichten der zentralen Administration von entfernten Datenbanken auf dem Server und auf dem Client.

### Einrichten der zentralen Administration auf dem Server

1. Erstellen Sie in Sybase Central ein MobiLink-Projekt. Siehe „[Erstellen eines MobiLink-Projekts](#)“ [[MobiLink - Erste Orientierung](#)].
2. Mit Sybase Central definieren Sie einen oder mehrere Namen für entfernte Schemata zum Identifizieren der entfernten Datenbank(en) im System.
3. Verwenden Sie Sybase Central, um alle Agenten, die entfernte Datenbanken verwalten, in Ihrem System bekannt zu machen.
4. Verwenden Sie Sybase Central, um Aufgaben zu erstellen und Agenten zuzuweisen. Siehe „[Entfernte Aufgaben](#)“ auf Seite 169.

#### **Einrichten der zentralen Administration auf dem Client**

1. Konfigurieren Sie den MobiLink-Agenten auf jedem Gerät, auf dem die Anwendung ausgeführt wird. Dies gibt ihm im System eine Identität und liefert die MobiLink-Verbindungsinformationen.
2. Starten Sie den MobiLink-Agenten auf dem Gerät, damit er in der Lage ist, Aufgaben vom Server zu empfangen und auszuführen. Siehe „[MobiLink-Agent auf dem Clientgerät](#)“ auf Seite 158.

## **MobiLink-Agenten**

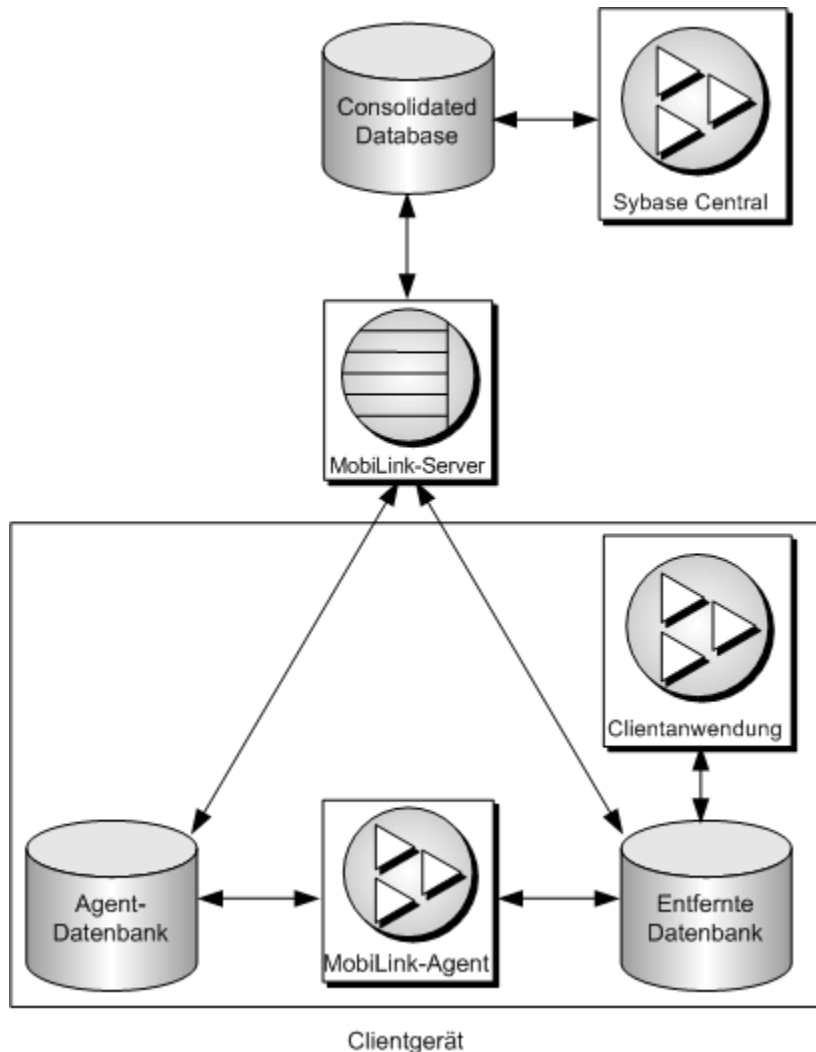
Der Agent verwaltet die Ausführung von entfernten Aufgaben auf einem Gerät. Er speichert die auszuführenden Aufgaben und die Ergebnisse von ausgeführten Aufgaben in der Agenten-Datenbank. Der Agent synchronisiert die Agenten-Datenbank mithilfe der normalen MobiLink-Synchronisation. Während der Synchronisation empfängt der Agent neue auszuführende Aufgaben und lädt Informationen zu ausgeführten Aufgaben hoch.

Der Agent synchronisiert die Agenten-Datenbank, wenn eines der folgenden Ereignisse eintritt:

- Der Agent wird gestartet.
- Der Agent erhält eine Benachrichtigung vom Server zur Synchronisation der Agenten-Datenbank. Der Agent wartet auf Benachrichtigungen vom Server und synchronisiert die Agenten-Datenbank, wenn er benachrichtigt wird.
- Ein vom Benutzer festgelegter Zeitraum ist seit der letzten Synchronisation verstrichen.
- Eine Aufgabe wird abgeschlossen und ist so konfiguriert, dass sie ihren Status unmittelbar danach sendet.

Der Agent verwendet mehrere Threads und führt möglicherweise mehrere Aufgaben parallel aus. Eine Aufgabe wird erst gelöscht, wenn sie in einen "endgültigen" Status eintritt. Endgültige Status sind "Erfolgreich", "Fehlgeschlagen", "Abgelaufen" und "Abgebrochen". Eine Aufgabe tritt nur in einen endgültigen Status ein, wenn sie exklusiv oder sofort ausgeführt wird und dann erfolgreich ist oder fehlschlägt, wenn es sich um eine geplante Aufgabe handelt und sie abgelaufen ist oder wenn sie auf dem Server abgebrochen wird. Bei Bedarf ausgeführte Aufgaben treten nicht in einen endgültigen Status ein, es sei denn, sie werden abgebrochen. Solche Aufgaben befinden sich im Agenten und werden durch vom Server initiierte Anforderungen ausgeführt (SIRT).

Das folgende Diagramm zeigt den Datenfluss zum und vom Agenten.



## MobiLink-Agent auf dem Clientgerät

Der Agent kann in zwei Modi ausgeführt werden: Konfigurationsmodus und Normalmodus. Im Konfigurationsmodus werden in der Befehlszeile angegebene Optionen in der Agenten-Datenbank gespeichert, damit sie bei der nächsten Ausführung im Normalmodus verwendet werden können. Wenn die angegebenen Optionen gespeichert wurden, wird der Agent beendet.

Bei der Ausführung im Normalmodus liest der Agent die in der Agenten-Datenbank gespeicherten Konfigurationsoptionen und läuft weiter. Während der Agent läuft, führt er ggf. entfernte Aufgaben aus, die er empfangen hat, und synchronisiert die Agenten-Datenbank zu unterschiedlichen Zeitpunkten, um

neue entfernte Aufgaben zu empfangen und Ergebnisse von ausgeführten entfernten Aufgaben hochzuladen.

Bei der Ausführung im Normalmodus versucht der Agent immer, eine Synchronisation beim Start durchzuführen. Dies kann sinnvoll sein, wenn Sie den Agenten zwingen wollen, aktuelle Informationen von MobiLink abzurufen.

### Siehe auch

- „mlagent-Befehl“ auf Seite 159

## mlagent-Befehl

Führt den MobiLink-Agenten auf dem Clientgerät aus, entweder im Konfigurationsmodus oder im Normalmodus.

### Syntax

**mlagent** [ *options* ]

Um den Agenten im Konfigurationmodus auszuführen, geben Sie in der mlagent-Befehlszeile -c oder -cr ein.

| Option | Beschreibung                                                                                                                                                                                                                                                                                                                                                 |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| @data  | Liest Optionen aus der angegebenen Umgebungsvariablen oder Konfigurationsdatei ein. Wenn beide mit demselben Namen vorhanden sind, wird die Umgebungsvariable verwendet. Siehe „Konfigurationsdateien“ [ <a href="#">SQL Anywhere Server - Datenbankadministration</a> ].                                                                                    |
| -c     | Legt die Konfigurationsoptionen fest, aktualisiert dabei nur die Optionen, die sich von den aktuellen Optionen unterscheiden, und stoppt dann den Agenten.                                                                                                                                                                                                   |
| -cr    | Legt die Konfigurationsoptionen fest, setzt dabei alle vorhandenen Optionen auf die Standardwerte zurück und stoppt dann den Agenten. Wenn Sie diese Option verwenden, werden alle Informationen in der Agenten-Datenbank zurückgesetzt. Daher sollten Sie dies nur tun, wenn sich die Agenten-Datenbank in einem nicht wiederherstellbaren Status befindet. |

| Option                                                                                                                                                                                               | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-a</b> <i>agentid</i>                                                                                                                                                                             | <p>Nur gültig mit -c oder -cr. Bestimmt die ID dieses Agenten.</p> <p>Wenn die Option -a nicht festgelegt ist, lautet der Standardwert <code>Agent_computername_UUID</code>, wobei <i>computername</i> für den Hostnamen des Computers steht, auf dem der Agent ausgeführt wird, und <i>UUID</i> für einen universell eindeutigen Bezeichner (Universally Unique Identifier = UUID).</p> <p>Es wird dringend empfohlen, nicht die Groß- und Kleinschreibung für die Unterscheidung zwischen Agenten-IDs zu benutzen. Sie sollten beispielsweise nicht die Agenten-IDs <b>Agent_XYZ</b> und <b>agent_xyz</b> als verschiedene Agenten-IDs erstellen. Wenn die konsolidierte Datenbank die Groß- und Kleinschreibung nicht berücksichtigt, muss diese Empfehlung beachtet werden. Wenn die konsolidierte Datenbank die Groß- und Kleinschreibung berücksichtigt, wird diese Empfehlung nicht erzwungen.</p> |
| <b>-db</b> <i>database location</i>                                                                                                                                                                  | Nur gültig mit -c oder -cr. Geben Sie den Pfad an, in dem entfernte Datenbanken auf dem Clientgerät gespeichert werden.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>-x</b> <i>protocol[protocol-options] ...</i><br><br><i>protocol</i> : <b>tcpip</b>   <b>tls</b><br>  <b>http</b>   <b>https</b><br><br><i>protocol-options</i> :<br>( <i>option=value</i> ; ... ) | Nur gültig mit -c oder -cr. Geben Sie die Netzwerkprotokolloptionen für den MobiLink-Client an. Diese Optionen legen fest, wie sich der Agent bei der Synchronisation der Agentendatenbank mit dem MobiLink-Server verbindet.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>-ap</b> <i>parameters</i>                                                                                                                                                                         | Bestimmt die MobiLink Authentifizierungsparameter, die bei der Synchronisation der Agenten-Datenbank verwendet werden.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>-ek</b> <i>key</i>                                                                                                                                                                                | Bestimmt den Chiffrierschlüssel für den Zugriff auf die Agenten-Datenbank.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>-n</b> <i>name</i>                                                                                                                                                                                | Bestimmt den Namen der Agenten-Datenbank. Der Standardwert ist <code>taskdb</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>-o</b> <i>file</i>                                                                                                                                                                                | Protokolliert die Ausgabe in der angegebenen Logdatei.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>-on</b> <i>size</i>                                                                                                                                                                               | Hängt <i>.old</i> an den Logdateinamen an und startet eine neue Datei mit dem ursprünglichen Namen, wenn das Log die angegebene <i>size</i> erreicht. Die Größe muss mindestens 10K betragen. Diese Option kann nicht mit -os verwendet werden.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>-os</b> <i>size</i>                                                                                                                                                                               | Benennt die Logdatei in <i>JJMMTTxx.mla</i> um und startet eine neue Datei mit dem ursprünglichen Namen, wenn das Log die angegebene <i>Größe</i> erreicht. Die Größe muss mindestens 10K betragen. Diese Option kann nicht mit -on verwendet werden.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>-ot</b> <i>file</i>                                                                                                                                                                               | Kürzt die Logdatei und protokolliert Ausgabemeldungen darin.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

| Option                    | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>-p</b> <i>password</i> | Bestimmt das MobiLink-Kennwort für die Synchronisation der Agenten-Datenbank.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>-pi</b>                | <p>Testet, ob der Agent synchronisieren kann. Diese Option veranlasst den Agenten, eine Ping-Synchronisation seiner Agentendatenbank mit dem MobiLink-Server auszuführen, wobei er die aktuell konfigurierten Netzwerkprotokolloptionen für den MobiLink-Client und Benutzerauthentifizierungsparameter verwendet. Der mlagent-Prozess gibt sofort den Wert 0 zurück, wenn die Ping-Synchronisation erfolgreich war, oder den SQL-Fehlercode der Synchronisationsanforderung, wenn die Synchronisation fehlgeschlagen ist. Weitere Hinweise zu Ping-Synchronisationen finden Sie unter „dbmlsync-Option -pi“ [<a href="#">MobiLink - Clientadministration</a>].</p> <p>Die MobiLink-Agent-Datenbank muss durch Ausführen von mlagent mit -c oder -cr konfiguriert werden, bevor mlagent mit -pi aufgerufen wird.</p> <p>Der MobiLink Agent kann nicht mit den Anweisungen -pi und entweder -c oder -cr in der Befehlszeile aufgerufen werden.</p> |
| <b>-q</b>                 | Führt den Agenten in einem minimierten Fenster aus.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>-qi</b>                | Zeigt kein Taskleistensymbol oder Fenster an.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>-u</b> <i>user</i>     | Bestimmt den MobiLink-Benutzernamen für die Synchronisation der Agenten-Datenbank.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>-v</b> <i>level</i>    | Bestimmt die Ausführlichkeitsstufe der Ausgabe von 0-9. Standardwert ist "3".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

### Beispiel

Das folgende Beispiel zeigt, wie Sie den Agenten im Konfigurationsmodus ausführen. Es verwendet die Standard-Agenten-Datenbank und setzt die Agenten-ID auf denselben Wert wie die MobiLink-Benutzer-ID:

```
mlagent -c -a username -u username -p password -x
http{host=myhost.example.com;port=8080} -o logfile.mla
```

Das folgende Beispiel zeigt, wie Sie den Agenten im Normalmodus ausführen und dabei alle Einstellungen aus der Standard-Agenten-Datenbank übernehmen:

```
mlagent
```

### Interaktive Konfiguration des MobiLink-Agenten

Der MobiLink-Agent kann entweder mit einem Konfigurationsfenster oder über die Befehlszeile konfiguriert werden. Das Konfigurationsfenster erscheint automatisch, wenn der Agent ausgeführt wird und nicht genügend Informationen für den Betrieb der Agentendatenbank vorhanden sind. Alternativ können Sie das Konfigurationsfenster über das Menü im Hauptfenster des MobiLink-Agenten anzeigen.

Unter Windows klicken Sie auf das Systemmenü in der oberen rechten Ecke des Fenster **MobiLink-Agent** und klicken Sie dann auf **Konfigurieren**. Für Windows Mobile wählen Sie in der Menüleiste das Element "Konfigurieren".

**MobiLink-Agent**

Agent-ID: AID\_JOHN

Benutzer: JOHN Kennwort: ...

Kennwort wiederholen: ...

Authentifizierungsparameter:

Netzwerkprotokolloptionen für MobiLink-Client:

tcpip{host=localhost};port=2439}

Beispiel: http(host=relayserver.sybase.com)  
Kann vom Server nach der nächsten Synchronisation außer Kraft gesetzt werden.

Standort der entfernten Datenbank:

OK Abbrechen

Die folgende Tabelle zeigt, wie die Felder des Konfigurationsfensters mit den Konfigurationsoptionen des MobiLink-Agenten zusammenhängen.

| Feldname im Konfigurationsfenster             | Äquivalente mlagent-Konfigurationsoption |
|-----------------------------------------------|------------------------------------------|
| Agent-ID                                      | Option -a für mlagent                    |
| Benutzer                                      | Option -u für mlagent                    |
| Kennwort                                      | Option -p für mlagent                    |
| Authentifizierungsparameter                   | Option -ap für mlagent                   |
| Netzwerkprotokolloptionen für MobiLink-Client | Option -x für mlagent                    |
| Standort der entfernten Datenbank             | Option -db für mlagent                   |

Weitere Hinweise zu Konfigurationsoptionen für den MobiLink-Agenten finden Sie unter „[MobiLink-Agent auf dem Clientgerät](#)“ auf Seite 158.

## Verschlüsselung

Wenn die Agentendatenbank nicht existiert, werden Sie in einer Meldung gefragt, ob ein Standard-Chiffrierschlüssel verwendet werden soll. Der Standardschlüssel ist schwer zu erraten, kann aber ermittelt werden, da er im Agenten festkodiert wird. Wenn Sie nicht den Standard-Chiffrierschlüssel verwenden möchten, wird das folgende Fenster angezeigt, damit Sie den Chiffrierschlüssel eingeben und überprüfen können:

The screenshot shows a dialog box titled "MobiLink-Agent". It contains two text input fields. The first field is preceded by the label "Geben Sie den Datenbank-Chiffrierschlüssel ein:". The second field is preceded by the label "Geben Sie den Chiffrierschlüssel erneut ein:". To the right of the first field are two buttons: "OK" and "Abbrechen".

Wenn die Agentendatenbank vorhanden ist, aber der MobiLink-Agent keine Verbindung zu ihr herstellen kann, weil ein Nicht-Standard-Chiffrierschlüssel nicht mit der Option `-ek` angegeben wurde, wird das folgende Fenster angezeigt, damit Sie den Schlüssel eingeben können:

The screenshot shows a dialog box titled "MobiLink-Agent". It contains one text input field preceded by the label "Geben Sie den Datenbank-Chiffrierschlüssel ein:". To the right of the field are two buttons: "OK" and "Abbrechen".

## Dienstprogramm zum MobiLink-Agent-Stopp

Mit dem Dienstprogramm zum MobiLink-Agent-Stopp können Sie eine Instanz des MobiLink-Agenten stoppen, die auf demselben Gerät ausgeführt wird wie das Dienstprogramm.

### Syntax

**mlastop** [ *options* ]

| Option               | Beschreibung                                                                                                                              |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-n Name</code> | Der Name des Agenten, der gestoppt werden soll. Wenn <code>-n</code> nicht angegeben ist, werden alle Agenten der auf dem Gerät gestoppt. |

## Siehe auch

- „mlagent-Befehl“ auf Seite 159
- „Bereitstellung von SQL Anywhere MobiLink-Clients“ auf Seite 782
- „Deployment des MobiLink-Clients für UltraLite“ auf Seite 785

## MobiLink-Agenten in Sybase Central

Nachdem der Agent auf dem Clientgerät mithilfe des mlagent Befehls erstellt und konfiguriert wurde, muss er auch in Sybase Central erstellt werden, bevor ihm Aufgaben zugeordnet werden können. Der **Assistent zum Erstellen von MobiLink-Agenten** führt Sie durch die Erstellung des Agenten, bei der Sie die Informationen angeben, die für die Kennzeichnung des Agenten erforderlich ist.

Sie müssen den Namen eines entfernten Schemas definieren, bevor Sie in Sybase Central einen Agenten erstellen können.

Siehe:

- „Namen für entfernte Schemata hinzufügen“ auf Seite 164
- „Namen für entfernten Schemata importieren“ auf Seite 165
- „Agenten hinzufügen“ auf Seite 165
- „Agenteneigenschaften“ auf Seite 166
- „Agenteneigenschaften anzeigen und ändern“ auf Seite 166
- „Verwaltete entfernte Datenbanken hinzufügen“ auf Seite 167
- „Eine Gruppe hinzufügen“ auf Seite 167

## Namen für entfernte Schemata hinzufügen

Verwenden Sie den **Assistenten zur Erstellung eines Namens für ein entferntes Schema**, um einen Namen für das entfernte Schema hinzuzufügen.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Kontext und Bemerkungen

Der Name eines entfernten Schemas muss definiert sein, bevor Sie in Sybase Central einen Agenten erstellen können.

### Aufgabe

1. Doppelklicken Sie auf das MobiLink-Projekt.

Doppelklicken Sie auf **Namen für entfernte Schemata** und klicken Sie dann auf **Neu » Name für entferntes Schema**.

2. Befolgen Sie die Anweisungen im **Assistenten zur Erstellung eines Namens für ein entferntes Schema** und klicken Sie auf **Fertig stellen**.

## Ergebnisse

Der Name des entfernten Schemas wird hinzugefügt.

## Namen für entfernten Schemata importieren

Schemanamen können aus einer anderen Datenbank importiert werden.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Aufgabe

1. Doppelklicken Sie auf das MobiLink-Projekt.
2. Rechtsklicken Sie auf **Namen für entfernte Schemata** und klicken Sie auf **Importieren**.
3. Wählen Sie die Datenbank, aus der Sie die Namen für entfernte Schemata importieren möchten, in der angezeigten Liste der konsolidierten Datenbanken aus und klicken Sie auf **OK**.

## Ergebnisse

Wenn die ausgewählte Datenbank Namen für entfernte Schemata enthält, die nicht bereits im MobiLink-Projekt vorhanden sind, werden diese importiert.

## Agenten hinzufügen

Fügen Sie einen Agenten, um die zentrale Administration von entfernten Datenbanken nutzen zu können.

### Voraussetzungen

Der Name eines entfernten Schemas muss definiert sein, bevor Sie in Sybase Central einen Agenten erstellen können.

### Aufgabe

1. Doppelklicken Sie auf das MobiLink-Projekt.
2. Doppelklicken Sie auf **Konsolidierte Datenbanken**.
3. Doppelklicken Sie auf **Agenten** und klicken Sie dann auf **Datei » Neu » Agent**.
4. Befolgen Sie die Anweisungen im **Assistenten zum Erstellen von MobiLink-Agenten**.

## Ergebnisse

Der Agent wird erstellt.

## Nächste Schritte

Wenn der Agent in Sybase Central erstellt wurde, können Sie folgende Aufgaben ausführen:

- Agenteneigenschaften anzeigen und ändern
- Entfernte Datenbanken hinzufügen, die ein Agent verwalten soll
- Agenten synchronisieren
- Agenten löschen
- Ereignisse für einen Agenten anzeigen
- Aufgaben für einen Agenten anzeigen
- Informationen über entfernte Datenbanken anzeigen, die von einem Agenten verwaltet werden

## Agenteneigenschaften

Agenteneigenschaften können über das Fenster **Eigenschaften des Agenten** in Sybase Central angezeigt und bearbeitet werden. Außerdem können Sie die folgenden Eigenschaften festlegen, indem Sie mit der rechten Maustaste auf einen Agenten klicken, **Einstellen** wählen und anschließend die gewünschte Eigenschaft:

- **Synchronisationsintervall** Das Synchronisationsintervall steuert, wie häufig der Agent seine Agentendatenbank synchronisiert.
- **Administrations-Abrufintervall** Das Administrations-Abrufintervall legt fest, wie häufig der Agent nachsieht, ob Anforderungen vom Server eingetroffen sind, eine Synchronisation oder andere Aktionen durchzuführen.
- **Netzwerkprotokolloptionen für MobiLink-Client** Die Netzwerkprotokolloptionen für den MobiLink-Client sind eine Agenteneigenschaft, die auf dem Client angegeben und bei der ersten Synchronisation des Agenten an den Server gesendet wird. Wenn ein Administrator die MobiLink-Protokolloptionen ändert, wird der neue Wert bei der Synchronisation an den Agenten gesendet und der Agent verwendet den neuen Wert für die gesamte nachfolgende MobiLink-Kommunikation.

Wenn ein Administrator ungültige MobiLink-Kommunikationsoptionen sendet (z.B. einen falsch festgelegten Server-Hostnamen), kann es für den Agenten, der diesen Wert empfängt, unmöglich werden, mit dem MobiLink-Server zu kommunizieren. In diesem Fall muss der Administrator die Netzwerkprotokolloptionen für den MobiLink-Client in Sybase Central korrigieren und dann den Agenten auf dem Gerät mit den richtigen Optionen neu konfigurieren lassen.

- **Verbindungszeichenfolgen für verwaltete Datenbanken** Die Verbindungszeichenfolge(n), die der MobiLink-Agent verwenden kann, um eine Verbindung mit der oder den entfernten Datenbank(en) herzustellen.

## Agenteneigenschaften anzeigen und ändern

Sie können Agenteneigenschaften anzeigen oder aktualisieren.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Aufgabe

1. Doppelklicken Sie auf die konsolidierte Datenbank.
2. Doppelklicken Sie auf **Agenten**, rechtsklicken Sie auf den Agenten, mit dem Sie arbeiten möchten, und klicken Sie auf **Eigenschaften**.
3. Falls erforderlich, nehmen Sie Änderungen an den Eigenschaften vor und klicken Sie auf **Übernehmen**.

### Ergebnisse

Die Agenteneigenschaften werden aktualisiert.

## Verwaltete entfernte Datenbanken hinzufügen

Damit ein Agent eine entfernte Datenbank verwalten kann, muss die Datenbank in Sybase Central mit dem Agenten verknüpft werden. Dies erfolgt durch die Angabe eines Namen für das entfernte Schema.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Aufgabe

1. Doppelklicken Sie auf die konsolidierte Datenbank.
2. Doppelklicken Sie auf den Agenten, mit dem Sie arbeiten möchten, und klicken Sie auf **Verwaltete entfernte Datenbank hinzufügen**.
3. Die verfügbaren entfernten Datenbanken werden durch Namen für entfernte Schemata dargestellt. Wählen Sie einen **Namen für das entfernte Schema** aus der Dropdown-Liste.
4. Geben Sie eine Verbindungszeichenfolge für die entfernte Datenbank ein und klicken Sie auf **OK**.

Die Verbindungszeichenfolge wird auf dem Clientgerät verwendet. Alle ODBC-Datenquellen oder Pfade und/oder Datenbankdateien müssen für das Gerät gültig sein.

### Ergebnisse

Die Datenbank wird hinzugefügt.

## Eine Gruppe hinzufügen

Eine Gruppe ist eine Sammlung von MobiLink-Agenten.

## Voraussetzungen

Sie müssen einen oder mehrere Agenten definiert haben, bevor Sie eine Gruppe erstellen können.

## Aufgabe

1. Vergewissern Sie sich, dass Ihre konsolidierten Datenbanken ausgeführt werden.
2. Doppelklicken Sie auf das MobiLink-Projekt.
3. Doppelklicken Sie auf den Projektnamen und klicken Sie auf **Neu » Gruppe**.
4. Befolgen Sie die Anweisungen im **Assistenten zum Erstellen einer Gruppe** und klicken Sie auf **Fertig stellen**.

## Ergebnisse

Die Gruppe wird erstellt.

## Agentenauthentifizierung

Der Agent agiert bei der Synchronisation der Agenten-Datenbank als MobiLink-Synchronisationsclient. Die Synchronisationsskripten zur Unterstützung des Agenten verwenden die Skriptversion *ml\_ra\_agent\_16*. Die Agenten-Synchronisationsskripten werden automatisch installiert, wenn Sie ein MobiLink-Setup für eine konsolidierte Datenbank ausführen. Es werden jedoch keine Authentifizierungsskripten für den Agenten zur Verfügung gestellt. Siehe [„Einrichten einer konsolidierten Datenbank“ auf Seite 3](#).

Mindestens eines der folgenden Ereignisse muss definiert werden, um ein sicheres System zu gewährleisten:

- [„authenticate\\_user \(Verbindungsereignis\)“](#)
- [„authenticate\\_user\\_hashed \(Verbindungsereignis\)“](#)
- [„authenticate\\_parameters \(Verbindungsereignis\)“](#)

Dazu können Sie eine der folgenden Methoden verwenden:

- Rufen Sie die gespeicherte Prozedur `ml_add_connection_script` auf. Beispiel:

```
ml_add_connection_script( 'ml_ra_agent_16', 'authenticate_user', '<DEFINE  
YOUR AUTHENTICATION LOGIC>' )
```

- Verwenden Sie **Verbindungsskripten** in Sybase Central. Sie können die Skriptversion `ml_ra_agent_16` oder `ml_global` verwenden. In der Regel sollten Sie für die Anwendungsdatensynchronisation und für den Agenten dieselbe Authentifizierung verwenden. Durch die Verwendung von `ml_global` können Sie einfach eine Gruppe von Authentifizierungsskripten für beides definieren. Dies ist die empfohlene Methode für die Authentifizierung. Siehe [Skriptversion ml\\_global auf Seite 256](#).

## Entfernte Aufgaben

Eine entfernte Aufgabe ist die Arbeitseinheit beim Ausführen der zentralen Administration von entfernten Datenbanken. Eine entfernte Aufgabe umfasst folgende Komponenten:

- Einen oder mehrere Auslösemechanismen
- Optionale Bedingungen
- Eine sortierte Sammlung von Befehlen
- Andere Eigenschaften

Aufgaben werden vom Administrator mit dem MobiLink 16-Plug-In für Sybase Central erstellt. Im Design-Schritt werden sie im Projekt auf dem Computer des Administrators lokal gespeichert. Eine Liste der Befehle, die zum Erstellen von Aufgaben verwendet werden können, finden Sie unter [„Befehle“ auf Seite 181](#).

Im Allgemeinen gilt, dass eine Aufgabe nicht von der Fertigstellung einer anderen abhängig sein sollte. Sie können jedoch eine Aufgabe erstellen, die von einer anderen abhängt, indem Sie eine Aufgabe etwas in die entfernte Datenbank schreiben und diesen Wert durch die Bedingung einer anderen Aufgabe abfragen lassen.

Wenn der Administrator die Agenten für den Empfang einer Aufgabe vorbereitet hat, stellt er die Aufgabe bereit. Bereitgestellte Aufgaben werden in die konsolidierte Datenbank kopiert. Es gibt nun zwei Kopien der Aufgabe: die bereitgestellte Aufgabe in der konsolidierten Datenbank und die im Design-Schritt befindliche Aufgabe im Projekt. Bereitgestellte Aufgaben können nicht geändert werden. Sie können jedoch die bei der Erstellung verwendete, im Design-Schritt befindliche Aufgabe ändern und erneut bereitstellen (um eine zweite bereitgestellte Aufgabe zu erstellen). Bereitgestellte Aufgaben können abgebrochen, initiiert (SIRT), reaktiviert und neuen Empfängern zugewiesen werden.

Nachdem eine Aufgabe bereitgestellt wurde, kann sie einem oder mehreren Agenten zur Ausführung zugewiesen werden. Wenn die Aufgabe einem Agenten zugewiesen wurde, wird sie in den Agenten heruntergeladen. Der Agent führt dann die Aufgabe zum geeigneten Zeitpunkt aus und sendet optional die Ergebnisse per Upload zurück an die konsolidierte Datenbank, wo sie von einem Administrator mit dem MobiLink 16-Plug-In für Sybase Central überprüft werden können.

Aufgaben haben die folgenden Attribute:

- **Name** Eine entfernte Aufgabe hat zwei Namen: eine kennzeichnet die im Design-Schritt befindliche, im Projekt gespeicherte Version der Aufgabe, der andere die bereitgestellte Aufgabe. Häufig sind die beiden Namen identisch.

Der Design-Name wird der Aufgabe bei der Erstellung zugewiesen und muss unter den Aufgaben im Projekt eindeutig sein. Der Name einer bereitgestellten Aufgabe wird bei der Bereitstellung zugewiesen und dieser Name muss unter den bereitgestellten Aufgaben in der konsolidierten Datenbank eindeutig sein.

- **Beschreibung** Eine Beschreibung der Aufgabe kann eingegeben werden und beliebigen Text enthalten, den Sie der Aufgabe zuordnen möchten. Die Beschreibung wird im Projekt und in der

konsolidierten Datenbank (sobald die Aufgabe bereitgestellt wurde) gespeichert, aber nicht an den Agenten gesendet.

- **Auslösemechanismen** Die Auslösemechanismen einer entfernten Aufgabe bestimmen, wann der Agent die Aufgabe auszuführen versucht. Eine Aufgabe kann mehr als einen Auslösemechanismus aufweisen. Es gibt drei unterstützte Auslösemechanismen:
  - **Basiert auf einem Zeitplan** Die Aufgabe wird zu bestimmten Zeiten bzw. in bestimmten Zeitintervallen ausgelöst. Diese Option muss für eine Aufgabe explizit festgelegt werden.
  - **Beim Empfang von einem Agenten** Die Aufgabe wird ausgelöst, wenn sie vom Agenten empfangen wurde, und nur einmal ausgeführt. Diese Option muss für eine Aufgabe explizit festgelegt werden.
  - **Bei Bedarf** Die Aufgabe kann jederzeit durch eine Nachricht vom Server ausgelöst werden. Dieser Prozess wird als "serverinitiierte entfernte Aufgabe" (SIRT) bezeichnet. Alle Aufgaben, die nicht so konfiguriert sind, dass sie nur einmal ausgeführt werden, unterstützen die Auslöseoption **Bei Bedarf**.
- **Bedingungen** Bevor eine Aufgabe ausgeführt werden kann, müssen alle ihre Bedingungen erfüllt sein. Wenn eine Aufgabe ausgelöst wird, aber nicht alle ihre Bedingungen erfüllt sind, gilt die Ausführung als fehlgeschlagen und die Aufgabe muss erneut ausgelöst werden, bevor die Bedingungen neu ausgewertet werden.
- **Name des entfernten Schemas** Eine Aufgabe kann optional mit dem Namen eines entfernten Schemas verknüpft werden. Aufgaben, die einen Zugriff auf eine entfernte Datenbank erfordern, müssen mit dem Namen eines entfernten Schemas verknüpft werden. Der Name des entfernten Schemas wird vom Agenten verwendet, um zu bestimmen, auf welche der Datenbanken der Agent bei der Ausführung der Aufgabe zugreifen muss. Aufgaben müssen mit dem Namen eines entfernten Schemas verknüpft werden, wenn sie einen der folgenden Befehle enthalten: "Datenbank erstellen", "Datenbank löschen", "SQL ausführen" oder "Synchronisieren". Außerdem muss eine entfernte Aufgabe mit dem Namen eines entfernten Schemas verknüpft werden, wenn mithilfe einer SQL-Bedingung ermittelt werden soll, ob die Aufgabe ausgeführt werden kann.
- **Befehle** Eine Aufgabe enthält eine geordnete Gruppe von Befehlen, die die erforderliche Arbeit ausführen. Die Reihenfolge, in der Sie die Befehle angeben, bestimmt die Reihenfolge, in der die Befehle innerhalb der Aufgabe ausgeführt werden. Es ist wichtig, auf die Reihenfolge der Befehle zu achten, weil Befehle voneinander abhängig sein könnten. Siehe „[Befehle](#)“ auf Seite 181.
- **Maximale Anzahl der Neuversuche** Jeder Befehl hat eine Aktion **Bei Fehler**, die verwendet werden kann, um eine erneute Ausführung der Aufgabe bzw. des Befehls zu veranlassen, wenn der Befehl fehlschlägt. Mit dieser Option können Sie die Anzahl der Neuversuche begrenzen, die während eines einzigen Ausführungsversuchs zulässig sind.
- **Zeitspanne zwischen Neuversuchen** Diese Option legt die Zeitspanne fest, die nach einem fehlgeschlagenen Befehl gewartet werden soll, bevor der Befehl bzw. die Aufgabe erneut versucht wird. Durch diese Verzögerung kann eine vorübergehende Bedingung (z.B. eine gesperrte Datenbanktabelle oder eine gesperrte Datei), die den Fehlschlag verursacht hat, möglicherweise verschwinden, bevor der Befehl bzw. die Aufgabe erneut versucht wird. Es wird davon ausgegangen,

dass es sich bei Zeitspannen zwischen Neuversuchen um "kurze" Zeiträume handelt. Eine Aufgabenbedingung wird nicht zwischen Neuversuchen neu ausgewertet.

- **Maximale Laufzeit** Es ist möglich, dass eine Aufgabe sich bei der Ausführung nicht so verhält wie vom Administrator vorgesehen. Eine Betriebssystemaufruf kann hängen, ein Synchronisierungsversuch sehr langsam sein – eine SQL-Anweisung kann auf einer anderen Verbindung blockiert werden, die die Datenbank verwendet. Durch das Festlegen einer maximalen Laufzeit für eine Aufgabe können Sie begrenzen, wie lange die Ausführung der Aufgabe dauern darf. Wenn die maximale Laufzeit erreicht ist, wird die Aufgabe beendet. (Der tatsächliche Zeitpunkt, zu dem die Aufgabe beendet wird, hängt davon ab, ob es möglich ist, den Vorgang zu unterbrechen.) Der Status für die Aufgabe wird entsprechend dem Timeout gesetzt und die Aufgabe wird erst dann wieder versucht, wenn sie erneut ausgelöst wird. Wenn ein Befehl in einer Aufgabe fehlschlägt und die Aufgabe bzw. der Befehl erneut versucht werden muss, wird die maximale Laufzeit nach Ablauf der Verzögerung zurückgesetzt. Daher gilt die maximale Laufzeit als maximale Laufzeit pro Ausführung bzw. Neuversuch. Sie umfasst nicht die aggregierte Zeit aller Neuversuche und enthält auch keine über die Eingabeaufforderung eingegebenen Befehle.
- **Schemaänderung** Schemaänderungsaufgaben ändern das Schema für eine entfernte Datenbank. Wenn die Aufgabe erfolgreich ist, wird der Schemaname für die entfernte Datenbank ebenfalls aktualisiert. Schemaänderungsaufgaben sind immer Aufgaben mit hoher Priorität und melden nach der Ausführung ihren Status.
- **Hohe Priorität** Aufgaben mit hoher Priorität werden immer ausgelöst, wenn sie von einem Agenten empfangen werden, und ausgeführt, sobald alle noch laufenden Aufgaben abgeschlossen sind. Sie können nicht basierend auf einem Zeitplan ausgeführt werden und haben möglicherweise keine Bedingungen für ihre Ausführung. Eine mit "Hohe Priorität" markierte Aufgabe wird erst dann ausgeführt, wenn keine anderen Aufgaben mehr ausgeführt werden, um zu gewährleisten, dass keine andere Aufgabe die Ausführung einer Aufgabe mit hoher Priorität beeinträchtigt.
- **Statusmeldung** Mithilfe der Statusmeldungsoptionen für eine Aufgabe können Sie festlegen, ob und wann Ergebnisse an die konsolidierte Datenbank zurückgegeben werden, sowohl bei erfolgreich abgeschlossenen Aufgaben als auch bei fehlgeschlagenen. Folgende Optionen sind verfügbar:
  - **Nur Status senden** Aufgabenergebnisse werden nicht gemeldet. Informationen über die Anzahl von erfolgreichen und fehlgeschlagenen Aufgaben werden jedoch gespeichert und gemeldet.
  - **Ergebnisse bei der nächsten Synchronisation der Agenten-Datenbank zurückgeben** Aufgabenergebnisse und Status werden gespeichert und bei der nächsten Synchronisation der Agenten-Datenbank gemeldet.
  - **Ergebnisse sofort nach Abschluss der Aufgabenausführung zurückgeben** Aufgabenergebnisse und Status werden gemeldet, sobald die Aufgabe abgeschlossen ist.
- **Zufälliges Verzögerungsintervall** Wenn eine bestimmte Aufgabe nach dem Ausführen Ergebnisse an den MobiLink-Server sendet oder bewirkt, dass eine entfernte Datenbank eine Synchronisation mit dem Server durchführt, und die Aufgabe gleichzeitig für eine große Anzahl von entfernten Datenbanken ausgelöst wird, dann verteilt eine zufällige Verzögerung für die Aufgabe die Arbeitslast der Synchronisation für den Server gleichmäßig über eine konfigurierbare Zeitspanne.

Eine entfernte Aufgabe kann ein zufälliges Verzögerungsintervall haben, d.h. ein Intervall  $N$ , in Sekunden, mit dem jeder Agent eine zufällige Anzahl von Sekunden in  $[0, N)$  generiert, um die jede Ausführung der Aufgabe verzögert wird. Wenn die Aufgabe eine geplante Aufgabe ist, wird die zufällige Verzögerung vor der ersten Ausführung der Aufgabe generiert und für jede Ausführung verwendet. Die Aufgabe wird zu den geplanten Zeiten ausgeführt, verschoben um die zufällige Verzögerung. Damit wird sichergestellt, dass die Deltas der Aufgabenausführungszeiten mit dem Zeitplan übereinstimmen.

Es wird nicht empfohlen, das zufällige Verzögerungsintervall größer zu wählen als die kleinste Delta-Zeit einer geplanten Aufgabe. Wenn die Aufgabe **bei Bedarf** läuft, d.h. vom Server initiiert wird, wird die zufällige Verzögerung generiert und jedes Mal, wenn die Aufgabe initiiert wird, dazu verwendet, die Ausführung zu verzögern. Wenn die Aufgabe **nach Empfang** läuft, wird die zufällige Verzögerung generiert und bei der ersten und einzigen Ausführung der Aufgabe dazu verwendet, die Ausführung zu verzögern.

### Siehe auch

- „Logik für entfernte Aufgaben“ auf Seite 172
- „Entfernte Aufgaben erstellen“ auf Seite 173
- „Entfernte Aufgaben bearbeiten“ auf Seite 174
- „Entfernte Aufgaben bereitstellen“ auf Seite 174
- „Bereitgestellte entfernte Aufgaben“ auf Seite 176

## Logik für entfernte Aufgaben

Im Folgenden finden Sie eine Beschreibung der Logik zum Ausführen einer entfernten Aufgabe.

```
current_command = 1;
num_tries = 0;
EXECUTE_TASK:
loop {
    num_tries = num_tries + 1;
    EXECUTE_COMMANDS;
    if( task_success or task_abort ) break EXECUTE_TASK;
    if( task_retry and at maximum tries ) {
        break EXECUTE_TASK;
    } else {
        continue;
    }
}

EXECUTE_COMMANDS:
for each command starting at current_command {
    execute current_command;
    if( command_failed ) {
        if( action on failure is "abort task" ) {
            break EXECUTE_COMMANDS, returning task_abort;
        } else if( action on failure is "continue" ) {
            // no action, continue at next command
        } else if( action on failure is "retry task" ) {
            current_command = 1;
            break EXECUTE_COMMANDS, returning task_retry;
        } else if( action on failure is "retry command" ) {
            // no change to current_command
            break EXECUTE_COMMANDS, returning task_retry;
        }
    }
}
```

```

    }
    current_command = next command number;
}
return task_success;

```

Die maximale Laufzeit für eine Aufgabe wird zurückgesetzt, wenn die Aufgabe gestartet wird und wenn ein Befehl oder die gesamte Aufgabe erneut versucht wird, weil ein Befehl fehlgeschlagen ist.

Wenn ein Befehl oder eine Aufgabe erneut versucht wird, weil ein Befehl fehlgeschlagen ist, wird die Bedingung nicht erneut ausgewertet und es ist kein neues Trigger-Ereignis erforderlich. Es ist wichtig zu bedenken, ob in der Bedingung referenzierte Eigenschaften sich während der Ausführung der Aufgabe ändern können und ob ein neuer Versuch nach einem Fehlschlag der Bedingung unerwünschte Ergebnisse liefern würde.

## Entfernte Aufgaben erstellen

Aufgaben werden im Kontext eines MobiLink-Projekts definiert und verwaltet.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Kontext und Bemerkungen

Ein Administrator kann eine Aufgabe erstellen, ändern und entfernen, ohne dass eine Verbindung mit der konsolidierten Datenbank erforderlich ist. Aufgaben, die noch nicht bereitgestellt wurden, gelten als in der Entwicklung befindlich und sind nur lokal im MobiLink-Projekt vorhanden.

### Aufgabe

1. Doppelklicken Sie auf das MobiLink-Projekt.
2. Doppelklicken Sie auf **Entfernte Aufgaben** und klicken Sie auf **Neu » Entfernte Aufgabe**.
3. Befolgen Sie die Anweisungen im **Assistenten für die Erstellung einer entfernten Aufgabe** und klicken Sie auf **Fertig stellen**.

#### Hinweis

Wenn Sie einen neuen Namen für ein entferntes Schema erstellen müssen, der eine Gruppe von entfernten Datenbanken mit demselben Schema kennzeichnet, klicken Sie auf der Seite **Willkommen des Assistenten für die Erstellung einer entfernten Aufgabe** auf **Namen für ein entferntes Schema erstellen**. Der **Assistent zur Erstellung eines Namens für ein entferntes Schema** erscheint und der **Assistent für die Erstellung einer entfernten Aufgabe** bleibt geöffnet. Sobald Sie den Namen für das neue entfernte Schema erstellt haben, erscheint er in der Dropdown-Liste **Name für entferntes Schema** des **Assistenten für die Erstellung einer entfernten Aufgabe**.

4. Befolgen Sie die Vorgehensweise, um der Aufgabe einen oder mehrere Befehle hinzuzufügen. Siehe [„Befehle zu entfernten Aufgaben hinzufügen“ auf Seite 186](#).

### Ergebnisse

Die Aufgabe wird erstellt.

### Nächste Schritte

Nach dem Erstellen kann die Aufgabe in der konsolidierten Datenbank bereitgestellt und anschließend Empfängern (Agenten) zugeordnet werden: Siehe [„Entfernte Aufgaben bereitstellen“ auf Seite 174](#).

## Entfernte Aufgaben bearbeiten

Eigenschaften von entfernten Aufgaben können bearbeitet werden.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Aufgabe

1. Doppelklicken Sie auf das MobiLink-Projekt.
2. Klicken Sie auf **Eigenschaften**.
3. Bearbeiten Sie die Eigenschaften der entfernten Aufgabe.
4. Wenn Sie die Bearbeitung der Eigenschaften der entfernten Aufgabe abgeschlossen haben, klicken Sie auf **Übernehmen**.

### Ergebnisse

Die Eigenschaften der entfernten Aufgabe werden aktualisiert.

### Nächste Schritte

Eine entfernte Aufgabe kann bereitgestellt und Agenten zugeordnet werden. Siehe [„Entfernte Aufgaben bereitstellen“ auf Seite 174](#).

## Entfernte Aufgaben bereitstellen

Wenn Sie bereit sind, die Aufgabe im System hinzuzufügen, müssen Sie die Aufgabe bereitstellen, d.h. ein Deployment ausführen.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

## Kontext und Bemerkungen

Das Deployment einer entfernten Aufgabe bedeutet, dass die Aufgabe in die konsolidierte Datenbank kopiert und der neuen Kopie ein Name zugeteilt wird. Der Name der bereitgestellten Aufgabe ist häufig derselbe Name, den die Aufgabe während der Entwicklung hatte.

Im Rahmen des Deployments können Sie eine Aufgabe auch einer Liste von Agenten zuordnen. Agenten können nach dem Deployment einer Aufgabe hinzugefügt werden, sodass es nicht notwendig ist, dies beim ersten Deployment zu tun. Dies ist nützlich, wenn neue Agenten zum System hinzugefügt werden, nachdem das Deployment einer Aufgabe bereits ausgeführt wurde.

## Aufgabe

Eine entfernte Aufgabe muss mindestens einen Befehl enthalten bevor sie bereitgestellt werden kann. Hinweise zu Befehlen finden Sie unter [„Befehle“ auf Seite 181](#) und [„Befehle zu entfernten Aufgaben hinzufügen“ auf Seite 186](#).

1. Doppelklicken Sie auf das MobiLink-Projekt.
2. Rechtsklicken Sie auf die entfernte Aufgabe, die Sie bereitstellen möchten, und klicken Sie auf **Bereitstellen**.
3. Befolgen Sie die Anweisungen des **Assistenten für das Deployment einer entfernten Aufgabe**.
4. Klicken Sie auf **Fertig stellen**.

## Ergebnisse

Die entfernte Aufgabe wird bereitgestellt.

## Nächste Schritte

Bereitgestellte Aufgaben können abgebrochen, initiiert und reaktiviert werden sowie Empfänger hinzugefügt bekommen. Siehe [„Bereitgestellte entfernte Aufgaben“ auf Seite 176](#).

## Entfernte Aufgaben exportieren

Entfernte Aufgaben können in eine Datei exportiert werden.

### Voraussetzungen

Eine entfernte Aufgabe muss definiert sein.

### Aufgabe

1. Doppelklicken Sie auf das MobiLink-Projekt.
2. Rechtsklicken Sie auf die entfernte Aufgabe, die Sie exportieren möchten, und klicken Sie auf **Exportieren**.

3. Geben Sie einen Namen und einen Speicherort für die Datei ein. Klicken Sie auf **Speichern**.

## Ergebnisse

Die entfernte Aufgabe wird in die angegebene Datei exportiert.

## Bereitgestellte entfernte Aufgaben

Sie können weiterhin mit einer entfernten Aufgabe arbeiten, nachdem diese bereitgestellt wurde. Bereitgestellte Aufgaben können abgebrochen, initiiert und reaktiviert werden sowie Empfänger hinzugefügt bekommen.

Bereitgestellte entfernte Aufgaben finden Sie in Sybase Central an den folgenden Stellen:

- Wenn Sie auf der Ebene eines einzelnen Agenten mit bereitgestellten entfernten Aufgaben arbeiten möchten, vergewissern Sie sich, dass die konsolidierte Datenbank, mit der Sie arbeiten, in der Ansicht **Ordner** des MobiLink 16-Plug-Ins für Sybase Central erweitert ist. Erweitern Sie **Agenten**, klicken Sie auf den Agenten, mit dem Sie arbeiten möchten, und klicken Sie anschließend im rechten Fensterausschnitt auf die Registerkarte **Aufgaben**. Sie enthält eine Liste aller entfernten Aufgaben, die für den ausgewählten Agenten bereitgestellt wurden.
- Um für alle Agenten mit bereitgestellten entfernten Aufgaben arbeiten zu können, vergewissern Sie sich, dass das MobiLink-Projekt, mit dem Sie arbeiten, im linken Fensterausschnitt in der Ansicht **Ordner** ausgewählt ist, klicken Sie mit der rechten Maustaste auf **Entfernte Aufgaben** und klicken Sie auf **Bereitgestellte Aufgaben**. Bereitgestellte entfernte Aufgaben werden für alle Agenten aufgelistet.

### Siehe auch

- „Bereitgestellte entfernte Aufgaben für alle Agenten abbrechen“ auf Seite 176
- „Bereitgestellte entfernte Aufgaben für einen einzelnen Agenten abbrechen“ auf Seite 177
- „Bereitgestellte entfernte Aufgaben für alle Agenten einleiten“ auf Seite 177
- „Bereitgestellte entfernte Aufgaben für einen einzelnen Agenten einleiten“ auf Seite 178
- „Bereitgestellte entfernte Aufgaben für einen einzelnen Agenten reaktivieren“ auf Seite 178
- „Empfänger für eine bereitgestellte entfernte Aufgabe hinzufügen“ auf Seite 179

## Bereitgestellte entfernte Aufgaben für alle Agenten abbrechen

Es kann sinnvoll sein, eine bereitgestellte entfernte Aufgabe abzubrechen, wenn sie nicht mehr benötigt wird oder wenn Sie ein Upgrade des entfernten Schemas durchführen und damit eine neue Version der Aufgabe bereitstellen.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Aufgabe

1. Doppelklicken Sie auf das MobiLink-Projekt.

2. Klicken Sie auf **Entfernte Aufgaben » Bereitgestellte Aufgaben** und wählen Sie die bereitgestellte Aufgabe aus, die Sie abbrechen möchten.
3. Rechtsklicken Sie auf die bereitgestellte Aufgabe, die Sie abbrechen möchten, und klicken Sie auf **Für alle Empfänger abbrechen**.

### Ergebnisse

Die Aufgabe wird abgebrochen.

## Bereitgestellte entfernte Aufgaben für einen einzelnen Agenten abbrechen

Es kann sinnvoll sein, eine bereitgestellte entfernte Aufgabe abzubrechen, wenn sie nicht mehr benötigt wird oder wenn Sie ein Upgrade des entfernten Schemas durchführen und damit eine neue Version der Aufgabe bereitstellen.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Aufgabe

1. Doppelklicken Sie auf die konsolidierte Datenbank.
2. Doppelklicken Sie auf **Agenten** und klicken Sie auf den Agenten, mit dem Sie arbeiten möchten.
3. Im rechten Fensterausschnitt klicken Sie auf die Registerkarte **Aufgaben**.
4. Klicken Sie mit der rechten Maustaste auf die bereitgestellte Aufgabe, die Sie abbrechen möchten, und klicken Sie auf **Abbrechen**.

### Ergebnisse

Ergebnis

## Bereitgestellte entfernte Aufgaben für alle Agenten einleiten

Das Einleiten einer bereitgestellten entfernten Aufgabe bewirkt, dass der Server den Agenten auf dem Client benachrichtigt, dass die Aufgabe sofort ausgeführt werden soll.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Aufgabe

1. Doppelklicken Sie auf das MobiLink-Projekt.
2. Klicken Sie auf **Entfernte Aufgaben » Bereitgestellte Aufgaben** und wählen Sie die bereitgestellte Aufgabe aus, die Sie einleiten möchten.

3. Rechtsklicken Sie auf die bereitgestellte Aufgabe, die Sie einleiten möchten, und klicken Sie auf **Für alle Empfänger einleiten**.

### **Ergebnisse**

Die Aufgabe wird für alle Agenten eingeleitet.

## **Bereitgestellte entfernte Aufgaben für einen einzelnen Agenten einleiten**

Das Einleiten einer bereitgestellten entfernten Aufgabe bewirkt, dass der Server den Agenten auf dem Client benachrichtigt, dass die Aufgabe sofort ausgeführt werden soll.

### **Voraussetzungen**

Es gibt keine Voraussetzungen für diese Aufgabe.

### **Aufgabe**

1. Doppelklicken Sie auf die gewünschte konsolidierte Datenbank.
2. Doppelklicken Sie auf **Agenten** und klicken Sie auf den Agenten, mit dem Sie arbeiten möchten.
3. Im rechten Fensterausschnitt klicken Sie auf die Registerkarte **Aufgaben**.
4. Klicken Sie mit der rechten Maustaste auf die bereitgestellte Aufgabe, die Sie einleiten möchten, und klicken Sie auf **Einleiten**.

### **Ergebnisse**

Die Aufgabe wird für den ausgewählten Agenten eingeleitet.

## **Bereitgestellte entfernte Aufgaben für einen einzelnen Agenten reaktivieren**

Einige bereitgestellte entfernte Aufgaben können reaktiviert werden.

### **Voraussetzungen**

Es gibt keine Voraussetzungen für diese Aufgabe.

### **Aufgabe**

1. Doppelklicken Sie auf die konsolidierte Datenbank.
2. Doppelklicken Sie auf **Agenten** und klicken Sie auf den Agenten, mit dem Sie arbeiten möchten.
3. Im rechten Fensterausschnitt klicken Sie auf die Registerkarte **Aufgaben**.
4. Klicken Sie mit der rechten Maustaste auf die bereitgestellte Aufgabe, die Sie reaktivieren möchten, und klicken Sie auf **Reaktivieren**.

## Ergebnisse

Die bereitgestellte Aufgabe wird reaktiviert.

## Empfänger für eine bereitgestellte entfernte Aufgabe hinzufügen

Sie können einer bereitgestellten entfernten Aufgabe Empfänger hinzufügen. Beispielsweise müssten Sie einer bereitgestellten Aufgabe Agenten hinzufügen, wenn neue Agenten erstellt werden, nachdem eine Aufgabe bereitgestellt wurde.

## Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

## Aufgabe

1. Doppelklicken Sie auf das MobiLink-Projekt.
2. Klicken Sie auf **Entfernte Aufgaben » Bereitgestellte Aufgaben** und wählen Sie die bereitgestellte Aufgabe aus, der Sie Empfänger hinzufügen möchten.
3. Klicken Sie mit der rechten Maustaste auf die bereitgestellte Aufgabe, der Sie Empfänger hinzufügen möchten, und klicken Sie auf **Empfänger hinzufügen**.
4. Wählen Sie Agenten aus der Liste **Agenten** und klicken Sie auf **Hinzufügen**, um sie der Liste **Empfänger** hinzuzufügen, oder wählen Sie **Alle hinzufügen**, um alle Agenten auszuwählen.
5. Klicken Sie auf **OK**.

## Ergebnisse

Die ausgewählten Agenten werden als Empfänger hinzugefügt.

## Serverinitiierte entfernte Aufgaben (SIRT)

Eine serverinitiierte entfernte Aufgabe ist eine entfernte Aufgabe, die ausgeführt wird, wenn der Agent eine entsprechende Benachrichtigung vom Server empfängt. Wenn ein Administrator entscheidet, eine entfernte Aufgabe einzuleiten, sendet der MobiLink-Server eine Meldung an die betroffenen Agenten, die diese anweist, die angegebene Aufgabe auszuführen. Die Ausführung einer Aufgabe kann bereits für einen bestimmten Zeitpunkt geplant sein, jedoch kann die Aufgabe trotzdem vom Server eingeleitet werden, wenn der Administrator entsprechend vorgeht.

Eine SIRT kann über Sybase Central eingeleitet werden, indem entweder **Einleiten** für einen bestimmten Agenten oder **Für alle Empfänger einleiten** für eine bereitgestellte Aufgabe gewählt wird. Außerdem kann eine SIRT mit der Systemprozedur "ml\_ra\_notify\_task" auf dem MobiLink-Server eingeleitet werden.

**Siehe auch**

- „ml\_ra\_notify\_task-Systemprozedur“ auf Seite 694

**Notifier für entfernte Aufgaben (RTNotifier)**

Ein Notifier namens RTNotifier ist in den MobiLink-Server integriert, um SIRT-Anforderungen zu verfolgen. Der RTNotifier prüft die MobiLink-Systemtabellen. Wenn eine SIRT eingeleitet wurde und der MobiLink-Agent den MobiLink-Server abrufen, sendet der RTNotifier die Informationen zur entsprechenden entfernten Aufgabe an den Client und die entfernte Aufgabe wird ausgeführt.

Der RTNotifier wird standardmäßig ausgeführt. Wenn Sie nicht mit der zentralen Administration von entfernten Aufgaben arbeiten, können Sie den RTNotifier mithilfe der folgenden Optionen deaktivieren.

RTNotifier Optionen werden als Option/Wert-Paare angegeben, die in die ml\_property-Systemtabelle eingefügt werden. Das nachstehende Beispiel zeigt, wie Sie den RTNotifier deaktivieren. In diesem Beispiel lautet die RTNotifier Option **enable** und der Wert ist auf **no** gesetzt. **SIRT** ist der Komponentename und **RTNotifier(RTNotifier1)** der Notifier-Name, aber diese zwei Spalten sind nur für den internen Gebrauch gedacht und sollten nicht geändert werden.

```
call ml_add_property( 'SIRT', 'RTNotifier(RTNotifier1)', 'enable', 'yes' );
```

Die folgende Tabelle enthält die RTNotifier-Optionen, die in der ml\_property-Tabelle angegeben werden können.

| Option             | Wert                       | Beschreibung                                                                                                                                                                                                                                                                                                                                        |
|--------------------|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| autoset_poll_every | { <b>yes</b>   <b>no</b> } | Gibt an, ob die poll_every-Eigenschaft automatisch in Abhängigkeit davon angepasst werden soll, wie häufig die Agenten den MobiLink-Server nach Anforderungen entfernter Aufgaben abfragen. Wenn poll_every angepasst wird, wird die Aktualisierung bemerkt, wenn der RTNotifier nach Aktualisierungen auf Basis des update_poll_every-Werts sucht. |
| enable             | { <b>yes</b>   <b>no</b> } | Gibt an, ob der RTNotifier aktiviert werden soll, wenn der MobiLink-Server gestartet wird. Diese Eigenschaft kann nicht dynamisch aktualisiert werden. Diese Eigenschaft kann nicht dynamisch aktualisiert werden.                                                                                                                                  |
| poll_every         | <i>Zeit in Sekunden</i>    | Gibt die Häufigkeit in Sekunden an, mit der der RTNotifier request_cursor ausführt, um den In-Memory-Cache erneut mit Anforderungen entfernter Aufgaben aufzufüllen. Wenn der Wert dieser Eigenschaft 2147483647 lautet, führt der RTNotifier request_cursor nicht aus.                                                                             |
| update_poll_every  | <i>Zeit in Sekunden</i>    | Gibt die Häufigkeit in Sekunden an, mit der der RTNotifier nach Aktualisierungen der Eigenschaften suchen soll.                                                                                                                                                                                                                                     |

| Option         | Wert | Beschreibung                                                                                                                                                    |
|----------------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| request_cursor |      | Diese Option ist nur für den internen Gebrauch gedacht. Gibt eine Abfrage an, um die entfernte Aufgabeanforderungen aus der konsolidierten Datenbank abzurufen. |

## Befehle

Ein Befehl ist eine Anweisung in einer Aufgabe, die eine bestimmte Aktion ausführt. Eine Aufgabe kann mehrere Befehle enthalten und für die Befehle gilt eine festgelegte Reihenfolge. Ein Befehl enthält eine auszuführende Aktion, Eingabeparameter und Anweisungen für den Fall, dass der Befehl fehlschlägt.

### Der Befehl "Datei kopieren"

Erstellt eine Kopie einer Datei auf dem entfernten Gerät.

#### Parameter

- **Originaldateiname** Der Name der Datei, die kopiert werden soll.
- **Neuer Dateiname** Der Name der Datei, in die die ursprüngliche Datei kopiert werden soll.
- **Bestehende Datei erforderlichenfalls überschreiben** Diese Option veranlasst, dass die Datei auch dann kopiert wird, wenn bereits eine Datei mit dem unter **Neuer Dateiname** angegebenen Namen vorhanden ist.
- **Schreibschutzattribut ignorieren** Dieser Parameter kann nur verwendet werden, wenn die Option **Bestehende Datei erforderlichenfalls überschreiben** verwendet wird. Wenn diese Option verwendet wird, erfolgt der Kopiervorgang auch dann, wenn eine Datei mit dem unter **Neuer Dateiname** angegebenen Namen bereits vorhanden und schreibgeschützt ist.

#### Bemerkungen

Dateinamen können absolut oder relativ angegeben werden. Wenn ein relativer Dateiname angegeben wird, gilt er als relativ zum aktuellen Arbeitsverzeichnis des Agenten.

### Der Befehl "Datenbank erstellen"

Erstellt eine neue entfernte Datenbank auf dem entfernten Gerät, das von dem Agenten verwaltet wird.

#### Parameter

- **Dateiname** Der Dateiname für die neue Datenbank. In der Regel sollten Sie das Makro {db\_location} verwenden, um den Pfad für Ihre Datenbank anzugeben. Beispiel: {db\_location} \MeineDatenbank.db. Wenn ein Agent mehr als eine SQL Anywhere-Datenbank verwalten soll, müssen Sie jede Datenbank in einem separaten Verzeichnis erstellen. Es wird empfohlen, dass Sie jede Datenbank in ein eigenes Unterverzeichnis des {db\_location}-Verzeichnisses legen.

- **CHAR-Kollation** Gibt die Kollation für die Datentypen CHAR, VARCHAR und LONG VARCHAR in der neuen Datenbank an.
- **NCHAR-Kollation** Gibt nur bei SQL Anywhere die Kollation für die Datentypen NCHAR, NVARCHAR und LONG NVARCHAR in der neuen Datenbank an.

### Bemerkungen

Dieser Befehl kann nur in einer entfernten Aufgabe verwendet werden, die als **benötigt oder erstellt eine entfernte Datenbank** markiert ist.

Der Typ der Datenbank (UltraLite oder SQL Anywhere), die durch den Befehl erstellt wird, ist abhängig von dem für die entfernte Aufgabe angegebenen Namen des entfernten Schemas. Wenn der angegebene Dateiname ein Verzeichnis verwendet, das auf dem entfernten Gerät nicht vorhanden ist, wird das Verzeichnis erstellt. Der Dateiname kann absolut oder relativ angegeben werden. Wenn ein relativer Dateiname angegeben wird, gilt er als relativ zum aktuellen Arbeitsverzeichnis des Agenten. Dies wird nicht für SQL Anywhere in Windows Mobile unterstützt.

Mit der CREATE DATABASE-Anweisung können Sie eine Datenbank auf einem PC initialisieren und sie später auf ein Windows Mobile-Gerät kopieren.

## Der Befehl "Datei löschen"

Löscht eine Datei auf dem entfernten Gerät.

### Parameter

- **Dateiname** Der Name der Datei auf dem entfernten Gerät, die gelöscht werden soll.
- **Schreibschutzattribut ignorieren** Das Aktivieren dieser Option führt dazu, dass die Datei auch dann gelöscht wird, wenn sie als schreibgeschützt markiert ist.

### Bemerkungen

Der Dateiname kann absolut oder relativ angegeben werden. Wenn ein relativer Dateiname angegeben wird, gilt er als relativ zum aktuellen Arbeitsverzeichnis des Agenten.

Standardmäßig schlägt eine Aufgabe fehl, wenn die zu löschende Datei nicht vorhanden ist. Damit eine Aufgabe erfolgreich ausgeführt werden kann, auch wenn die zu löschende Datei nicht vorhanden ist, deaktivieren Sie im MobiLink 16-Plug-In für Sybase Central die Option **Fehlschlag, wenn die Datei nicht existiert** auf der Eigenschaftenseite **Befehle** für den Befehl "Datei löschen".

## Der Befehl "Datei herunterladen"

Lädt eine Datei vom Server auf das entfernte Gerät herunter.

### Parameter

- **Serverdateiname** Der Name der Datei auf dem Server, die auf das entfernte Gerät heruntergeladen werden soll. Der Dateiname kann nicht absolut angegeben werden. Er gilt als relativ

zum Download-Stammverzeichnis des MobiLink-Servers. Dieses Verzeichnis wird unter Verwendung der Option -ftr auf dem MobiLink-Server angegeben. Siehe „[mlsrv16-Option -ftr](#)“ auf Seite 63.

- **Name der entfernten Datei** Gibt den Speicherort auf dem entfernten Gerät an, unter dem die Datei gespeichert werden soll. Der Dateiname kann absolut oder relativ angegeben werden. Wenn ein relativer Dateiname angegeben wird, gilt er als relativ zum aktuellen Arbeitsverzeichnis des Agenten.

## Der Befehl "Datenbank löschen"

Löscht eine verwaltete entfernte Datenbank.

### Parameter

Keine

### Bemerkungen

Dieser Befehl kann nur in einer entfernten Aufgabe verwendet werden, die als "benötigt oder erstellt eine entfernte Datenbank" markiert ist. Die gelöschte Datenbank ist diejenige, die mit dem Namen des für die entfernte Aufgabe angegebenen Schemas verknüpft ist.

Die Verbindungszeichenfolge für den Namen des entfernten Schemas, das der Aufgabe mit dem Befehl "Datenbank löschen" zugeordnet ist, muss einen DBF-Parameter enthalten.

Die zu löschende Datenbank darf nicht laufen, wenn der Befehl "Datenbank löschen" ausgeführt wird.

Alle Daten in der entfernten Datenbank gehen verloren, wenn die Datenbank gelöscht wird.

Das Löschen einer Datenbank wird unter Windows Mobile nicht unterstützt.

### Siehe auch

- „UltraLite-Verbindungsparameter DBF“ [[UltraLite - Datenbankverwaltung](#)]
- „Verbindungsparameter DatabaseFile (DBF)“ [[SQL Anywhere Server - Datenbankadministration](#)]

## Der Befehl "SQL ausführen"

Führt SQL-Befehle in einer entfernten Datenbank aus.

### Parameter

- **SQL** Die auszuführende SQL-Anweisung. Dabei müssen Sie die einzelnen SQL-Anweisungen mit GO in einer eigenen Zeile trennen. Bei SQL-Anweisungen, die von einer BEGIN- und END-Anweisung umschlossen werden, dürfen Sie innerhalb des BEGIN-END-Blocks kein GO angeben. Das folgende Beispiel zeigt die richtige Verwendung von GO zum Trennen von Anweisungen:

```
SELECT * FROM systable
GO
CREATE PROCEDURE p1()
BEGIN
    CREATE TABLE t1( pk INTEGER PRIMARY KEY );
    INSERT INTO t1 VALUES( 5 );
    COMMIT;
```

```
END
GO
SELECT * FROM sysprocedure
```

## Bemerkungen

Dieser Befehl kann nur in einer entfernten Aufgabe verwendet werden, die als "benötigt oder erstellt eine entfernte Datenbank" markiert ist. Die SQL-Anweisung wird in der Datenbank ausgeführt, die mit dem Namen des für die entfernte Aufgabe angegebenen Schemas verknüpft ist.

Beachten Sie bei der Ausführung von SQL, dass der Agent kein COMMIT von Anweisungen ausführt. Wenn die ausgeführte SQL-Anweisung kein COMMIT enthält, wird die Anweisung zurückgesetzt. Dies ist wichtig, wenn es sich um die SQL-Anweisungen INSERT, UPDATE und DELETE oder um andere Anweisungen handelt, die nicht explizit ein COMMIT bewirken.

Der Status bzw. die Ergebnisse für den Befehl spiegeln die Ergebnisse von der ausgeführten SQL-Anweisung wider. DDL-Anweisungen geben keine Ergebnisse zurück. INSERT/UPDATE/DELETE-Anweisungen geben die Anzahl der betroffenen Zeilen als einzelnen Wert in einer Zeile zurück. SELECT-Anweisungen geben die Ergebnisse im .csv-Format mit Spaltentiteln als erste Zeile zurück. Die Ergebnisse mehrerer Anweisungen werden in einem großen Ergebnis aneinandergehängt.

## Der Befehl "Eingabeaufforderung"

Zeigt ein Meldungsfeld auf dem entfernten Gerät an.

### Parameter

- **Meldung** Die in dem Meldungsfeld anzuzeigende Meldung.

### Bemerkungen

Die Meldung auf dem entfernten Gerät ist so lange sichtbar, bis sie durch Klicken auf **OK** geschlossen wird.

Wenn die Aufgabe weitere Befehle enthält, die auf den Befehl "Eingabeaufforderung" folgen, werden diese erst nach dem Schließen der Meldung ausgeführt. Der Zeitraum zwischen dem Anzeigen der Eingabeaufforderung und dem Klicken auf **OK** wird nicht in die Berechnung der Laufzeit für die Aufgabe einbezogen.

## Der Befehl "Datei umbenennen"

Benennt eine Datei auf dem entfernten Gerät um.

### Parameter

- **Originaldateiname** Der aktuelle Name der umzubenennenden Datei.
- **Neuer Dateiname** Der Name der Datei, nachdem sie umbenannt wurde.
- **Bestehende Datei erforderlichenfalls überschreiben** Das Aktivieren dieser Option führt dazu, dass die Datei auch dann umbenannt wird, wenn eine Datei mit dem neuen Dateinamen bereits vorhanden ist.

- **Schreibschutzattribut ignorieren** Diese Option kann nur ausgewählt werden, wenn **Bestehende Datei** ausgewählt ist. Wenn diese Option ausgewählt ist, wird die Umbenennung auch dann durchgeführt, wenn eine Datei mit dem unter "Neuer Dateiname" angegebenen Namen bereits vorhanden und schreibgeschützt ist.

### Bemerkungen

Dateinamen können absolut oder relativ angegeben werden. Wenn ein relativer Dateiname angegeben wird, gilt er als relativ zum aktuellen Arbeitsverzeichnis des Agenten.

## Der Befehl "Programm ausführen"

Führt ein Programm auf dem entfernten Gerät aus.

### Parameter

- **Befehlszeile** Die Befehlszeile, die ausgeführt werden soll.

### Bemerkungen

Die Ausführung der Aufgabe wird erst fortgesetzt, wenn die Ausführung des Programms abgeschlossen ist. Der Befehl gilt als erfolgreich, wenn der Beendigungscode für das ausgeführte Programm 0 ist.

## Der Befehl "Synchronisieren"

Synchronisiert eine entfernte Datenbank.

### Parameter

- **Synchronisationsprofil** Gibt ein bereits in der entfernten Datenbank definiertes Synchronisationsprofil an, das die Optionen enthält, die für die Synchronisation verwendet werden sollen.
- **Zusatzoptionen** Gibt zusätzliche Optionen an, die für die Synchronisation verwendet werden sollen. Wenn sowohl bei den Zusatzoptionen als auch im Synchronisationsprofil eine Option angegeben ist, hat die Einstellung aus den Zusatzoptionen Vorrang vor der Einstellung im Synchronisationsprofil. Diese Option kann leer gelassen werden.

### Bemerkungen

Dieser Befehl kann nur in einer entfernten Aufgabe verwendet werden, die als "benötigt oder erstellt eine entfernte Datenbank" markiert ist. Die synchronisierte Datenbank ist diejenige, die mit dem Namen des für die entfernte Aufgabe angegebenen Schemas verknüpft ist.

## Der Befehl "Datei hochladen"

Lädt eine Datei vom entfernten Gerät auf den Server hoch.

## Parameter

- **Name der entfernten Datei** Der Name der Datei auf dem entfernten Gerät, die auf den Server hochgeladen werden soll. Dieser Dateiname kann absolut oder relativ angegeben werden. Wenn ein relativer Dateiname angegeben wird, gilt er als relativ zum aktuellen Arbeitsverzeichnis des Agenten.
- **Serverdateiname** Gibt den Speicherort auf dem Server an, unter dem die Datei gespeichert werden soll. Dieser Dateiname kann nicht absolut angegeben werden und darf nicht mehr als einen Backslash ("\") enthalten. Er gilt als relativ zum Upload-Stammverzeichnis des MobiLink-Servers. Dieses Verzeichnis wird unter Verwendung der Option -ftru auf dem MobiLink-Server angegeben.

Es empfiehlt sich, im Serverdateinamen ein Makro zu verwenden, um sicherzustellen, dass jeder Agent, der den Befehl ausführt, seine Datei an einen anderen Speicherort auf dem Server hochlädt. Andernfalls haben Sie möglicherweise Probleme, weil Agenten ihre Dateien gegenseitig überschreiben. Eine gute Konvention besteht darin, die Dateien von den einzelnen Agent in einem separaten Verzeichnis zu speichern, wobei der Name des Verzeichnisses der Agenten-ID entspricht. Sie können das Makro {agent\_id} verwenden, um dies zu erreichen. Wenn Sie beispielsweise eine Datei namens *myuploadfile.txt* hochladen möchten, können Sie den Namen der Zieldatei auf {agent\_id}\myuploadfile.txt setzen. Siehe „[Variablen in Parametern](#)“ auf Seite 187.

## Befehlsverwendung

Verwenden Sie eine der folgenden Methoden, einen Befehl zu einer Aufgabe hinzuzufügen:

- Klicken Sie in der Ansicht **Ordner** im linken Fensterausschnitt mit der rechten Maustaste auf die Aufgabe und klicken Sie auf **Befehl hinzufügen**.
- Wählen Sie in der Ansicht **Ordner** im linken Fensterausschnitt die Aufgabe und klicken Sie in der Symbolleiste auf **Befehl hinzufügen**.
- Beim Erstellen einer Aufgabe erhalten Sie automatisch einen Befehl im rechten Fensterausschnitt. Drücken Sie die Tabulatortaste, um von Parameter zu Parameter zu wechseln. Wenn Sie die Tabulatortaste gedrückt halten, wird der Aufgabe automatisch ein neuer Befehl hinzugefügt.
- Klicken Sie mit der rechten Maustaste auf einen vorhandenen Befehl und klicken Sie auf **Befehl hinzufügen**.
- Klicken Sie mit der rechten Maustaste auf den leeren Bereich unter den vorhandenen Befehlen und klicken Sie auf **Befehl hinzufügen**.
- Wählen Sie die Aufgabe in der Ansicht **Ordner** im linken Fensterausschnitt. Klicken Sie im Menü **Datei** auf die Option **Befehl hinzufügen**.

Siehe „[Befehle zu entfernten Aufgaben hinzufügen](#)“ auf Seite 186.

## Befehle zu entfernten Aufgaben hinzufügen

Das Deployment einer entfernten Aufgabe kann erst durchgeführt werden, wenn sie zumindest einen Befehl enthält.

## Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

## Aufgabe

1. Doppelklicken Sie auf das MobiLink-Projekt.
2. Doppelklicken Sie auf **Entfernte Aufgaben**, rechtsklicken Sie auf die entfernte Aufgabe, mit der Sie arbeiten möchten, und klicken Sie auf **Befehl hinzufügen**. Der Bereich **Befehle** erscheint im rechten Fensterausschnitt.
3. Wählen Sie aus der Dropdown-Liste **Befehlstyp** den Typ des benötigten Befehls. Weitere Hinweise zu den verfügbaren Einstellungen finden Sie unter [„Befehle“ auf Seite 181](#).
4. Geben Sie die geeigneten Parameter für den ausgewählten Befehl ein.
5. Wählen Sie aus der Dropdown-Liste **Bei Fehler** eine der folgenden Optionen, um anzugeben, wie vorgegangen werden soll, wenn der Befehl fehlschlägt:
  - **Aufgabe abbrechen** Der aktuelle Versuch zum Ausführen der Aufgabe wird beendet und als fehlgeschlagen markiert.
  - **Fortsetzen** Die Ausführung der Aufgabe wird mit dem nächsten Befehl fortgesetzt.
  - **Befehl wiederholen** Die Aufgabe wird erneut versucht, beginnend beim fehlgeschlagenen Befehl. Wenn die maximale Anzahl der Versuche für die Aufgabe erreicht ist, wird der Befehl nicht erneut versucht.
  - **Aufgabe neu starten** Die Aufgabe wird erneut versucht, beginnend mit dem ersten Befehl. Wenn die maximale Anzahl der Versuche für die Aufgabe erreicht ist, wird die Aufgabe nicht erneut versucht.

## Ergebnisse

Ergebnis

## Nächste Schritte

Die Aufgabe kann bereitgestellt werden. Siehe [„Entfernte Aufgaben bereitstellen“ auf Seite 174](#).

## Variablen in Parametern

Die folgenden Makros bieten Zugriff auf Daten, die zwischen entfernten Geräten variieren können. Diese Werte können verwendet werden in Bedingungen für entfernte Aufgaben, in Parametern für Befehle in entfernten Aufgaben und in Verbindungszeichenfolgen:

| Variable            | Ersetzung                                                                                                                                                                                                                                                                                  |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {agent_db}          | Der vollständige Pfad und Dateiname für die Agenten-Datenbankdatei. Diese Datei kann bei Bedarf aus dem Gerät hochgeladen werden, um die Problem diagnosis zu unterstützen.                                                                                                                |
| {agent_id}          | Die Agenten-ID.                                                                                                                                                                                                                                                                            |
| {agent_log}         | Der vollständige Pfad und Dateiname für die Agenten-Logdatei. Diese Datei kann bei Bedarf aus dem Gerät hochgeladen werden, um die Problem diagnosis zu unterstützen. Wenn der Agent ohne Logdatei ausgeführt wird, ist diese Variable eine leere Zeichenfolge.                            |
| {battery_level}     | Der Akkustand für das entfernte Gerät. Der Bereich ist null bis einhundert (0-100).                                                                                                                                                                                                        |
| {db_location}       | Das Verzeichnis für die entfernte Datenbank des Agenten, wie durch die Option <code>mlagent -db</code> angegeben.                                                                                                                                                                          |
| {is_on_ac_power}    | Zeigt an, ob das entfernte Gerät an das Stromnetz angeschlossen ist. 1 bedeutet, dass das Gerät an das Stromnetz angeschlossen ist, und 0 bedeutet, dass das Gerät im Akkubetrieb läuft.                                                                                                   |
| {is_online}         | Diese Variable wird nur dann als 1 (TRUE) ausgewertet, wenn das Clientgerät mit einem Netzwerk verbunden, sodass eine Route zur IP-Adresse des MobiLink-Servers vorhanden ist. Wenn der Hostcomputer offline ist, wird die Variable dennoch als 1 (TRUE) ausgewertet.                      |
| {ml_password}       | Das MobiLink-Kennwort (pwd), das vom Agenten für die Synchronisation mit der Agenten-Datenbank verwendet wird.                                                                                                                                                                             |
| {ml_stream}         | Die Protokollparameter für die Verbindung mit dem MobiLink-Server. Zum Beispiel: <code>HTTP{host=Sybase.com;port=9376}</code>                                                                                                                                                              |
| {ml_username}       | Der MobiLink-Benutzername (uid), der vom Agenten für die Synchronisation mit der Agenten-Datenbank verwendet wird.                                                                                                                                                                         |
| {network_conn_name} | Gibt den Namen der Netzwerkverbindung an, die vom Agenten für die Kommunikation mit dem MobiLink-Server verwendet wird. Wenn es keine Netzwerkverbindung gibt, die vom Agenten für die Kommunikation mit dem MobiLink-Server verwendet werden kann, wird diese Variable als ? ausgewertet. |
| {remote_id}         | Die entfernte ID für die entfernte Datenbank, die dieser Aufgabe zugeordnet ist. Dieser Wert ist nur in entfernten Aufgaben relevant, die als <b>Diese Aufgabe benötigt oder erstellt eine entfernte Datenbank</b> markiert sind.                                                          |

| Variable         | Ersetzung                                                                                                                                                           |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {rows_to_upload} | Nur für UltraLite-Datenbanken. Die Anzahl der Zeilen in der entfernten Datenbank, die hochgeladen werden, wenn eine vollständige Synchronisation durchgeführt wird. |

## Status

Wenn der Agent eine Aufgabe ausführt, werden Statusinformationen über diese Ausführung in der Agenten-Datenbank gespeichert, es sei denn, die Aufgabe ist so markiert, dass keine Statusinformationen gemeldet werden. Statusinformationen in der Agenten-Datenbank werden auf den Server hochgeladen, wenn die Agenten-Datenbank synchronisiert wird.

Die Statusinformationen sind unter Verwendung des MobiLink 16-Plug-Ins für Sybase Central zugänglich. Um Statusinformationen für eine bestimmte bereitgestellte Aufgabe anzuzeigen, wählen Sie die Aufgabe in der Ansicht **Ordner** im linken Fensterausschnitt aus und zeigen Sie im rechten Fensterausschnitt die Registerkarte **Empfänger** oder **Ergebnisse** an. Um Statusinformationen für eine bestimmte bereitgestellte Aufgabe anzuzeigen, wählen Sie die Aufgabe in der Ansicht **Ordner** im linken Fensterausschnitt aus und zeigen Sie im rechten Fensterausschnitt die Registerkarte **Aufgaben** an.

Die folgenden gespeicherten Prozeduren geben ebenfalls Statusinformationen zurück:

- „ml\_ra\_get\_agent\_events-Systemprozedur“
- „ml\_ra\_get\_agent\_ids-Systemprozedur“
- „ml\_ra\_get\_agent\_properties-Systemprozedur“
- „ml\_ra\_get\_latest\_event\_id-Systemprozedur“
- „ml\_ra\_get\_orphan\_taskdbs-Systemprozedur“
- „ml\_ra\_get\_remote\_ids-Systemprozedur“
- „ml\_ra\_get\_task\_results-Systemprozedur“
- „ml\_ra\_get\_task\_status-Systemprozedur“

## Systemprozeduren

Zusätzlich zu den Funktionen für die Verwaltung von entfernten Aufgaben in Sybase Central gibt es auch MobiLink-Systemprozeduren in der konsolidierten Datenbank, die verwendet werden können, um Verwaltungsaufgaben zu automatisieren. Mit Ausnahme der Systemprozedur ml\_ra\_cancel\_notification und der Fehlerbehebungsprozeduren können alle Vorgänge, die mit Systemprozeduren durchgeführt werden können, auch unter Verwendung des MobiLink 16-Plug-Ins für Sybase Central durchgeführt werden. Die folgenden Aufgaben können jedoch nur unter Verwendung der MobiLink 16-Plug-Ins für Sybase Central durchgeführt werden:

- Neue Aufgaben erstellen
- Namen für ein entferntes Schema erstellen
- Beschreibungen zu Agenten, entfernten Datenbanken, Namen für entfernte Schemata und Aufgaben hinzufügen

Alle neuen MobiLink-Systemtabellen und -Systemprozeduren sowie die Agenten-Skriptversion beginnen mit dem Präfix **ml\_ra\_**.

Im Folgenden finden Sie eine Liste der Systemprozeduren, die bei der zentralen Administration von entfernten Aufgaben verwendet werden:

- „ml\_ra\_add\_agent\_id-Systemprozedur“
- „ml\_ra\_assign\_task-Systemprozedur“
- „ml\_ra\_cancel\_notification-Systemprozedur“
- „ml\_ra\_cancel\_task\_instance-Systemprozedur“
- „ml\_ra\_clone\_agent\_properties-Systemprozedur“
- „ml\_ra\_delete\_agent\_id-Systemprozedur“
- „ml\_ra\_delete\_events\_before-Systemprozedur“
- „ml\_ra\_delete\_remote\_id-Systemprozedur“
- „ml\_ra\_delete\_task-Systemprozedur“
- „ml\_ra\_get\_agent\_events-Systemprozedur“
- „ml\_ra\_get\_agent\_ids-Systemprozedur“
- „ml\_ra\_get\_agent\_properties-Systemprozedur“
- „ml\_ra\_get\_latest\_event\_id-Systemprozedur“
- „ml\_ra\_get\_orphan\_taskdbs-Systemprozedur“
- „ml\_ra\_get\_remote\_ids-Systemprozedur“
- „ml\_ra\_get\_task\_results-Systemprozedur“
- „ml\_ra\_get\_task\_status-Systemprozedur“
- „ml\_ra\_manage\_remote\_db-Systemprozedur“
- „ml\_ra\_notify\_agent\_sync-Systemprozedur“
- „ml\_ra\_reassign\_taskdb-Systemprozedur“
- „ml\_ra\_set\_agent\_property-Systemprozedur“
- „ml\_ra\_unmanage\_remote\_db-Systemprozedur“

## Deployment und Konfiguration

Der folgende Abschnitt enthält Informationen über das Deployment und die Konfiguration. Eine Liste der für das Deployment des MobiLink-Agenten erforderlichen Dateien finden Sie unter [„Bereitstellung von SQL Anywhere MobiLink-Clients“](#) auf Seite 782 und [„Deployment des MobiLink-Clients für UltraLite“](#) auf Seite 785.

### Hinweise zum Deployment auf Agentenseite

Damit die entfernten Datenbanken in Ihrem MobiLink-Synchronisationssystem richtig verwaltet werden, müssen Sie einige technische Aspekte beachten:

- Der MobiLink-Agent läuft nur auf Windows- und Windows Mobile-Geräten. Entfernte Datenbanken auf anderen Plattformen als diesen können zurzeit nicht über einen MobiLink-Agenten verwaltet werden.
- Zum Verwalten von entfernten UltraLite-Datenbanken erfordert die Verwendung der UltraLite-Engine, damit Anwendungen, die auf die entfernten Datenbanken zugreifen, dies ebenfalls über die UltraLite-Engine tun müssen. Der Versuch, die prozessintegrierte Version von UltraLite zu verwenden, führt zu

Fehlermeldungen, wonach die Datei bereits verwendet wird. Siehe „[UltraLite-Datenverwaltungskomponenten für Windows Mobile](#)“ [[UltraLite - Datenbankverwaltung](#)].

- Zentrale Administration ist nur möglich, wenn der MobiLink-Agent auf einem Gerät läuft. Im Allgemeinen wird angenommen, dass der Agent auf einem Gerät immer läuft. Sie können einen Agenten unter Verwendung der Datei *mlastop.exe* stoppen. Dann müssen Sie jedoch den Agenten neu starten, damit die zentrale Administration wieder wirksam wird. Siehe „[Dienstprogramm zum MobiLink-Agent-Stopp](#)“ auf Seite 163.

## Deployment von UltraLite-Anwendungen und -Datenbanken mit dem MobiLink-Agenten unter Windows Mobile

Es gibt verschiedene Verfahren für das Deployment von UltraLite mit dem Agenten auf einem Windows Mobile-Gerät.

Sie können mit dem SQL Anywhere-**Deployment-Assistenten** eine Cabinet-Datei (.CAB) erstellen, die verwendet werden kann, um das Deployment von SQL Anywhere unter Windows Mobile durchzuführen. Der **Deployment-Assistent** umfasst jedoch keine Unterstützung für das Erstellen von Deployments von Benutzeranwendungen und -datenbanken.

Bei Abschluss des **Deployment-Assistenten** wird eine .INF-Datei erstellt. Die .INF-Datei besteht aus einer Reihe von Abschnitten mit Beschreibungen des Zielspeicherorts für die Dateien, Verknüpfungen und Registrierungseinstellungen, die in der .CAB-Datei enthalten sind. Diese .INF-Datei kann so geändert werden, dass sie auch die Logik enthält, durch die Benutzeranwendungen und -datenbanken mit dem Agenten installiert werden.

## Deployment mit SQL Anywhere für Windows Mobile - Deployment-Assistent

Sie können mit **SQL Anywhere für Windows Mobile - Deployment-Assistent** ein Deployment der für die zentrale Administration von entfernten Datenbanken erforderlichen Dateien durchführen.

**SQL Anywhere für Windows Mobile - Deployment-Assistent** umfasst die Option **Entfernte SQL Anywhere-Datenbanken verwalten** oder **Entfernte UltraLite-Datenbanken verwalten**.

## Konfiguration des Agenten unter Windows

Eine Möglichkeit für die Konfiguration des Agenten auf einem Windows-Computer besteht darin, ein Installationsprogramm (dasselbe wie bei der Installation Ihrer Anwendung) für den Agenten zu verwenden und die Befehle zum Konfigurieren, Validieren und Starten des Agenten auszuführen. Ein Installationsprogramm fordert Sie möglicherweise auf, Identifizierungs- und/oder Authentifizierungsparameter für MobiLink einzugeben. Diese Parameter können dann verwendet werden, um den Agenten mithilfe der Option `mlagent -cr` in der Befehlszeile zu konfigurieren.

Nachdem der Agent konfiguriert ist, könnte das Installationsprogramm den Befehl **mlagent -pi** ... ausführen, um zu überprüfen, ob die Authentifizierungsparameter gültig sind. Eine Verbindung zum MobiLink-Server ist erforderlich, wenn dieser Befehl ausgeführt wird, damit die Authentifizierung richtig validiert wird.

Schließlich könnte der Agent als letzter Schritt im Installationsprozess gestartet werden. Der Agent ist dann bereit, auf dem Zielgerät Aufgaben zu empfangen und auszuführen. Der Prozessrückgabecode des Agenten kann verwendet werden, um Informationen zur Ausführung des Agenten bereitzustellen.

Beispiel: Wenn das Ping von mlagent an den MobiLink-Server fehlgeschlagen ist, entspricht der Rückgabecode aus *mlagent.exe* dem SQLCODE, der sich aus einer Synchronisation der Agentendatenbank mit den konfigurierten mlagent-Optionen ergibt.

### Konfiguration des Agenten unter Windows Mobile

Wie Sie den MobiLink Agent unter Windows Mobile konfigurieren, hängt davon ab, wie die Anwendung des Benutzers auf dem Gerät installiert wird.

Eine Möglichkeit, den MobiLink-Agenten zu konfigurieren und auszuführen, besteht darin, dass die Benutzeranwendung den Benutzer zur Eingabe der Identifizierungs- und/oder Authentifizierungsparameter für MobiLink auffordert und dann den MobiLink-Agenten als Teil ihres Startvorgangs konfiguriert und/oder startet.

- Die Anwendung kann versuchen, den Agenten auszuführen. Wenn sie nicht richtig konfiguriert wurde, gibt das mlagent-Programm einen Fehlercode zurück.
- Wenn Sie diesen Code erhalten, führen Sie den Agenten im Konfigurationsmodus aus, um ihn einzurichten, und versuchen Sie anschließend erneut, den Agenten im Normalmodus auszuführen. Siehe „[MobiLink-Agent auf dem Clientgerät](#)“ auf Seite 158.

## MobiLink-Profiler

Der MobiLink-Profiler ist ein MobiLink-Administrationstool, das Ihnen detaillierte Informationen zur Performance Ihrer Synchronisationen liefert, mit deren Hilfe Sie Engpässe analysieren und die Performance maximieren können. Sie können den SQL Anywhere-Monitor verwenden, um grundlegende Performanceinformationen abzurufen, und den MobiLink-Profiler für tiefergehende Details zu Synchronisationen, bis hinunter auf die Ereignisebene.

Synchronisationsdaten aus Ihrer Profilerstellungssitzung werden in einer entsprechenden Datenbankdatei gespeichert, die in Ihrem Datenverzeichnis mit Standardwerten für Dateiname, Benutzer und Kennwort erstellt wird. Auf der Registerkarte **Allgemein** des Fensters **Optionen** können Sie eine andere Profilerstellungsdatenbank angeben.

Wenn Sie den MobiLink-Profiler starten und mit einem MobiLink-Server verbinden, beginnt der MobiLink-Profiler, statistische Informationen zu allen Synchronisationen zu sammeln, die in dieser Profilerstellungssitzung stattfinden. Der MobiLink-Profiler sammelt weiterhin Daten, bis Sie die Profilerstellungssitzung beenden oder den MobiLink-Server herunterfahren. Sie können die Daten über die MobiLink-Profiler-Benutzeroberfläche in tabellarischer oder grafischer Form anzeigen.

Die MobiLink-Profiler-Ausgabe bietet eine Vielzahl von Informationen zu Ihren Synchronisationen. Sie können beispielsweise schnell Synchronisationen oder Ereignisse identifizieren, die zu Fehlern führen oder andere von Ihnen festgelegte Kriterien erfüllen. Sie können mögliche Konflikte in Synchronisationsskripten identifizieren, indem Sie überprüfen, ob Synchronisationen unterschiedlicher Dauer Phasen haben, die ungefähr zur selben Zeit enden (weil Synchronisationen vor dem Fortfahren warten, bis die vorherige Phase abgeschlossen ist). Außerdem können Sie Ereignisse identifizieren, für die der MobiLink-Server eine Blockierung erkannt hat.

Es wird empfohlen, dass Sie den Profiler hauptsächlich in einer Entwicklungsumgebung einsetzen, um die Performance zu testen, bevor Sie ein Deployment in einem Produktionssystem durchführen.

### SQL Anywhere-Monitor

Der SQL Anywhere-Monitor ist ein browserbasiertes Administrationstool, das Informationen über den Zustand und die Verfügbarkeit von SQL Anywhere-Datenbanken und MobiLink-Servern bereitstellt. Der Monitor hilft bei der Beurteilung des allgemeinen Zustands und der Verfügbarkeit des Systems sowie bei der Analyse von allgemeinen Synchronisationsstatistiken. Der SQL Anywhere-Monitor stellt keine Informationen über einzelne Synchronisationen bereit. Detaillierte Informationen zu einzelnen Synchronisationen, einschließlich Timing- und anderer Synchronisationsstatistiken, erhalten Sie mithilfe des MobiLink-Profilers.

Weitere Hinweise zum SQL Anywhere-Monitor finden Sie unter „[SQL Anywhere-Monitor](#)“ [[SQL Anywhere Server - Datenbankadministration](#)].

## MobiLink-Profiler starten (Administrationstools)

Sie können für jeden MobiLink-Server mehrere Instanzen des MobiLink-Profilers ausführen. Es wird jedoch empfohlen, nur eine MobiLink-Profiler-Instanz pro MobiLink-Server auszuführen.

### Voraussetzungen

Starten Sie für neue Profilerstellungssitzungen Ihre konsolidierte Datenbank und den MobiLink-Server, sofern sie noch nicht laufen.

### Kontext und Bemerkungen

#### Hinweis

Die Version des MobiLink-Profilers muss mit der Version des MobiLink-Servers übereinstimmen, den Sie verwenden.

### Aufgabe

- Klicken Sie auf **Start » Programme » SQL Anywhere 16 » Administrationstools » MobiLink-Profiler**.

### Ergebnisse

Der MobiLink-Profiler wird gestartet.

### Nächste Schritte

Beginnen Sie eine Profilerstellungssitzung, um das Sammeln von Daten zu starten. Siehe „[Profilerstellungssitzungen starten](#)“ auf Seite 194.

# MobiLink-Profiler (mlprof) in der Befehlszeile

Mithilfe von Befehlszeilenoptionen können Sie veranlassen, dass der MobiLink-Profiler beim Start geöffnet wird und eine Verbindung mit einem MobiLink-Server herstellt. Dies ist nützlich, um während einer Testsitzung eine automatische Profilerstellung ohne Bedienereingriff zu gewährleisten.

Verwenden Sie die folgende Syntax:

mlprof [ options ]

| Option                                           | Beschreibung                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -c                                               | Schließt den MobiLink-Profiler am Ende der Profilerstellungssitzung.                                                                                                                                                                                                                                                                             |
| -p password                                      | Das Kennwort für den MobiLink-Benutzer.                                                                                                                                                                                                                                                                                                          |
| -r                                               | Erstellt die Profilerstellungsdatenbank neu.<br><br>Verwenden Sie diese Option, wenn Sie alle früheren Profilerstellungssitzungen entfernen möchten oder wenn es ein Problem mit dem Datenbankschema für die Profilerstellung gibt                                                                                                               |
| -u ml_username                                   | Der MobiLink-Benutzer. Diese Option ist erforderlich, um eine über die Befehlszeile initiierte Profilerstellungssitzung zu beginnen.                                                                                                                                                                                                             |
| -x {tcpip tls http https}[(key-word=value;... )] | Das Netzwerkprotokoll und die Parameter für die Verbindung mit dem MobiLink-Server. Die Schlüsselwort=Wert-Paare können den Host, das Protokoll und zusätzliche Netzwerkparameter angeben. Diese Option ist erforderlich, um eine über die Befehlszeile initiierte Profilerstellungssitzung zu beginnen. Siehe „mlsrv16-Option -x“ auf Seite 93. |

Sie können **mlprof -?** eingeben, um die mlprof-Syntax anzuzeigen.

## Profilerstellungssitzungen starten

Mit dem Starten einer MobiLink-Profiler-Sitzung beginnt die Sammlung von Daten und die Daten werden in einer Profilerstellungsdatenbank gespeichert.

### Voraussetzungen

Starten Sie Ihre konsolidierte Datenbank und den MobiLink-Server, sofern sie noch nicht laufen.

### Aufgabe

1. Klicken Sie im MobiLink-Profiler auf **Datei » Profiler-Sitzung beginnen**. Damit wird die Sammlung von Daten gestartet.

2. Eine MobiLink-Profiler-Verbindung wird genauso gestartet wie eine Synchronisationsverbindung mit dem MobiLink-Server. Für alle MobiLink-Profiler-Sitzungen wird die Skriptversion auf **for\_ML\_Monitor\_only** gesetzt.

Das Dialogfeld **Mit dem MobiLink-Server verbinden** muss wie folgt ausgefüllt werden.

- **User** Geben Sie hier den Namen des MobiLink-Benutzers für die Verbindung ein. Ein Benutzername muss angegeben werden, aber wenn Sie den MobiLink-Server mit -zu+ gestartet haben, spielt es keine Rolle, welchen MobiLink-Benutzer Sie angeben, weil nicht erkannte MobiLink-Benutzernamen bei der Synchronisation automatisch zur Tabelle ml\_user hinzugefügt werden.
- **Kennwort** Geben Sie ein Kennwort für die Verbindung ein. Das angegebene Kennwort muss das Kennwort des vorher angegebenen MobiLink-Benutzers sein. Lassen Sie dieses Feld leer, wenn der MobiLink-Benutzer kein Kennwort hat.
- **Host** Der Netzwerkname oder die IP-Adresse des Computers, auf dem der MobiLink-Server läuft. Standardmäßig ist der Host der Computer, auf dem der MobiLink-Profiler läuft. Sie können **localhost** verwenden, wenn der MobiLink-Server auf demselben Computer läuft wie der MobiLink-Profiler.
- **Protokoll** Hier muss das Netzwerkprotokoll angegeben werden, das auch vom MobiLink-Server für Synchronisationsanforderungen verwendet wird.
- **Port** Hier muss der Netzwerkport angegeben werden, der auch vom MobiLink-Server für Synchronisationsanforderungen verwendet wird.
- **Verschlüsselung** Wenn Sie ein HTTPS- oder TLS-Protokoll auswählen, ist diese Option aktiviert. Wählen Sie einen Verschlüsselungstyp aus der Dropdown-Liste.

Damit HTTPS und TLS verwendet werden können, muss die clientseitige MobiLink-Datenstromverschlüsselung auf dem Computer installiert sein, auf dem der MobiLink-Profiler läuft. Weitere Hinweise zu Sicherheitsfragen finden Sie unter „[Verschlüsselung der MobiLink-Client/Server-Kommunikation](#)“ [*SQL Anywhere Server - Datenbankadministration*].

#### Hinweis

Erforderliche getrennt lizenzierbare Komponenten.

FIPS-zertifizierte Verschlüsselung erfordert eine separate Lizenz. Alle Technologien für starke Verschlüsselungen unterliegen Exportbestimmungen.

Siehe „[Getrennt lizenzierbare Komponenten](#)“ [*SQL Anywhere 16 - Einführung*].

- **Datei des vertrauenswürdigen Stammzertifikats** Wenn Sie HTTPS oder TLS als Protokoll wählen, geben Sie den Namen der Datei für das vertrauenswürdige Zertifikat an, das für sichere Verbindungen zum MobiLink-Server verwendet werden soll. Bei Windows-Plattformen wird der Speicher für vertrauenswürdige Zertifikate verwendet, wenn keine Datei für das vertrauenswürdige Zertifikat angegeben wird. Auf Nicht-Windows-Plattformen muss für eine sichere Verbindung eine Datei für das vertrauenswürdige Zertifikat angegeben werden.
- **Weitere Protokolloptionen** Geben Sie in diesem Feld optionale Netzwerkparameter an. Die zulässigen Werte hängen vom Typ des Verbindungsdatenstroms ab. Mehrere Parameter müssen durch Semikola voneinander getrennt werden.

Alle gültigen Netzwerkprotokolloptionen für den MobiLink-Client werden unterstützt, ausgenommen jene, die in diesem Fenster bereits festgelegt sind, wie etwa der Host, der Port und das vertrauenswürdige Zertifikat.

Siehe „Netzwerkprotokolloptionen des MobiLink-Clients“ [[MobiLink - Clientadministration](#)].

3. Starten Sie die Synchronisation.
4. Klicken Sie auf die Schaltfläche **Pause**, um das automatische Scrollen der Informationen in Diagramm und Auslastungsdiagramm anzuhalten.

### Ergebnisse

Der MobiLink-Profiler beginnt, Daten zu sammeln, und die Profilerstellungsdaten werden während des Sammelns angezeigt.

## Profilerstellungssitzungen beenden

Durch Beenden einer MobiLink-Profiler-Sitzung wird die Sammlung der Daten gestoppt, aber der MobiLink-Profiler läuft weiter, sodass Sie Ihre Daten anzeigen oder eine neue Profilerstellungssitzung starten können.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Aufgabe

1. Klicken Sie auf **Datei » Profiler-Sitzung beenden**. Dadurch wird die Sammlung der Daten gestoppt und der Profiler vom MobiLink-Server getrennt.

Sie können das Sammeln der Daten auch stoppen, indem Sie den MobiLink-Server herunterfahren oder den MobiLink-Profiler schließen.

2. Wenn Sie den MobiLink-Profiler schließen möchten, klicken Sie auf **Datei » Schließen**.

### Ergebnisse

Der MobiLink-Profiler stoppt das Sammeln von Profilerstellungsinformationen.

## Frühere Profilerstellungssitzungen öffnen oder löschen

Wählen Sie im Fenster **MobiLink-Profiler-Sitzung öffnen** eine frühere MobiLink-Profiler-Sitzung, die Sie öffnen oder löschen möchten.

### Voraussetzungen

In der Profilerstellungsdatenbank müssen frühere Profiler-Sitzungen vorhanden sein.

## Aufgabe

1. Klicken Sie im MobiLink-Profiler auf **Datei » Sitzung öffnen**. Vorherige Profilerstellungssitzungen in der Profilerstellungsdatenbank werden angezeigt.
2. Wählen Sie die Profilerstellungssitzung aus, die Sie öffnen oder löschen möchten.
3. Klicken Sie auf **OK**, um die ausgewählte Sitzung zu öffnen, oder auf **Löschen**, um die ausgewählte Sitzung aus der Profilerstellungsdatenbank zu entfernen.

## Ergebnisse

Wenn Sie auf **OK** geklickt haben, werden die Daten aus der ausgewählten Sitzung angezeigt.

Wenn Sie auf **Löschen** geklickt haben, wird die ausgewählte Sitzung aus der Profilerstellungsdatenbank gelöscht und nicht mehr in der Liste angezeigt.

## Nächste Schritte

Sie können die Daten überprüfen, wenn Sie eine frühere Profiler-Sitzung geöffnet haben.

# Die Profilerstellungsdatenbank

Wenn der MobiLink-Profiler erstmals gestartet wird, erstellt er eine Profilerstellungsdatenbank mit Standardwerten für Dateiname, Benutzer und Kennwort. Die Profilerstellungsdatenbank wird im Ordner *MLProfiler16* Ihres Dokumentenverzeichnisses gespeichert. Wenn Sie für die Profilerstellungsdatenbank nicht den Standardbenutzer und das Standardkennwort verwenden möchten, ändern Sie den Speicherort der Datenbank in eine vorhandene SQL Anywhere-Datenbank, die Sie verwenden möchten, oder kopieren Sie diese Datenbank an den Standardspeicherort. Beim nächsten Starten des MobiLink-Profilers können Sie die Benutzer-ID und das Kennwort für diese Datenbank eingeben und sie wird zur Profilerstellungsdatenbank. Verbindungsinformationen werden gespeichert, sobald eine Verbindung erfolgreich hergestellt wurde.

Wenn die Verbindung mit der Profilerstellungsdatenbank besteht und ein nicht kompatibles Schema erkannt wird, werden Sie gefragt, ob Sie das Datenbankschema für die Profilerstellung neu erstellen möchten, und anschließend wird der MobiLink-Profiler geschlossen. Wenn Sie die Option wählen, die Profilerstellungsdatenbank neu zu erstellen, erstellt der Profiler beim Neustart das Schema neu, wodurch auch alle Daten aus früheren Profilerstellungssitzungen gelöscht werden. Daten können verloren gehen, es sei denn, Sie erstellen vor dem Neustart des MobiLink-Profilers eine Sicherung.

Wenn Sie den Namen und den Speicherort der Profilerstellungsdatenbank ermitteln möchten, wechseln Sie auf die Seite **Allgemein** des Fensters **Optionen**.

# MobiLink-Profiler-Benutzeroberfläche

Der MobiLink-Profiler verfügt über die folgenden Fensterausschnitte:

- **Detailtabelle** **Detailtabelle** ist, sofern aktiviert, der oberste Fensterausschnitt. Diese Tabelle zeigt standardmäßig an, wie lang die einzelnen Synchronisationen gedauert haben, wobei jeweils aufgeschlüsselt ist, wie viel Zeit die einzelnen Phasen der Synchronisation in Anspruch genommen haben.

Siehe „Fensterausschnitt Detailtabelle“ auf Seite 198.

- **Auslastungsdiagramm** **Auslastungsdiagramm** ist, sofern aktiviert, der zweite Fensterausschnitt. Er enthält eine grafische Darstellung der Anzahl der Synchronisationen in den einzelnen Phasen. Für die Fensterausschnitte **Auslastungsdiagramm** und **Diagramm** wird die gleiche horizontale Skala verwendet. Die Skala am unteren Rand des Fensterausschnitts **Diagramm** zeigt die Zeit an. Sie können die im Auslastungsdiagramm angezeigten Daten auswählen, indem Sie die Daten im Fensterausschnitt **Übersicht** ziehen und auswählen oder auf **Ansicht » Gehe zu** klicken.

Siehe „Fensterausschnitt Auslastungsdiagramm“ auf Seite 200.

- **Diagramm** **Diagramm** ist der dritte Fensterausschnitt und wird immer angezeigt. Er enthält eine grafische Darstellung der Synchronisationen, in der die Synchronisationsphasen farblich gekennzeichnet sind. Die Skala am unteren Rand dieses Fensterausschnitts zeigt die Zeit an. Sie können die im Diagramm angezeigten Daten auswählen, indem Sie die Daten im Fensterausschnitt **Übersicht** ziehen und auswählen oder auf **Ansicht » Gehe zu** klicken.

Siehe „Fensterausschnitt Diagramm“ auf Seite 201.

- **Übersicht** **Übersicht** ist der unterste Fensterausschnitt. Es enthält eine Übersicht über alle Synchronisationen in der Sitzung. Dieser Fensterausschnitt enthält das sogenannte **Auswahlrechteck**, mit dem Sie den in den Fensterausschnitten **Diagramm** und **Auslastungsdiagramm** anzuzeigenden Bereich auswählen können.

Siehe „Fensterausschnitt Übersicht“ auf Seite 203.

Außerdem können Sie die Anzeige mithilfe des Fensters **Optionen** anpassen sowie verschiedene Eigenschaftsfenster anzeigen. Siehe „Fenster Optionen“ auf Seite 204.

## Fensterausschnitt Detailtabelle

Die **Detailtabelle** enthält Informationen zu den Synchronisationen, einschließlich der Dauer der Phasen. Alle Zeitangaben werden vom MobiLink-Server gemessen. Möglicherweise ist die Dauer einiger Phasen ungleich Null, auch wenn Sie keine entsprechenden Skripten definiert haben.

Sie können die im Fensterausschnitt **Detailtabelle** anzuzeigenden Spalten wählen, indem Sie auf **Extras » Optionen** klicken und anschließend die Registerkarte **Tabelle** öffnen. Weitere Hinweise zu den verfügbaren Statistiken finden Sie unter „Statistikeigenschaften für die MobiLink-Synchronisation“ auf Seite 209.

Die folgenden Spalten werden standardmäßig angezeigt:

- **Nummer** Gibt die einzelnen Synchronisationen an. Diese Nummer wird vom MobiLink-Server zugeordnet, nicht vom MobiLink-Profiler. Daher beginnt sie für MobiLink-Profiler-Sitzungen nicht

unbedingt bei 1 und wird nicht unbedingt in numerischer Reihenfolge vergeben. Diese Nummer ist gleich der Synchronisationsnummer, die in Warnungen, Fehlermeldungen und Logs des MobiLink-Servers angezeigt wird. Dieselbe Nummer wird auch im Fenster **Synchronisationseigenschaften** angezeigt. Siehe „[Synchronisationseigenschaften](#)“ auf Seite 205.

- **Entfernte\_ID** Der eindeutige Bezeichner der entfernten Datenbank. Siehe „[Entfernte IDs](#)“ [[MobiLink - Clientadministration](#)].
- **Benutzer** Der MobiLink-Synchronisationsbenutzer. Siehe „[MobiLink-Benutzer](#)“ [[MobiLink - Clientadministration](#)].
- **Version** Die Version des Synchronisationsskripts.  
Siehe „[Skriptversionen](#)“ auf Seite 254.
- **Startzeit** Das Datum und die Uhrzeit des Beginns der Synchronisation durch den MobiLink-Server. (Dieser Zeitpunkt kann nach der Synchronisationsanforderung durch den Client liegen.)
- **Dauer** Die gesamte Dauer der Synchronisation in Sekunden.

Die Dauer aller folgenden Phasen wird in Sekunden angegeben.

- **Synch\_Anforderung-Phase** Die Zeit ab der Herstellung der Netzwerkverbindung zwischen der entfernten Datenbank und dem MobiLink-Server bis zum Empfang der ersten Bytes des Upload-Datenstroms. Diese Zeit ist unerheblich, sofern Sie nicht -sm auf einen kleineren Wert als -nc gesetzt haben. In diesem Fall kann diese Zeit auch die Zeit enthalten, für die eine Synchronisation angehalten wird, wenn die Anzahl der Synchronisationen größer ist als die mit -sm festgelegte maximale Anzahl aktiver Synchronisationen.
- **Upload\_empfangen-Phase** Die Zeitspanne ab dem Empfang der ersten Bytes des Uploaddatenstroms durch den MobiLink-Server bis zum vollständigen Abschluss der Übertragung des Uploaddatenstroms aus der entfernten Datenbank. Der Uploaddatenstrom enthält Tabellendefinitionen und die Uploaddatenbankzeilen, daher kann die Zeitspanne auch bei einer nur als Download durchgeführten Synchronisation signifikant sein. Die Zeit hängt von der Größe des Uploaddatenstroms und der Netzwerkbandbreite für die Übertragung ab.
- **DB\_Worker\_abrufen-Phase** Die Zeit, die benötigt wird, um einen freien Datenbank-Worker-Thread abzurufen.
- **Verbinden-Phase** Die Zeit, die der Datenbank-Worker-Thread benötigt, um bei Bedarf eine neue Datenbankverbindung herzustellen, z.B. nach einem Fehler oder wenn sich die Skriptversion geändert hat.
- **Authentifizierung\_Benutzer-Phase** Die Zeit, die MobiLink benötigt, um die Synchronisationsanforderung, den Benutzernamen und das Kennwort zu validieren (sofern Ihre Synchronisationsumgebung eine Authentifizierung erfordert). Dies ist die Länge der Transaktion zur Authentifizierung des Benutzers (vom Beginn der Authentifizierung bis unmittelbar vor dem Beginn des Ereignisses begin\_synchronization).
- **Synch\_Beginn-Phase** Die Zeit, in der das begin\_synchronization-Skript ausgeführt wurde, falls zutreffend.

- **Upload\_übernehmen-Phase** Die Zeit, in der der Upload in die konsolidierte Datenbank übernommen wird. Dies ist die Zeit zwischen der Ausführung des begin\_upload- und des end\_upload-Skripts.
- **Download\_Vorbereitung-Phase** Die Zeit, in der das prepare\_for\_download-Skript ausgeführt wurde, falls zutreffend.
- **Download\_abrufen-Phase** Die Zeit, in der die herunterzuladenden Zeilen aus der konsolidierten Datenbank abgerufen werden. Dies ist die Zeit zwischen der Ausführung des begin\_download- und des end\_download-Skripts.
- **Synch\_Ende-Phase** Die Zeit, in der das end\_synchronization-Skript ausgeführt wurde, falls zutreffend.
- **Download\_senden-Phase** Die Zeit, die benötigt wird, um den Downloaddatenstrom an die entfernte Datenbank zu senden.
- **Auf\_Download\_Quittierung\_warten-Phase** Bei aktivierter Downloadbestätigung enthält dieser Wert die Wartezeit, bis der Download in die entfernte Datenbank übernommen und die Downloadbestätigung von der entfernten Datenbank gesendet wird.
- **DB\_Worker\_für\_Download\_Quittierung-Phase** Bei aktivierter Downloadbestätigung enthält dieser Wert die Zeit, die auf einen freien Datenbank-Worker-Thread gewartet wird, nachdem die Downloadbestätigung empfangen wurde.
- **Verbinden\_für\_Download\_Quittierung-Phase** Bei aktivierter Downloadbestätigung enthält dieser Wert die Zeit, die der Datenbank-Worker-Thread benötigt, um bei Bedarf eine neue Datenbankverbindung herzustellen.
- **Nicht\_blockierende\_Download\_Quittierung-Phase** Bei aktivierter Downloadbestätigung enthält dieser Wert die Zeit, die für die Verbindungsereignisse publication\_nonblocking\_download\_ack und nonblocking\_download\_ack benötigt wird.

Wenn Sie die Tabelle nach einer bestimmten Spalte sortieren wollen, klicken Sie auf die entsprechende Spaltenüberschrift. Neu im MobiLink-Profiler angezeigte Daten werden beim Hinzufügen sortiert.

Sie können den Fensterausschnitt **Detailtabelleschließen**, indem Sie im Menü **Ansicht** die Option **Detailtabelle** deaktivieren.

## Fensterausschnitt Auslastungsdiagramm

Das **Auslastungsdiagramm** ist der zweite Fensterausschnitt von oben. Die Anzahl der Synchronisationen, die sich in den einzelnen Phasen befinden, wird hier in einem Zeitdiagramm angezeigt.

Weitere Hinweise zu den Daten in diesem Fensterausschnitt finden Sie unter [„Funktionsweise des Fensterausschnitts Auslastungsdiagramm“ auf Seite 201](#).

Das **Auslastungsdiagramm** wird mit der gleichen horizontalen Bildlaufleiste, den gleichen horizontalen Zeitlabeln sowie der gleichen horizontalen Zoomstufe angezeigt wie das **Diagramm**. Das bedeutet, dass Zeitpunkte vertikal zwischen den beiden Diagramm-Fensterausschnitten abgeglichen werden.

Es gibt zwei Möglichkeiten, den in den beiden Diagramm-Fensterausschnitten anzuzeigenden Zeitbereich auszuwählen:

- Klicken Sie im Menü **Ansicht** auf die Option **Gehe zu**.
- Bewegen Sie das **Auswahlrechteck** im Fensterausschnitt **Übersicht**. Das **Auswahlrechteck** ist das kleine Kästchen im Fensterausschnitt **Übersicht**.

Wenn Sie auf einen Bereich im **Auslastungsdiagramm** doppelklicken, wird das Fenster **Beispieleigenschaften** geöffnet, das Details zum jeweiligen Beispielintervall enthält. Die Dauer des Beispielintervalls beträgt ca. eine Sekunde.

Ziehen Sie die Maus im Fensterausschnitt **Auslastungsdiagramm**, um Daten für einen Bereich von Beispielen anzuzeigen. Das Fenster **Beispiele für Bereichseigenschaften** wird angezeigt.

#### Siehe auch

- [„Funktionsweise des Fensterausschnitts Auslastungsdiagramm“ auf Seite 201](#)
- [Auswahlrechteck auf Seite 203](#)
- [„Beispieleigenschaften“ auf Seite 204](#)

## Funktionsweise des Fensterausschnitts Auslastungsdiagramm

Wenn Sie den Fensterausschnitt **Auslastungsdiagramm** anpassen möchten, klicken Sie auf **Extras » Optionen** und anschließend auf die Registerkarte **Diagramm**. Auf dieser Registerkarte werden die Zeiten im **Auslastungsdiagramm** farblich gekennzeichnet und Sie können das Diagramm anpassen.

#### Phasenzähler

Jede Eigenschaft zeigt die Anzahl der Synchronisationen, die sich derzeit in der betreffenden Phase befinden.

#### Antialiasing

Eine Ihrer Anpassungsoptionen ist das sogenannte Antialiasing. Mit Antialiasing können Sie die Darstellung des Diagramms verbessern, es kann jedoch sein, dass es sich langsamer zeichnen lässt.

## Fensterausschnitt Diagramm

Der Fensterausschnitt **Diagramm** enthält dieselben Informationen wie die Standardspalten der **Detailtabelle**, jedoch in grafischem Format. Die Balken im Fenster **Diagramm** repräsentieren die Dauer der einzelnen Synchronisationen und die Unterabschnitte der Balken repräsentieren die Synchronisationsphasen.

#### Daten anzeigen

Durch Klicken auf eine Synchronisation in der **Detailtabelle** wählen Sie sie aus.

Doppelklicken Sie auf eine Synchronisation, um das Fenster **Synchronisationseigenschaften** zu öffnen. Siehe „[Synchronisationseigenschaften](#)“ auf Seite 205.

### Daten nach ID der entfernten Datenbank oder kompakt gruppieren

Um die Daten nach entfernter ID zu gruppieren, klicken Sie auf **Ansicht » Nach ID der entfernten Datenbank**.

Alternativ können Sie die Daten kompakt anzeigen, wobei alle Synchronisationen in möglichst wenigen Zeilen angezeigt werden. Klicken Sie auf **Ansicht » Kompaktansicht**. In der **Kompaktansicht** sind die Nummern der Zeilen ohne Bedeutung.

### Daten zoomen

Es gibt mehrere Möglichkeiten, die in den Fensterausschnitten **Diagramm** und **Auslastungsdiagramm** sichtbaren Daten auszuwählen:

- **Zoomoptionen** Mit den Zoomoptionen im Menü **Ansicht** und den Zoomschaltflächen in der Symbolleiste können Sie die Darstellung vergrößern und verkleinern. Mit **Ausschnitt vergrößern** können Sie eine Synchronisation über den verfügbaren Bereich ausdehnen.
- **Bildlaufleiste** Klicken Sie auf die Bildlaufleiste unten im Fensterausschnitt **Diagramm** und verschieben Sie sie.
- **Fenster Gehe zu** Zum Öffnen dieses Fensters klicken Sie auf **Ansicht » Gehe zu**.

**Startdatum und Startzeit** Legen Sie die Startzeit für die im Fensterausschnitt **Diagramm** angezeigten Daten fest. Wenn Sie diese Einstellungen ändern, müssen Sie mindestens das Jahr, den Monat und den Tag angeben.

**Diagrammbereich** Legen Sie die Dauer des angezeigten Zeitraums fest. Der Diagrammbereich kann in Millisekunden, Sekunden, Minuten, Stunden oder Tagen angegeben werden. Der Diagrammbereich legt die Granularität der Daten fest: Bei einer kürzeren Dauer werden mehr Details angezeigt.

- **Auswahlrechteck** Im Fensterausschnitt **Übersicht** können Sie das **Auswahlrechteck** ziehen und ändern. Das **Auswahlrechteck** ist das Kästchen im Fensterausschnitt **Übersicht**. Siehe [Auswahlrechteck auf Seite 203](#).

### Zeitachse

Unten im Fensterausschnitt **Diagramm** befindet sich eine Skala, die die Zeitperioden anzeigt. Das Format der Zeit wird automatisch abhängig von der angezeigten Zeitspanne angepasst. Sie können Datum und Uhrzeit jederzeit vollständig anzeigen, indem Sie den Cursor auf die Skala positionieren.

### Standard-Farbschema

Sie können die Farben im Fensterausschnitt **Diagramm** anzeigen, indem Sie das Fenster **Optionen** öffnen (verfügbar über das Menü **Extras**). Das Standardfarbschema für den Fensterausschnitt **Diagramm** verwendet die Farbe Limonengrün für Uploads, Korallenrot für Downloads und Blau für Beginn- und Endphasen mit einem dunkleren Farbton für die früheren Teile einer Phase.

Weitere Hinweise zur Einstellung der Farben finden Sie unter „[Fenster Optionen](#)“ auf Seite 204.

## Fensterausschnitt Übersicht

Der Fensterausschnitt **Übersicht** gibt Ihnen einen Überblick über die gesamte MobiLink-Profiler-Sitzung. Sie können mit dem **Auswahlrechteck**, d.h. dem Kästchen im Fensterausschnitt **Übersicht**, durch die Sitzung navigieren.

Standardmäßig werden aktive Synchronisationen, blockierte Synchronisationen, abgeschlossene Synchronisationen und fehlgeschlagene Synchronisationen über Überwachungen farbig dargestellt. Wenn Sie die Farben einstellen möchten, öffnen Sie den MobiLink-Profiler, klicken Sie auf **Extras » Optionen** und anschließend auf die Registerkarte **Übersicht**. Sie können auch die entsprechenden Überwachungen bearbeiten, indem Sie auf **Extras » Überwachungsmanager** und anschließend auf **Bearbeiten** klicken.

Siehe:

- „Fenster Optionen“ auf Seite 204
- „Statistikanpassung“ auf Seite 205

Sie können den Fensterausschnitt **Übersicht** schließen, indem Sie ihn im Menü **Ansicht** deaktivieren.

Außerdem können Sie den Fensterausschnitt **Übersicht** vom Rest des MobiLink-Profiler-Fensters trennen. Klicken Sie dazu im Fenster **Optionen** auf die Registerkarte **Übersicht** und deaktivieren Sie das Kontrollkästchen **Übersichtsfenster an Hauptfenster angehängt lassen**.

### Auswahlrechteck

Das **Auswahlrechteck** ist das kleine Kästchen im Fensterausschnitt **Übersicht**. Sie können das **Auswahlrechteck** verwenden, um andere Daten anzuzeigen oder Daten mit unterschiedlicher Granularität darzustellen. Der innerhalb des Auswahlrechtecks dargestellte Bereich wird in den Diagramm-Fensterausschnitten angezeigt. Sie können das **Auswahlrechteck** folgendermaßen verwenden:

- Klicken Sie in den Fensterausschnitt **Übersicht**, um das **Auswahlrechteck** und die Startzeit der Daten zu verschieben, die im Diagramm bzw. Auslastungsdiagramm angezeigt werden.
- Im Fensterausschnitt **Übersicht** ziehen Sie mit der Maus an der Umrandung des **Auswahlrechtecks**, um Position und Größe des Rechtecks und damit die Startzeit und den Bereich der Daten zu ändern. Wenn Sie das Auswahlrechteck kleiner machen, verkürzen Sie damit das Intervall der im Diagramm angezeigten Daten, wodurch mehr Details zu sehen sind.

### Farbe des Auswahlrechtecks ändern

Sie können die Farbe des **Auswahlrechtecks** ändern.

#### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

#### Ändern der Farbe des Auswahlrechtecks

1. Klicken Sie auf **Extras » Optionen**.

2. Klicken Sie auf die Registerkarte **Überblick**.
3. Wählen Sie im Feld **Auswahlrechteck** eine neue Farbe aus.
4. Klicken Sie auf **OK**.

### Ergebnisse

Die Farbe des **Auswahlrechtecks** wird geändert.

## Fenster Optionen

Mithilfe von Optionen können Sie zahlreiche Einstellungen festlegen, einschließlich Farben und Muster für die grafische Anzeige in den Fensterausschnitten **Diagramm**, **Auslastungsdiagramm** und **Übersicht**.

Im Fenster **Optionen**, auf der Registerkarte **Allgemein**, können Sie die zum Speichern der Profilerstellungsdaten verwendete Datenbank ändern sowie die Profilerstellungsdatenbank neu erstellen, wodurch alle früheren Profilerstellungssitzungen entfernt werden.

Sie können das Fenster **Optionen** öffnen, indem Sie den MobiLink-Profiler öffnen und auf **Extras » Optionen** klicken.

## Sitzungseigenschaften

Das Fenster **Sitzungseigenschaften** enthält Statistiken zur Profilerstellungssitzung. Es liefert Eigenschaftswerte für die aktuelle Profilerstellungssitzung. Um das Fenster **Sitzungseigenschaften** zu öffnen, klicken Sie auf **Datei » Eigenschaften**.

## Beispieleigenschaften

Sie können das Fenster **Beispieleigenschaften** öffnen, indem Sie im Fensterausschnitt **Diagramm** auf den Zeitraum klicken, den Sie untersuchen möchten. Das Fenster **Beispieleigenschaften** enthält Statistiken zu den Zeitintervallen. Jedes Zeitintervall dauert eine Sekunde. Beispiele werden vom MobiLink-Profiler in der Reihenfolge ihres Eingangs nummeriert.

Das Fenster **Beispieleigenschaften** enthält drei Registerkarten:

- **Allgemein** Liefert eine zusammenfassende Aufschlüsselung der Synchronisationsvorgänge, die zum Zeitpunkt der Beispielerstellung ausgeführt wurden.
- **Phasen** Liefert Anzahlen der Phasen, in denen Ihre Synchronisationen zum Zeitpunkt der Beispielerstellung waren.
- **Ereignisse** Liefert Informationen zu Ereignisskripten, die während der Synchronisation ausgeführt werden.

Sie können das Erscheinungsbild des Diagramms so anpassen, dass Eigenschaften ausgeblendet werden. Im Fenster **Beispieleigenschaften** werden jedoch alle Eigenschaften angezeigt. Wenn Sie eine Phase

ausgeblendet haben, wird sie auf der Registerkarte **Phasen** des Fensters **Beispieleigenschaften** als **Ausgeblendet** gekennzeichnet. Andernfalls wird die betreffende Farbe angezeigt.

Im Fenster **Beispiele für Bereichseigenschaften** werden Informationen für mehrere Beispiele angezeigt, wenn Sie durch Ziehen im **Auslastungsdiagramm** mehrere Beispiele ausgewählt haben.

Die **Beispiele für Bereichseigenschaften** Fenster hat dieselben Registerkarten wie das Fenster **Beispieleigenschaften**. Durchschnittlicher und maximaler Wert werden jedoch für den Bereich angezeigt.

## Synchronisationseigenschaften

Sie können die Eigenschaften für eine Synchronisation anzeigen, indem Sie im Fensterausschnitt **Detailtabelle** oder im Fensterausschnitt **Diagramm** auf eine Synchronisation doppelklicken.

Sie können wählen, Statistiken für alle Tabellen (also die Summe für alle Tabellen in der Synchronisation) oder nur für einzelne Tabellen anzuzeigen. Die Dropdown-Liste enthält eine Liste der Tabellen, die an der Synchronisation beteiligt waren.

Auf der Seite **Synchronisation** werden Warnungen bzw. Fehler angezeigt und auf der Seite **Ereignisse** wird angezeigt, wie oft die Ereignisskripten aufgerufen wurden und wie lang sie gedauert haben. Wenn der MobiLink-Server erkannt hat, dass die Synchronisation blockiert war, ist die Seite **Blockiert** verfügbar.

Um weitere Beschreibungen der auf einer Seite des Fensters **Synchronisationseigenschaften** angegebenen Mengen anzuzeigen, klicken Sie auf **Hilfe**.

Eine Erklärung der Statistiken im Fenster **Synchronisationseigenschaften** finden Sie unter [„Statistikeigenschaften für die MobiLink-Synchronisation“ auf Seite 209](#).

## Statistikanpassung

Der **Überwachungsmanager** gestattet es Ihnen, sichtbar zwischen Synchronisationen zu unterscheiden, die die von Ihnen angegebenen Kriterien erfüllen. Sie können z.B. große Synchronisationen, lange Synchronisationen und kleine Synchronisationen markieren, die lange dauern, oder Synchronisationen, die Warnungen erhalten.

Sie können den **Überwachungsmanager** öffnen, indem Sie den MobiLink-Profiler öffnen und auf **Extras » Überwachungsmanager** klicken.

Der linke Fensterausschnitt des **Überwachungsmanagers** enthält eine Liste aller verfügbaren Überwachungen. Der rechte Fensterausschnitt enthält eine Liste der aktiven Überwachungen. Um in der Liste der aktiven Überwachungen eine Überwachung hinzuzufügen oder zu entfernen, wählen Sie im linken Fensterausschnitt eine Überwachung aus und klicken auf die betreffende Schaltfläche.

Es gibt vier vordefinierte Überwachungen (**Aktiv**, **Blockiert**, **Abgeschlossen** und **Fehlgeschlagen**). Sie können vordefinierte Überwachungen bearbeiten, um ihre Darstellung zu ändern, oder sie deaktivieren, indem Sie sie aus dem rechten Fensterausschnitt entfernen.

Im Diagramm werden keine Synchronisationen angezeigt, wenn sie die Bedingungen einer Überwachung nicht erfüllen. Wenn Sie alle Überwachungen deaktivieren (indem Sie sie aus der Liste **Laufende Überwachungen** entfernen), werden in den Fensterausschnitten **Diagramm** und **Übersicht** keine Synchronisationen angezeigt.

Die Reihenfolge der Überwachungen im rechten Fensterausschnitt ist wichtig. Die Überwachungen oben in der Liste werden zuerst ausgeführt. Mit den Schaltflächen **Aufwärts** und **Abwärts** können Sie die Reihenfolge der Überwachungen im rechten Fensterausschnitt ändern.

Mithilfe der vordefinierten Überwachungen können Sie andere Überwachungen erstellen. Wenn Sie eine Überwachungsbedingung bearbeiten möchten, löschen Sie sie und fügen anschließend die neue Bedingung hinzu.

Wenn ein neuer MobiLink-Profilierer eine Verbindung mit demselben MobiLink-Server herstellt, wird er in allen bereits verbundenen MobiLink-Profilern als kurze Synchronisation angezeigt. Die MobiLink-Profilierer-Synchronisation hat den Versionsnamen `for_ML_Monitor_only`. Sie können diese MobiLink-Profilierer-Synchronisation ausblenden, indem Sie nur Überwachungen mit der folgenden Bedingung aktivieren:

- **Eigenschaft** Setzen Sie dies auf **Version**
- **Operator** Setzen Sie dies auf **ist nicht gleich**
- **Wert** Setzen Sie dies auf **for\_ML\_Monitor\_only**

## Neue Überwachungen erstellen

Fügen Sie eine Überwachung hinzu, um Synchronisationen anzuzeigen, die die definierten Überwachungskriterien erfüllen.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Aufgabe

1. Im **Überwachungsmanager** klicken Sie auf **Neu**.
2. Geben Sie den Namen der neuen Überwachung im Feld **Name** ein.
3. Wählen Sie eine **Eigenschaft**, einen **Operator** und einen **Wert**.

Eine vollständige Liste der Eigenschaften finden Sie unter „[Statistikeigenschaften für die MobiLink-Synchronisation](#)“ auf Seite 209.

4. Klicken Sie auf **Hinzufügen**. (Sie müssen auf **Hinzufügen** klicken, um die Bedingung zu speichern.)
5. Falls gewünscht, können Sie eine weitere **Eigenschaft**, einen weiteren **Operator** und einen weiteren **Wert** wählen und auf **Hinzufügen** klicken.

6. Wählen Sie ein **Diagrammmuster** für die Überwachung im Fensterausschnitt **Diagramm** aus.
7. Wählen Sie eine **Farbe in der Übersicht** für die Überwachung im Fensterausschnitt **Übersicht** aus.
8. Klicken Sie auf **OK**.

### Ergebnisse

Die neue Überwachung wird erstellt.

## Profilerstellungsdatenbank verwenden

In Sybase Central können Sie vordefinierte Ansichten verwenden, um Daten in der Profilerstellungsdatenbank zu überprüfen und zu analysieren.

### Voraussetzungen

Sie müssen mindestens eine Profilerstellungssitzung ausgeführt haben.

Sie müssen den Namen und den Speicherort der Profilerstellungsdatenbank kennen. Diese Informationen finden Sie auf der Seite **Allgemein** des Fensters **Optionen**.

### Kontext und Bemerkungen

Bei dieser Aufgabe wird davon ausgegangen, dass Sie die Standard-Profilerstellungsdatenbank verwenden.

Sie können auch Interactive SQL verwenden, um mit Profiler-Ansichten zu arbeiten.

### Aufgabe

1. Stellen Sie mithilfe des **SQL Anywhere 16**-Plug-Ins eine Verbindung mit der Profilerstellungsdatenbank her und verwenden Sie dabei die folgenden Optionen:
  - **Benutzer-ID** Geben Sie für die Benutzer-ID **mlprofiler** ein.
  - **Kennwort** Geben Sie für das Kennwort **sql** ein.
  - **Aktion** Wählen Sie **Eine Datenbank auf diesem Computer starten und eine Verbindung herstellen**.
  - **Datenbankdatei** Geben Sie die Pfadinformationen für die Profilerstellungsdatenbank ein oder klicken Sie auf **Durchsuchen**, um die Datei auszuwählen. Die Standard-Datenbankdatei heißt *mlprofiler.db* und befindet sich in einem Ordner namens *MLProfiler16* in Ihrem Dokumentenordner.
  - **Servername** Geben Sie **MLProfilerDB** ein.
  - **Startzeile** Wenn Sie den anfänglichen Speicher für das Caching von Datenbankseiten und anderen Datenbankserver-Informationen festlegen möchten, geben Sie Folgendes ein:

```
dbeng16.exe -c lg
```

2. Klicken Sie auf **Verbinden**.
3. Erweitern Sie die Datenbank mlprofiler und doppelklicken Sie auf **Ansichten**, um eine Liste der MobiLink-Profiler-Ansichten anzuzeigen.
4. Wählen Sie eine Ansicht aus. Die folgenden Ansichten sind verfügbar:
  - **category\_samples** (Basisansicht für Kategorie-Stichprobendaten)
  - **data\_event\_statistics**
  - **data\_event\_times**
  - **data\_phase\_statistics**
  - **data\_phase\_times**
  - **event\_samples** (Basisansicht für Ereignis-Stichprobendaten)
  - **event\_statistics**
  - **event\_times**
  - **event\_total\_times**
  - **phase\_samples** (Basisansicht für Phasen-Stichprobendaten)
  - **phase\_statistics**
  - **phase\_times**
  - **server\_cumulative\_samples** (Basisansicht für kumulative serverbezogene Stichprobendaten)
  - **server\_snapshot\_samples** (Basisansicht für nicht kumulative serverbezogene Stichprobendaten)
  - **sync\_as\_csv** (Ansicht wie im alten .csv-Dateiformat für den MobiLink-Monitor)
  - **sync\_blocked**
  - **sync\_statistics**
  - **sync\_times**
  - **syncs** (Basisansicht für Synchronisationen)

Im Fensterausschnitt "SQL" auf der rechten Seite wird oben ein Kommentar angezeigt, der die ausgewählte Ansicht beschreibt.

#### **Hinweis**

Die serverbezogenen Stichprobendaten werden für Metriken verwendet, die auch im SQL Anywhere-Monitor für MobiLink verfügbar sind.

## **Ergebnisse**

Daten aus der Profilerstellungsdatenbank werden für die einzelnen Ansichten auf der Seite **Daten** angezeigt.

## **Nächste Schritte**

- Überprüfen Sie die Profilerstellungsdaten.
- Verwenden Sie diesen Ansichten in Ihren Abfragen.

## **Beispiel**

Die folgende Beispielabfrage zeigt die Ereignisskripten, die bei allen Synchronisationen in der zweiten Sitzung die meiste Zeit in Anspruch genommen haben:

```
select * from event_total_times where session_id = 2 order by 1 desc
```

Die folgende Beispielabfrage zeigt die schnellsten Synchronisationsabschlüsse für alle Sitzungen:

```
select
    max( "Successful syncs/s" ) as "Max syncs/s",
    session_id
from server_throughput_samples
group by session_id
order by 1 desc, 2
```

## Statistikeigenschaften für die MobiLink-Synchronisation

Die folgende Liste enthält die Statistikeigenschaften, die im MobiLink-Profiler verfügbar sind. Diese Statistiken können im Fenster **Neue Überwachung**, dem Fensterausschnitt **Detaillabelle** oder im Fenster **Synchronisationseigenschaften** angezeigt werden. Im Fenster **Synchronisationseigenschaften** enthalten die Namen der Eigenschaften keine Unterstriche.

Weitere Hinweise zum Fenster **Neue Überwachung** finden Sie unter „[Statistikanpassung](#)“ auf Seite 205.

Weitere Hinweise zum Fenster **Detaillabelle** finden Sie unter „[Fensterausschnitt Detaillabelle](#)“ auf Seite 198.

Weitere Hinweise zum Fenster **Synchronisationseigenschaften** finden Sie unter „[Synchronisationseigenschaften](#)“ auf Seite 205.

### Synchronisationsstatistiken

MobiLink-Statistikeigenschaften geben die folgenden Informationen zu Synchronisationen zurück:

| Eigenschaft                      | Beschreibung                                                                                                                                                                     |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Aktiv                            | TRUE, wenn die Synchronisation gerade abläuft.                                                                                                                                   |
| Upload_übernehmen-Phase          | Die Zeit, die benötigt wird, um die hochgeladenen Daten in die konsolidierte Datenbank zu übernehmen.                                                                            |
| Authentifizierung_Benutzer-Phase | Gesamtzeit zur Ausführung der Benutzerauthentifizierung einschließlich Ausführung von Authentifizierungsereignissen (authenticate_*).                                            |
| Synch_Beginn-Phase               | Gesamtzeit für das Ereignis begin_synchronization.                                                                                                                               |
| Client                           | Der Typ des MobiLink-Clients und die vollständige Clientversion, z.B. Dbmlsync 16.0.0.xxxx.                                                                                      |
| Abgeschlossen                    | TRUE, wenn die Synchronisation erfolgreich abgeschlossen wurde.                                                                                                                  |
| Konflikt_Updates                 | Anzahl der aktualisierten Zeilen, die einen Konflikt verursacht haben. Eine Zeile wird nur dann einbezogen, wenn für sie ein Konfliktlösungsskript erfolgreich aufgerufen wurde. |

| Eigenschaft                              | Beschreibung                                                                                                                                                                             |
|------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Verbinden-Phase                          | Die Zeit, die der Datenbank-Worker-Thread benötigt, um bei Bedarf eine neue Datenbankverbindung herzustellen, Zum Beispiel nach einem Fehler oder wenn die Skriptversion geändert wurde. |
| Verbinden_für_Download_Quittierung-Phase | Die Zeit, die der Datenbank-Worker-Thread benötigt, um bei Bedarf eine neue Datenbankverbindung für eine Downloadbestätigung herzustellen,                                               |
| Verbindungsneuersuche                    | Anzahl der Versuche des MobiLink-Servers, eine Verbindung zur konsolidierten Datenbank herzustellen.                                                                                     |
| Download                                 | Diese Eigenschaft gibt an, dass die Synchronisation einen Downloadbefehl umfasst.                                                                                                        |
| Download-Quittierung                     | Kann "none" oder "non-blocking" sein.                                                                                                                                                    |
| Download_Byte                            | Der Speicherplatz, der im MobiLink-Server zum Speichern des Downloads und zum Senden an die entfernte Datenbank belegt wurde (vor Verschlüsselung oder Komprimierung).                   |
| Download_Löschzeilen                     | Anzahl der Zeilenlöschungen, die vom MobiLink-Server aus der konsolidierten Datenbank abgerufen wurden (mit download_cursor-Skripten).                                                   |
| Download_Fehler                          | Anzahl der Fehler, die während des Downloads aufgetreten sind.                                                                                                                           |
| Download_Abrufzeilen                     | Anzahl der Zeilen, die vom MobiLink-Server aus der konsolidierten Datenbank abgerufen wurden (mit download_delete_cursor-Skripten).                                                      |
| Download_Filterzeilen                    | Anzahl der abgerufenen Zeilen, die nicht auf den MobiLink-Client übertragen wurden, da sie mit Zeilen übereinstimmten, die der Client eingelesen hat.                                    |
| Download_Warnungen                       | Anzahl der Warnungen, die während des Downloads aufgetreten sind.                                                                                                                        |
| Dauer                                    | Gesamtzeit der Synchronisation, die vom MobiLink-Server gemessen wurde.                                                                                                                  |
| Synch_Ende-Phase                         | Gesamtzeit für das Ereignis end_synchronization.                                                                                                                                         |

| Eigenschaft                                   | Beschreibung                                                                                                                                                                                                                                                                                                                                    |
|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Download_abrufen-Phase                        | Die Zeit, die benötigt wird, um die herunterzuladenden Zeilen aus der konsolidierten Datenbank abzurufen, um den Download-Datenstrom zu erstellen.                                                                                                                                                                                              |
| DB_Worker_abrufen-Phase                       | Die Zeit, die benötigt wird, um einen freien Datenbank-Worker-Thread verfügbar zu machen.                                                                                                                                                                                                                                                       |
| DB_Worker_für_Download_Quittierung            | Die Zeit, die auf einen freien Datenbank-Worker-Thread gewartet wird, nachdem die Downloadbestätigung empfangen wurde.                                                                                                                                                                                                                          |
| Hat_blockiert                                 | TRUE, wenn vom MobiLink-Server eine Blockierung erkannt wird.                                                                                                                                                                                                                                                                                   |
| Ignorierte_Löschungen                         | Die Anzahl der Upload-Löschungszeilen, die zu Fehlern führten, während das Skript upload_delete aufgerufen wurde, wenn handle_error oder handle_odbc_error definiert ist und den Wert 1000 zurückgegeben hat oder wenn für die angegebene Tabelle kein upload_delete-Skript definiert ist.                                                      |
| Ignorierte_Einfügungen                        | Gesamtzahl der ignorierten einzufügenden Zeilen im Upload. Die Zeilen wurden ignoriert, weil 1) kein upload_insert-Skript im Normalmodus vorhanden ist oder 2) beim Aufrufen des entsprechenden Skripts durch den MobiLink-Server Fehler aufgetreten sind und das Ereignis handle_error oder handle_odbc_error der Wert 1000 zurückgegeben hat. |
| Ignorierte_Updates                            | Anzahl der Upload-Aktualisierungszeilen, die einen Konflikt verursacht haben, aber kein Konfliktlösungsskript erfolgreich aufgerufen oder kein upload_update-Skript definiert ist.                                                                                                                                                              |
| Nicht_blockierende_Download_Quittierung-Phase | Die Zeit, die für die Verbindungsereignisse publication_nonblocking_download_ack und nonblocking_download_ack benötigt wird.                                                                                                                                                                                                                    |
| Nummer                                        | Synchronisationsnummer.                                                                                                                                                                                                                                                                                                                         |
| Download_Vorbereitung-Phase                   | Gesamtzeit für das Ereignis prepare_for_download.                                                                                                                                                                                                                                                                                               |
| Upload_empfangen-Phase                        | Die Phasendauer für den Empfang des Uploads.                                                                                                                                                                                                                                                                                                    |
| Entfernte_ID                                  | Die entfernte ID, die die entfernte Datenbank eindeutig kennzeichnet.                                                                                                                                                                                                                                                                           |

| Eigenschaft             | Beschreibung                                                                                                                                                                                                                                                                                                 |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Download_senden-Phase   | Die Zeit, die erforderlich ist, um den Download-Datenstrom an die entfernte Datenbank zu senden. Die Zeit hängt von der Größe des Download-Datenstroms und der Netzwerkbandbreite für die Übertragung ab. Für eine reine Upload-Synchronisation ist der Download-Datenstrom einfach eine Upload-Bestätigung. |
| Server                  | Der MobiLink-Servername oder "Host:Port".                                                                                                                                                                                                                                                                    |
| Startzeit               | Datum-Uhrzeit (in erweitertem ISO-8601-Format) für den Start der Synchronisation.                                                                                                                                                                                                                            |
| Synch_Deadlocks         | Gesamtzahl der Deadlocks in der konsolidierten Datenbank, die für die Synchronisation festgestellt wurden.                                                                                                                                                                                                   |
| Synch_Fehler            | Anzahl der Fehler, die während der Synchronisation aufgetreten sind.                                                                                                                                                                                                                                         |
| Synch_Anforderung-Phase | Die Zeit vom Herstellen der Netzwerkverbindung zwischen der entfernten Datenbank und dem MobiLink-Server bis zum Empfang der ersten Bytes des Upload-Datenstroms.                                                                                                                                            |
| Synch_Tabellen          | Anzahl der Client-Tabellen, die an der Synchronisation beteiligt waren.                                                                                                                                                                                                                                      |
| Synch_Warnungen         | Anzahl der Warnungen, die während der Synchronisation ausgegeben wurden.                                                                                                                                                                                                                                     |
| Upload                  | Gibt an, dass die Synchronisation einen Uploadbefehl umfasst.                                                                                                                                                                                                                                                |
| Upload_Byte             | Verwendete Speichermenge im MobiLink-Server zum Speichern des Uploads. Nützliche Anzeige der Auswirkung einer Synchronisation auf den Serverspeicher.                                                                                                                                                        |
| Upload_Deadlocks        | Anzahl der Deadlocks in der konsolidierten Datenbank, die während des Uploads festgestellt wurden.                                                                                                                                                                                                           |
| Upload_Löscheilen       | Anzahl der Zeilen, die erfolgreich aus der konsolidierten Datenbank gelöscht wurden.                                                                                                                                                                                                                         |
| Upload_Fehler           | Anzahl der Fehler, die während des Uploads vorgekommen sind.                                                                                                                                                                                                                                                 |
| Upload_Einfügezeilen    | Anzahl der Zeilen, die erfolgreich in die konsolidierte Datenbank eingefügt wurden.                                                                                                                                                                                                                          |

| Eigenschaft                     | Beschreibung                                                                                                                                                |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Upload_Aktualisierungszeilen    | Anzahl der Zeilen, die erfolgreich in der konsolidierten Datenbank aktualisiert wurden.                                                                     |
| Upload_Warnungen                | Anzahl der Warnungen, die während des Uploads aufgetreten sind.                                                                                             |
| Benutzer                        | MobiLink-Benutzername.                                                                                                                                      |
| Version                         | Name der Synchronisationsversion                                                                                                                            |
| Auf_Download_Quittierung_warten | Die Zeit, die gewartet wird, bis der Download in die entfernte Datenbank übernommen und die Downloadbestätigung von der entfernten Datenbank gesendet wird. |

## Der Relay Server

Der Relay Server ermöglicht die sichere und ausgewogene Kommunikation zwischen mobilen Geräten und Backend-Servern, die über einen Webserver Daten austauschen. Zu den unterstützten Backend-Servern gehören MobiLink, Unwired Server, Afaria und Mobile Office. Der Relay Server stellt Folgendes zur Verfügung:

- Eine gemeinsame Architektur für mobile Geräte, die mit Backend-Servern kommunizieren.
- Ein Verfahren, das eine fehlertolerante Umgebung mit Lastverteilung für Backend-Server ermöglicht.
- Eine Möglichkeit, die Kommunikation zwischen mobilen Geräten und Backend-Servern so zu vereinfachen, dass sie problemlos in vorhandene Firewallkonfigurationen und Unternehmensrichtlinien integriert werden kann.

Weitere Hinweise zum Relay Server finden Sie unter [Relay Server](#).

## MobiLink - dateibasierte Downloads

Dateibasierte Downloads sind eine alternative Möglichkeit für den Download von Daten in entfernte SQL Anywhere-Datenbanken: Downloads können als Dateien verteilt werden, wodurch eine Offline-Verteilung von Synchronisationsänderungen möglich ist. Dies gestattet es Ihnen, eine Datei einmal zu erstellen und auf viele entfernte Datenbanken zu verteilen.

Bei dateibasierten Downloads können Sie Synchronisationsänderungen in einer Datei speichern, die Sie dann wie eine gewöhnliche Datei auf entfernte SQL Anywhere-Datenbanken übertragen. Zum Beispiel:

- Broadcast der Daten durch Satelliten-Multicast
- Aktualisierung mit Sybase Afaria anwenden
- Datei über E-Mail oder FTP an Benutzer senden

Sie wählen die Benutzer aus, die die Datei erhalten sollen. Bei dateibasierten Downloads wird die volle Synchronisationsintegrität bewahrt, einschließlich der Konflikterkennung und Konfliktlösung. Sie können gewährleisten, dass die Datei sicher ist, indem Sie eine Dritthersteller-Verschlüsselung auf sie anwenden.

## Verwendung

Dateibasierte Downloads sind sinnvoll, wenn sich große Datenmengen in der konsolidierten Datenbank ändern, aber die entfernte Datenbank die Daten nicht häufig oder überhaupt nicht aktualisiert. Beispiele hierfür sind Preis- und Produktlisten sowie Codetabellen.

Dateibasierte Downloads sind nicht sinnvoll, wenn die eingelesenen Daten häufig in der entfernten Datenbank aktualisiert werden oder wenn Sie häufig reine Upload-Synchronisationen durchführen. In diesen Fällen sind die entfernten Standorte möglicherweise aufgrund von Integritätsprüfungen, die bei der Anwendung von Download-Dateien ausgeführt werden, nicht in der Lage, Download-Dateien zu übernehmen.

Dateibasierte Downloads können derzeit nur für entfernte SQL Anywhere-Datenbanken verwendet werden.

## Reine Download-Publikationen

Normalerweise sollten Sie für Ihre dateibasierten Downloads eine reine Download-Publikation benutzen. Verwenden Sie nur dann eine reguläre Publikation, wenn Sie Uploads mit der gleichen Publikation durchführen müssen wie dateibasierte Downloads.

Wenn Sie eine reguläre Publikation verwenden, können dateibasierte Downloads nicht als einziges Mittel für die Aktualisierung von entfernten Datenbanken verwendet werden. In diesem Fall müssen Sie außerdem regelmäßig vollständige Synchronisationen oder reine Upload-Synchronisationen ausführen. Vollständige oder reine Upload-Synchronisationen sind erforderlich, um Log-Offsets zu verschieben und die Logdatei zu verwalten, die andernfalls zu groß wird und die Synchronisation verlangsamt. Eine vollständige Synchronisation kann auch erforderlich sein, um Fehler zu beheben.

## Siehe auch

- „Reine Download-Publikationen“ [[MobiLink - Clientadministration](#)]

## Dateibasierte Downloads einrichten

Nachfolgend erhalten Sie einen Überblick über die erforderlichen Aufgaben zur Einrichtung von dateibasierten Downloads. Dabei wird vorausgesetzt, dass die MobiLink-Synchronisation bereits eingerichtet ist.

1. Dateidefinitionsdatenbank erstellen

Siehe „[Dateidefinitionsdatenbank](#)“ auf Seite 215.

2. In der konsolidierten Datenbank Skripten mit einer neuen Skriptversion erstellen

Siehe „[Änderungen in der konsolidierten Datenbank](#)“ auf Seite 215.

### 3. Eine Download-Datei erstellen

Siehe „Erstellen einer Download-Datei“ auf Seite 216.

### 4. Die Download-Datei übernehmen

Siehe „Synchronisation neuer entfernter Datenbanken“ auf Seite 217.

## Weitere Ressourcen für die ersten Schritte

- „Beispiele zu datebasierten Downloads“

## Dateidefinitionsdatenbank

Um datebasierte Downloads einzurichten, erstellen Sie eine **Dateidefinitionsdatenbank**. Dies ist eine SQL Anywhere-Datenbank, die dieselben Synchronisationstabellen und Publikationen wie Ihre entfernte Datenbank enthält. Sie kann an einer beliebigen Position gespeichert werden. Diese Datenbank enthält keine Daten oder Statusinformationen. Sie braucht nicht gesichert oder verwaltet zu werden. Sie können sie sogar bei Bedarf löschen und neu erstellen.

Die Dateidefinitionsdatenbank muss Folgendes enthalten:

- Dieselben Publikationen wie die entfernten Datenbanken sowie die Tabellen und Spalten, die in der Publikation verwendet werden, die Fremdschlüsselbeziehungen und Integritätsregeln dieser Tabellen und Spalten sowie die Tabellen, die von diesen Fremdschlüsselbeziehungen benötigt werden.
- Einen MobiLink-Benutzernamen, der die Gruppe der entfernten Datenbanken identifiziert, die die Download-Datei übernehmen soll. Sie verwenden diesen Benutzernamen der MobiLink-Gruppen bei der Synchronisation, um die Gruppe der entfernten Datenbanken festzulegen.

## Änderungen in der konsolidierten Datenbank

Erstellen Sie in der konsolidierten Datenbank eine neue Skriptversion für Ihren datebasierten Download und implementieren Sie in dieser Version alle Skripten, die von Ihrem bestehenden Synchronisationssystem benötigt werden. Uploadskripten sind nicht erforderlich. Diese Skriptversion wird nur für datebasierte Downloads verwendet. In dieser Skriptversion verwenden alle Skripten, die MobiLink-Benutzernamen als Parameter haben, einen MobiLink-Benutzernamen, der sich auf eine Gruppe entfernter Datenbanken bezieht. Hierbei handelt es sich um den Benutzernamen, der in der Dateidefinitionsdatenbank festgelegt ist.

Für jede von Ihnen definierte Skriptversion implementieren Sie ein begin\_publication-Skript.

Für zeitstempelbasierte Downloads implementieren Sie für jede Skriptversion ein modify\_last\_download\_timestamp-Skript. Wie Sie dieses Skript implementieren hängt davon ab, wie viele Daten in den einzelnen Download-Dateien gesendet werden sollen. Eine Möglichkeit ist z.B., den frühesten Zeitpunkt zu verwenden, zu dem ein Benutzer aus der Gruppe zuletzt erfolgreich einen Download ausgeführt hat. Beachten Sie, dass der Parameter ml\_username, der an dieses Skript übergeben wird, der Gruppenname ist.

#### Hinweis

Es wird dringend empfohlen, dass Sie die auf dem MobiLink-Server die Option -dsd verwenden, wenn Sie dateibasierte Download-Dateien aus einer konsolidierten Microsoft SQL Server-Datenbank generieren. Wenn Sie nicht die Option -dsd verwenden, sind entfernte Datenbanken gelegentlich nicht in der Lage, eine Datei aus einem dateibasierten Download zu übernehmen, und melden einen Fehler wie den folgenden: Die letzte Downloadzeit für die Publikation <Publikation> ist <Zeitstempel>. Die vorletzte Downloadzeit der Download-Datei war <Zeitstempel>. Eine Download-Datei kann nicht übernommen werden, wenn ihre vorletzte Downloadzeit vor der letzten Downloadzeit der Publikation liegt.

Im Allgemeinen gilt: Die Häufigkeit, mit der eine entfernte Datenbank nicht in der Lage ist, eine Download-Datei zu übernehmen, ist proportional zu der Häufigkeit, mit der die entfernte Datenbank normale (verbundene) Synchronisationen ausführt, und zum Umfang an gleichzeitigen Aktivitäten in der konsolidierten Datenbank.

Siehe „mlsrv16-Option -dsd“ auf Seite 60.

#### Siehe auch

- „Skriptversionen“ auf Seite 254
- „begin\_publication (Verbindungsereignis)“ auf Seite 310
- „modify\_last\_download\_timestamp (Verbindungsereignis)“ auf Seite 392

## Erstellen einer Download-Datei

Die Download-Datei enthält die Daten, die synchronisiert werden sollen. Um die Download-Datei zu erstellen, richten Sie Ihre Dateidefinitionsdatenbank und die konsolidierte Datenbank wie oben beschrieben ein. Führen Sie dbmlsync mit der Option -bc aus und geben Sie einen Dateinamen mit der Erweiterung .df an. Beispiel:

```
dbmlsync -c "UID=DBA;PWD=sql;SERVER=fbd1_eng;DBF=fdef.db" -v+
-e "sv=filebased" -bc file1.df
```

Sie können auch Optionen angeben, wenn Sie die Download-Datei erstellen:

- **Option -be** Mit -be können Sie der Download-Datei eine Zeichenfolge hinzufügen, auf die in der entfernten Datenbank mit der gespeicherten Prozedur sp\_hook\_dbmlsync\_validate\_download\_file zugegriffen werden kann.
- **Option -bg** Verwenden Sie die Option -bg, um eine Download-Datei zu erstellen, die von entfernten Datenbanken verwendet werden kann, die noch nie synchronisiert wurden.

#### Siehe auch

- „dbmlsync-Option -be“ [MobiLink - Clientadministration]
- „sp\_hook\_dbmlsync\_validate\_download\_file“ [MobiLink - Clientadministration]
- „dbmlsync-Option -bg“ [MobiLink - Clientadministration]

## Synchronisation neuer entfernter Datenbanken

Wenn Sie eine Download-Datei in eine entfernte Datenbank übernehmen möchten, die noch nie mithilfe von MobiLink synchronisiert wurde, müssen Sie entweder vor dem Übernehmen der Download-Datei eine normale Synchronisation in der entfernten Datenbank ausführen oder beim Erstellen der Download-Datei die dbmlsync-Option -bg verwenden.

Bei der zeitstempelbasierten Synchronisation führen beide Aktionen zum Download eines ersten Snapshots der Daten. Sowohl bei der zeitstempel- als auch bei der snapshotbasierten Synchronisation setzt dieser Schritt die Generierungsnummer auf den Wert, der vom Skript begin\_publication in der konsolidierten Datenbank generiert wird.

### Normale Synchronisation ausführen

Sie können eine entfernte Datenbank auf den Empfang von Download-Dateien vorbereiten, indem Sie eine Synchronisation ausführen, die keine Download-Datei verwendet.

### Option -bg verwenden

Alternativ dazu können Sie auch eine Download-Datei mit der Option -bg erstellen, die für entfernte Datenbanken verwendet werden soll, die noch nicht synchronisiert wurden. Sie können diese Download-Ausgangsdatei anwenden, um die entfernte Datenbank auf die dateibasierte Synchronisation vorzubereiten.

- **Snapshot-Downloads** Wenn Sie Snapshot-Downloads ausführen, muss die Download-Ausgangsdatei lediglich die Generierungsnummer setzen. Sie können wählen, einen ersten Snapshot der Daten in diese Datei einzubeziehen, doch da jeder Snapshot-Download alle Daten enthält und nicht von früheren Downloads gesteuert wird, ist dies nicht erforderlich.

Bei Snapshot-Downloads ist die Verwendung der Option -bg einfach. Sie geben lediglich -bg in der dbmlsync-Befehlszeile ein, wenn Sie die Download-Datei erstellen. Sie können für die Erstellung der Download-Ausgangsdatei und für nachfolgende Download-Dateien dieselbe Skriptversion verwenden.

- **Zeitstempelbasierte Downloads** Wenn Sie zeitstempelbasierte Downloads ausführen, muss der Ausgangsdownload die Generierungsnummer in der entfernten Datenbank festlegen und einen Snapshot der Daten enthalten. Bei zeitstempelbasierten Downloads baut jeder Download auf dem jeweils vorherigen auf. Jede Download-Datei enthält einen Zeitstempel des letzten Downloads. Alle Zeilen, die in der konsolidierten Datenbank nach dem letzten Download-Zeitstempel der Datei geändert wurden, sind in der Datei enthalten. Damit eine Datei übernommen werden kann, muss eine entfernte Datenbank bereits alle Änderungen empfangen haben, die vor dem letzten Download-Zeitstempel der Datei vorgenommen wurden. Um dies zu bestätigen, wird überprüft, ob der Zeitstempel für den letzten Download größer oder gleich dem letzten Download-Zeitstempel der Datenbank ist (die Zeit, bis zu der die entfernte Datenbank alle Änderungen aus der konsolidierten Datenbank empfangen hat).

Damit eine entfernte Datenbank ihre erste normale Download-Datei übernehmen kann, muss sie alle Daten empfangen, die vor dem letzten Download-Zeitstempel und nach dem 1. Januar 1900 Daten geändert wurden. Die erste mit der Option -bg erstellte Download-Datei muss diese Daten enthalten. Die einfachste Möglichkeit, diese Daten auszuwählen, ist die Erstellung einer separaten Skriptversion, die dieselben download\_cursors wie die normale dateibasierte Synchronisationsskriptversion verwendet, aber kein modify\_last\_download\_timestamp-Skript hat. Wenn kein

modify\_last\_download\_timestamp-Skript festgelegt wurde, erhält der Zeitstempel für den letzten Download für einen dateibasierten Download den Wert 1. Januar 1900.

Wenn Sie Download-Dateien, die mit der Option -bg erstellt wurden, auf entfernte Datenbanken übernehmen, die bereits synchronisiert wurden, bewirkt die Option -bg, dass die Generierungsnummern auf der entfernten Datenbank mit dem Wert in der konsolidierten Datenbank zum Zeitpunkt der Erstellung der Download-Datei aktualisiert werden. Dadurch wird der Zweck der Generierungsnummern aufgehoben, der darin besteht zu verhindern, dass weitere dateibasierte Downloads übernommen werden, bis ein Upload durchgeführt wird, wie z.B. beim Wiederherstellen einer verlorenen oder beschädigten konsolidierten Datenbank.

#### Siehe auch

- „MobiLink-Generierungsnummern“ auf Seite 220
- „dbmsync-Option -bg“ [*MobiLink - Clientadministration*]

## Validierungsprüfungen

Vor der Übernahme einer Download-Datei in eine entfernte Datenbank führt dbmsync eine Reihe von Aufgaben aus, um zu gewährleisten, dass die Synchronisation gültig ist.

- dbmsync überprüft die Download-Datei, um sicherzustellen, dass die Dateidefinitionsdatenbank, die für ihre Erstellung verwendet wurde, über Folgendes verfügt:
  - Dieselbe Publikation wie die entfernte Datenbank.
  - Dieselben Tabellen und Spalten, die in der Publikation verwendet wurden.
  - Dieselben Fremdschlüsselbeziehungen und Integritätsregeln wie diese Tabellen und Spalten.
- dbmsync überprüft, ob Daten in der Publikation enthalten sind, die nicht von der entfernten Datenbank ausgelesen wurden. Ist dies der Fall, wird die Download-Datei nicht übernommen, da dies zum Verlust von Daten führen kann, deren Upload noch nicht abgeschlossen wurde.
- dbmsync überprüft den Zeitstempel des letzten und des vorletzten Downloads sowie den Zeitpunkt der Erstellung der Download-Datei, um Folgendes sicherzustellen:
  - Es werden keine neueren Daten auf der entfernten Datenbank durch ältere Daten aus der Download-Datei überschrieben.
  - Eine Download-Datei wird nicht übernommen, wenn dies dazu führen würde, dass die entfernte Datenbank einige Änderungen der konsolidierten Datenbank nicht berücksichtigt. Diese Situation kann eintreten, wenn die entfernte Datenbank frühere dateibasierte Downloads nicht übernommen hat.

Siehe „Automatische Validierung“ auf Seite 219.

- Optional überprüft dbmsync die Generierungsnummer in der entfernten Datenbank, um sicherzustellen, dass sie mit der Generierungsnummer in der Download-Datei übereinstimmt.

Siehe „MobiLink-Generierungsnummern“ auf Seite 220.

- Optional können Sie mit der gespeicherten Prozedur `sp_hook_dbmlsync_validate_download_file` eine angepasste Validierungslogik erstellen.

Weitere Hinweise finden Sie unter „Angepasste Validierung“ auf Seite 221.

## Automatische Validierung

Vor der Übernahme einer Download-Datei führt dbmlsync spezielle Prüfungen des letzten und des vorletzten Download-Zeitstempels, des Erstellungszeitpunkts der Download-Datei und des Transaktionslogs durch.

### Letzter und vorletzter Download-Zeitstempel

Jede Download-Datei enthält alle einzulesenden Änderungen, die in der konsolidierten Datenbank zwischen dem letzten und vorletzten Download-Zeitstempel der Datei durchgeführt wurden. Die Zeit der konsolidierten Datenbank wird für beide Zeitwerte verwendet. Standardmäßig ist der letzte Download-Zeitpunkt der Datei der 1. Januar 1900 um 12:00 Uhr und der nächste Zeitstempel für den letzten Download ist die Uhrzeit, zu der die Download-Datei erstellt wurde. Diese Standardwerte können durch die Implementierung der Skripten `generate_next_last_download_timestamp`, `modify_last_download_timestamp` und `modify_next_last_download_timestamp` in der konsolidierten Datenbank aufgehoben werden.

Ein entfernter Standort kann eine Download-Datei nur dann übernehmen, wenn der letzte Download-Zeitstempel der Datei kleiner oder gleich dem letzten Download-Zeitstempel der entfernten Datenbank ist. Damit wird sichergestellt, dass eine entfernte Datenbank alle Vorgänge aus der konsolidierten Datenbank verfolgt. Wenn ein dateibasierter Download aufgrund dieser Prüfung fehlschlägt, hat die entfernte Datenbank gewöhnlich eine oder mehrere Download-Dateien nicht übernommen. Diese Situation kann behoben werden, indem die fehlenden Download-Dateien übernommen werden oder eine vollständige Synchronisation bzw. eine Synchronisation durch reinen Download ausgeführt wird.

Außerdem kann ein entfernter Standort eine Download-Datei nur dann übernehmen, wenn der vorletzte Download-Zeitstempel der Datei größer oder gleich dem letzten Download-Zeitstempel der entfernten Datenbank ist. Der letzte Download-Zeitstempel der entfernten Datenbank ist die Uhrzeit (auf der konsolidierten Datenbank), bis zu der die entfernte Datenbank alle einzulesenden Änderungen empfangen hat. Die letzte Downloadzeit der entfernten Datenbank wird jeweils aktualisiert, wenn die entfernte Datenbank einen Download erfolgreich übernimmt (normal oder dateibasiert). Diese Prüfung stellt sicher, dass eine Download-Datei nicht übernommen wird, wenn bereits neuere Daten eingelesen wurden. Dies geschieht häufig, wenn Download-Dateien in einer falschen Reihenfolge übernommen werden. Beispiel: Die Download-Datei *F1.df* wird erstellt und später wird eine weitere Datei mit dem Namen *F2.df* erstellt. Diese Prüfung stellt sicher, dass *F1.df* nicht nach *F2.df* übernommen werden kann, da dadurch neuere Daten aus *F2.df* mit älteren Daten aus *F1.df* überschrieben werden können.

Wenn ein dateibasierter Download aufgrund des Zeitstempels für den vorletzten Download fehlschlägt, genügt es, die Datei zu löschen. Die Synchronisation ist erfolgreich, sobald eine neue Datei empfangen wird.

### Erstellungszeit

Die Erstellungszeit der Download-Datei gibt den Zeitpunkt an, zu dem in der konsolidierten Datenbank die Erstellung der Datei begonnen hat. Eine Download-Datei kann nur übernommen werden, wenn die

Erstellungszeit größer als die letzte Uploadzeit der entfernten Datenbank ist. Die letzte Uploadzeit der entfernten Datenbank ist der Zeitpunkt in der konsolidierten Datenbank, zu dem der letzte erfolgreiche Upload in der entfernten Datenbank festgeschrieben wurde. Diese Prüfung stellt sicher, dass Daten, die nach der Erstellung des Downloads ausgelesen wurden (und neuer sind als der Download), nicht von älteren Daten in der Download-Datei überschrieben werden.

Wenn eine Download-Datei aufgrund dieser Prüfung abgelehnt wird, ist keine Aktion erforderlich. Der entfernte Standort sollte in der Lage sein, die nächste Download-Datei zu übernehmen.

Wenn ein Upload fehlschlägt, weil dbmlsync nach dem Senden eines Uploads an den MobiLink-Server keine Quittierung erhalten hat, ist vielleicht die Zeit des letzten Uploads in der entfernten Datenbank falsch. In diesem Fall kann die Erstellungszeit nicht geprüft werden und die entfernte Datenbank ist nicht in der Lage, Download-Dateien zu übernehmen, bis eine normale Synchronisation ausgeführt wurde.

### Transaktionslog

Vor der Übernahme einer Download-Datei überprüft dbmlsync das Transaktionslog der entfernten Datenbank und erstellt eine Liste aller Änderungen, die heraufgeladen werden müssen. Dbmlsync übernimmt eine Download-Datei nur, wenn sie keine Vorgänge enthält, die Zeilen mit Änderungen betreffen, die heraufgeladen werden müssen.

## MobiLink-Generierungsnummern

Generationsnummern bieten einen Mechanismus, der entfernte Datenbanken zwingt, einen Datenupload durchzuführen, bevor weitere Download-Dateien übernommen werden. Dies ist besonders nützlich, wenn ein Problem in der konsolidierten Datenbank zu einem Datenverlust geführt hat und Sie verlorene Daten von den entfernten Datenbanken wiederherstellen müssen.

In der entfernten Datenbank wird für jede Subskription automatisch eine separate Generierungsnummer verwaltet. In der konsolidierten Datenbank wird die Generierungsnummer für die einzelnen Subskriptionen vom Skript `begin_publication` ermittelt. Bei jedem erfolgreichen Upload einer entfernten Datenbank wird die Generierungsnummer der entfernten Datenbank mit dem Wert aktualisiert, der vom Skript `begin_publication` in der konsolidierten Datenbank festgelegt wird.

Bei jeder Erstellung einer Download-Datei wird die vom Skript `begin_publication` festgelegte Generierungsnummer in der Download-Datei gespeichert. Ein entfernter Standort übernimmt eine Download-Datei nur, wenn die Generierungsnummer in der Datei mit der Generierungsnummer in der entfernten Datenbank übereinstimmt.

#### Hinweis

Wenn sich die vom Skript `begin_publication` für einen dateibasierten Download erstellte Generierungsnummer ändert, muss die entfernte Datenbank einen erfolgreichen Upload ausführen, bevor eine neue Download-Datei übernommen werden kann.

Mit der gespeicherten Prozedur `sp_hook_dbmlsync_validate_download_file` kann die Standardprüfung der Generierungsnummer aufgehoben werden.

Weitere Hinweise zur Verwaltung von MobiLink-Generierungsnummern finden Sie unter:

- „[begin\\_publication \(Verbindungsereignis\)](#)“ auf Seite 310
- „[end\\_publication \(Verbindungsereignis\)](#)“ auf Seite 350
- „[sp\\_hook\\_dbmlsync\\_validate\\_download\\_file](#)“ [*MobiLink - Clientadministration*]

## Angepasste Validierung

Sie können eine angepasste Validierungslogik erstellen, um festzulegen, ob eine Download-Datei in eine entfernte Datenbank übernommen werden soll. Hierzu verwenden Sie die gespeicherte Prozedur `sp_hook_dbmlsync_validate_download_file`. Mit dieser gespeicherten Prozedur können Sie sowohl eine Download-Datei ablehnen als auch die Standardprüfung der Generierungsnummer aufheben.

Um eine Zeichenfolge in die Datei einzubetten, verwenden Sie die `dbmlsync`-Option `-be`. Bei der Erstellung der Download-Datei verwenden Sie die Option `-be` für die Dateidefinitionsdatenbank. Diese Zeichenfolge wird über die Tabelle `#hook_dict` an `sp_hook_dbmlsync_validate_download_file` übergeben und kann in Ihrer Validierungslogik verwendet werden.

### Siehe auch

- „[sp\\_hook\\_dbmlsync\\_validate\\_download\\_file](#)“ [*MobiLink - Clientadministration*]

## Beispiele zu dateibasierten Downloads

Dieser Abschnitt enthält zwei Beispiele. Beide Beispiele richten jeweils eine dateibasierte Download-Synchronisation ein, wobei eine konsolidierte Datenbank mit nur einer Tabelle verwendet wird. Das erste Beispiel ist ein einfacher Snapshot, das zweite ist ein etwas komplexeres zeitstempelbasiertes Beispiel.

## Snapshot-Beispiel

Dieses Beispiel implementiert den dateibasierten Download für die Snapshot-Synchronisation. Es werden die drei Datenbanken eingerichtet, die für den dateibasierten Download erforderlich sind. Danach wird erläutert, wie die Daten übertragen werden. Sie können entweder das Beispiel einfach durchlesen oder den Text ausschneiden und einfügen, um das Beispiel auszuführen.

### Datenbanken für das Beispiel erstellen

Die folgenden Befehle erstellen die drei in dem Beispiel verwendeten Datenbanken, nämlich eine konsolidierte Datenbank, eine entfernte Datenbank und eine Dateidefinitionsdatenbank.

```
dbinit -dba DBA,sql scons.db
dbinit -dba DBA,sql sremote.db
dbinit -dba DBA,sql sfdef.db
```

Die folgenden Befehle starten die drei Datenbanken und erstellen einen Datenquellennamen, den MobiLink verwendet, um sich mit der konsolidierten Datenbank zu verbinden.

```
dbeng16 -n sfdef_eng sfdef.db
dbeng16 -n scons_eng scons.db
```

```
dbeng16 -n sremote_eng sremote.db
dbdsn -y -w fbd_demo -c
"SERVER=scons_eng;DBF=scons.db;UID=DBA;PWD=sql;ASTART=off;ASTOP=off"
```

Öffnen Sie Interactive SQL, stellen Sie eine Verbindung mit *scons.db* her, und führen Sie das MobiLink-Setupskript aus. Beispiel:

```
read "C:\Program Files\SQL Anywhere 16\MobiLink\setup\syncsa.sql"
```

Starten Sie den MobiLink-Server.

```
start mlsrv16 -v+ -c "DSN=fbd_demo" -zu+ -ot scons.txt
```

### Die konsolidierte Snapshot-Beispieldatenbank einrichten

In diesem Beispiel enthält die konsolidierte Datenbank eine einzige Tabelle namens T1. Nach der Verbindungserstellung zur konsolidierten Datenbank können Sie folgende SQL-Anweisung ausführen, um Tabelle T1 zu erstellen:

```
CREATE TABLE T1 (
  pk INTEGER PRIMARY KEY,
  c1 INTEGER
);
```

Der folgende Code erstellt eine sogenannte dateibasierte Skriptversion und dann ein Download-Skript für diese Skriptversion.

```
CALL ml_add_table_script( 'filebased',
  'T1', 'download_cursor',
  'SELECT pk, c1 FROM T1' );
```

Der folgende Code erstellt eine sogenannte normale Skriptversion und dann Upload- und Download-Skripten für diese Skriptversion.

```
CALL ml_add_table_script ( 'normal', 'T1',
  'upload_insert',
  'INSERT INTO T1 VALUES ({ml r.pk}, {ml r.c1})');

CALL ml_add_table_script( 'normal', 'T1',
  'upload_update',
  'UPDATE T1 SET c1 = {ml r.c1} WHERE pk = {ml r.pk} ' );

CALL ml_add_table_script( 'normal', 'T1',
  'upload_delete',
  'DELETE FROM T1 WHERE pk = {ml r.pk}' );

CALL ml_add_table_script( 'normal', 'T1',
  'download_cursor',
  'SELECT pk, c1 FROM T1' );

CALL ml_add_table_script( 'normal', 'T1',
  'download_delete_cursor',
  '--{ml_ignore}' );

COMMIT;
```

Der folgende Befehl erstellt die gespeicherte Prozedur *begin\_pub* und legt fest, dass *begin\_pub* das *begin\_publication*-Skript für die normale und die dateibasierte Skriptversion ist:

```
CREATE PROCEDURE begin_pub (
  INOUT generation_num integer,
```

```

        IN      username      varchar(128),
        IN      pubname       varchar(128) )
BEGIN
    SET generation_num=1;
END;

CALL ml_add_connection_script(
    'filebased',
    'begin_publication',
    '{ call begin_pub(
        {ml s.generation_number},
        {ml s.username},
        {ml s.publication_name} ) }' );

CALL ml_add_connection_script(
    'normal',
    'begin_publication',
    '{ call begin_pub(
        {ml s.generation_number},
        {ml s.username},
        {ml s.publication_name} ) }' );

```

### Die entfernte Snapshot-Beispieldatenbank erstellen

In diesem Beispiel enthält die entfernte Datenbank eine einzige Tabelle namens T1. Stellen Sie eine Verbindung zur entfernten Datenbank her und führen Sie die folgende SQL-Anweisung aus, um die Tabelle T1, eine Publikation namens P1 und einen Benutzer namens U1 zu erstellen. Die SQL-Anweisung erstellt außerdem eine Subskription für U1 zu P1.

```

CREATE TABLE T1 (
    pk INTEGER PRIMARY KEY,
    c1 INTEGER
);

CREATE PUBLICATION P1 (
    TABLE T1
);

CREATE SYNCHRONIZATION USER U1;

CREATE SYNCHRONIZATION SUBSCRIPTION
TO P1
FOR U1;

```

Der folgende Code erstellt die Einstiegsprozedur `sp_hook_dbmlsync_validate_download_file`, um benutzerdefinierte Validierungslogik in der entfernten Datenbank zu erstellen:

```

CREATE PROCEDURE sp_hook_dbmlsync_validate_download_file()
BEGIN
    DECLARE udata varchar(256);
    SELECT value
    INTO udata
    FROM #hook_dict
    WHERE name = 'user data';
    IF udata <> 'ok' THEN
        UPDATE #hook_dict
        SET value = 'FALSE'
        WHERE name = 'apply file';
    END IF;
END

```

## Dateidefinitions-Beispieldatenbank mit Snapshot erstellen

In MobiLink-Systemen, die dateibasierte Downloads verwenden, ist eine Dateidefinitionsdatenbank erforderlich. Diese Datenbank hat dasselbe Schema wie die entfernten Datenbanken, die durch dateibasierte Downloads aktualisiert werden, und enthält keine Daten oder Statusinformationen. Die Dateidefinitionsdatenbank dient nur zur Festlegung der Struktur der Daten, die in die Download-Datei einbezogen werden sollen. Eine Dateidefinitionsdatenbank kann für mehrere Gruppen von entfernten Datenbanken verwendet werden, die jeweils von ihrem eigenen MobiLink-Gruppen-Benutzernamen festgelegt werden.

Der folgende Code definiert die Dateidefinitionsdatenbank für dieses Beispiel. Er erstellt ein Schema, das identisch mit der entfernten Datenbank ist, und außerdem Folgendes:

- Eine Publikation namens P1, die alle Zeilen der Tabelle T1 veröffentlicht. Derselbe Publikationsname muss in der Dateidefinitionsdatenbank und den entfernten Datenbanken verwendet werden.
- Einen MobiLink-Benutzer namens G1. Dieser Benutzer repräsentiert alle entfernten Datenbanken, die beim dateibasierten Download aktualisiert werden sollen.
- Eine Subskription der Publikation.

Vor der Ausführung des Codes müssen Sie sich mit sfdef.db verbinden.

```
CREATE TABLE T1 (  
    pk INTEGER PRIMARY KEY,  
    c1 INTEGER  
);  
  
CREATE PUBLICATION P1 (  
    TABLE T1  
);  
  
CREATE SYNCHRONIZATION USER G1;  
  
CREATE SYNCHRONIZATION SUBSCRIPTION  
TO P1  
FOR G1;
```

## Erste Synchronisation vorbereiten

Sie müssen Ihre neue entfernte Datenbank für die Übernahme einer Download-Datei vorbereiten, indem Sie entweder eine normale Synchronisation ausführen oder die Download-Datei mit der dbmlsync-Option -bg erstellen. Dieses Beispiel veranschaulicht, wie Sie Ihre neue entfernte Datenbank initialisieren, indem Sie eine normale Synchronisation ausführen.

Sie können eine erste Synchronisation der entfernten Datenbank mit der als normal bezeichneten Skriptversion ausführen, die weiter oben erstellt wurde:

```
dbmlsync -c "UID=DBA;PWD=sql;SERVER=sremote_eng;DBF=sremote.db" -v+ -e  
"sv=normal"
```

## Dateibasierten Snapshot-Download veranschaulichen

Stellen Sie eine Verbindung zur konsolidierten Datenbank her und fügen Sie Daten ein, die durch einen dateibasierten Download synchronisiert werden, wie im folgenden Beispiel:

```
INSERT INTO T1 VALUES( 1, 1 );
INSERT INTO T1 VALUES( 2, 4 );
INSERT INTO T1 VALUES( 3, 9 );
COMMIT;
```

Der folgende Befehl muss auf dem Computer ausgeführt werden, auf dem sich die Dateidefinitionsdatenbank befindet. Er führt folgende Aktionen aus:

- Die dbmlsync-Option -bc erstellt die Download-Datei mit dem Namen *file1.df*.
- Die Option -be bezieht die Zeichenfolge "ok" in die Download-Datei ein, auf die die gespeicherte Einstiegsprozedur sp\_dbmlsync\_validate\_download\_file zugreifen kann.

```
dbmlsync -c "UID=DBA;PWD=sql;SERVER=sfdef_eng;DBF=sfdef.db" -v+ -e
"sv=filebased" -bc file1.df -be ok -ot fdef.txt
```

Um die Download-Datei zu übernehmen, führen Sie dbmlsync mit der Option -ba in der entfernten Datenbank aus, wobei Sie den Namen der Download-Datei übergeben, die übernommen werden soll:

```
dbmlsync -c "UID=DBA;PWD=sql;SERVER=sremote_eng; DBF=sremote.db" -v+ -ba
file1.df -ot remote.txt
```

Die Änderungen werden nun in die entfernte Datenbank übernommen. Öffnen Sie Interactive SQL, verbinden Sie sich mit der entfernten Datenbank und führen Sie die folgende SQL-Anweisung aus, um zu überprüfen, ob die entfernte Datenbank die Daten enthält:

```
SELECT * FROM T1
```

## Snapshot-Beispiel aufräumen

Die folgenden Befehle stoppen alle drei Datenbankserver und löschen die Dateien.

```
del file1.df
mlstop -h -w
dbstop -y -c "SERVER=sfdef_eng; UID=DBA; PWD=sql"
dbstop -y -c "SERVER=scons_eng; UID=DBA; PWD=sql"
dbstop -y -c "SERVER=sremote_eng; UID=DBA; PWD=sql"
dberase -y sfdef.db
dberase -y scons.db
dberase -y sremote.db
```

## Zeitstempelbasierte Beispiele

Dieses Beispiel implementiert den dateibasierten Download für die zeitstempelbasierte Synchronisation. Es werden die drei Datenbanken eingerichtet, und anschließend wird erläutert, wie die Daten per Datei heruntergeladen werden. Sie können entweder das Beispiel einfach durchlesen oder den Text ausschneiden und einfügen, um das Beispiel auszuführen.

### Datenbanken für das Beispiel erstellen

Die folgenden Befehle erstellen die drei in dem Beispiel verwendeten Datenbanken, nämlich eine konsolidierte Datenbank, eine entfernte Datenbank und eine Dateidefinitionsdatenbank.

```
dbinit -dba DBA,sql tcons.db
dbinit -dba DBA,sql tremote.db
dbinit -dba DBA,sql tfdef.db
```

Die folgenden Befehle starten die drei Datenbanken und erstellen einen Datenquellennamen, den MobiLink verwendet, um sich mit der konsolidierten Datenbank zu verbinden.

```
dbeng16 -n tfdef_eng tfdef.db
dbeng16 -n tcons_eng tcons.db
dbeng16 -n tremote_eng tremote.db
dbdsn -y -w tfbd_demo -c
"SERVER=tcons_eng;DBF=tcons.db;UID=DBA;PWD=sql;START=off;ASTOP=off"
```

Öffnen Sie Interactive SQL, stellen Sie eine Verbindung mit *tcons.db* her, und führen Sie das MobiLink-Setupskript aus. Beispiel:

```
read "C:\Program Files\SQL Anywhere 16\MobiLink\setup\syncsa.sql"
```

Starten Sie den MobiLink-Server.

```
start mlsrv16 -v+ -c "DSN=tfbd_demo" -zu+ -ot tcons.txt
```

### Die konsolidierte Zeitstempel-Beispieldatenbank einrichten

In diesem Beispiel enthält die konsolidierte Datenbank eine einzige Tabelle namens T1. Nach der Verbindungserstellung zur konsolidierten Datenbank können Sie folgenden Code ausführen, um Tabelle T1 zu erstellen:

```
CREATE TABLE T1 (
  pk  INTEGER PRIMARY KEY,
  c1  INTEGER,
  last_modified TIMESTAMP DEFAULT TIMESTAMP
);
```

Der folgende Code legt eine Skriptversion namens normal; mit einer minimalen Anzahl von Skripten fest. Diese Skriptversion wird für Synchronisationen verwendet, die **keinen** dateibasierten Download verwenden.

```
CALL ml_add_table_script( 'normal', 'T1',
  'upload_insert',
  'INSERT INTO T1( pk, c1) VALUES( {ml r.pk}, {ml r.c1} )' );

CALL ml_add_table_script( 'normal', 'T1',
  'upload_update',
  'UPDATE T1 SET c1 = {ml r.c1} WHERE pk = {ml r.pk}' );

CALL ml_add_table_script( 'normal', 'T1',
  'upload_delete',
  'DELETE FROM T1 WHERE pk = {ml r.pk}' );

CALL ml_add_table_script( 'normal', 'T1',
  'download_cursor',
  'SELECT pk, c1 FROM T1
  WHERE last_modified >= {ml s.last_table_download}' );
```

Der folgende Code setzt die Generierungsnummer für alle Subskriptionen auf 1. Generierungsnummern können nützlich sein, wenn Ihre konsolidierte Datenbank verloren geht oder beschädigt wird und Sie einen Upload erzwingen müssen.

```
CREATE PROCEDURE begin_pub (
  INOUT generation_num integer,
  IN    username         varchar(128),
  IN    pubname          varchar(128) )
```

```

BEGIN
  SET generation_num = 1;
END;

CALL ml_add_connection_script( 'normal',
  'begin_publication',
  '{ call begin_pub(
    {ml s.generation_number},
    {ml s.username},
    {ml s.publication_name},
    {ml s.last_publication_upload},
    {ml s.last_publication_download} ) }' );

COMMIT;

```

Der folgende Code legt die Skriptversion namens "filebased" fest. Diese Skriptversion wird für die Erstellung dateibasierter Downloads verwendet.

```

CALL ml_add_connection_script( 'filebased',
  'begin_publication',
  '{ call begin_pub(
    {ml s.generation_number},
    {ml s.username},
    {ml s.publication_name} ) }' );

CALL ml_add_table_script( 'filebased', 'T1',
  'download_cursor',
  'SELECT pk, c1 FROM T1
   WHERE last_modified >= {ml s.last_table_download}' );

```

Der folgende Code legt die letzte Downloadzeit fest, sodass alle Änderungen, die innerhalb der letzten fünf Tage vorgenommen wurden, in Download-Dateien einbezogen werden. Jede entfernte Datenbank, die keine der in den letzten fünf Tagen erstellten Download-Dateien übernommen hat, muss eine normale Synchronisation ausführen, bevor sie weitere dateibasierte Downloads übernehmen kann.

```

CREATE PROCEDURE ModifyLastDownloadTimestamp(
  INOUT last_download_timestamp TIMESTAMP,
  IN    ml_username             VARCHAR(128) )
BEGIN
  SELECT dateadd( day, -5, CURRENT_TIMESTAMP )
  INTO last_download_timestamp;
END;

CALL ml_add_connection_script( 'filebased',
  'modify_last_download_timestamp',
  'CALL ModifyLastDownloadTimestamp(
    {ml s.last_download}, {ml s.username} )' );

COMMIT;

```

### Die entfernte zeitstempelbasierte Beispieldatenbank erstellen

In diesem Beispiel enthält die entfernte Datenbank eine einzige Tabelle namens T1. Wenn Sie eine Verbindung zur entfernten Datenbank hergestellt haben, führen Sie den folgenden Code aus, um die Tabelle T1, eine Publikation namens P1 und einen Benutzer namens U1 zu erstellen. Der Code erstellt außerdem für U1 eine Subskription von P1.

```

CREATE TABLE T1 (
  pk INTEGER PRIMARY KEY,
  c1 INTEGER

```

```
);

CREATE PUBLICATION P1 (
    TABLE T1
);

CREATE SYNCHRONIZATION USER U1;

CREATE SYNCHRONIZATION SUBSCRIPTION
TO P1
FOR U1;
```

Der folgende Code definiert die gespeicherte Prozedur `sp_hook_dbmlsync_validate_download_file`. Diese gespeicherte Prozedur verhindert, dass Download-Dateien übernommen werden, in denen die Zeichenfolge OK nicht enthalten ist.

```
CREATE PROCEDURE sp_hook_dbmlsync_validate_download_file()
BEGIN
    DECLARE udata varchar(256);

    SELECT value
    INTO udata
    FROM #hook_dict
    WHERE name = 'user data';

    IF udata <> 'ok' THEN
        UPDATE #hook_dict
        SET value = 'FALSE'
        WHERE name = 'apply file';
    END IF;
END
```

### Dateidefinitions-Beispieldatenbank mit Zeitstempel erstellen

Der folgende Code definiert die Dateidefinitionsdatenbank für das Zeitstempelbeispiel. Er erstellt eine Tabelle, eine Publikation, einen Benutzer und eine Publikationssubskription für den Benutzer.

```
CREATE TABLE T1 (
    pk INTEGER PRIMARY KEY,
    c1 INTEGER
);

CREATE PUBLICATION P1 (
    TABLE T1
);

CREATE SYNCHRONIZATION USER G1;

CREATE SYNCHRONIZATION SUBSCRIPTION
TO P1
FOR G1;
```

### Erste Synchronisation vorbereiten

Sie müssen Ihre neue entfernte Datenbank für die Übernahme einer Download-Datei vorbereiten, indem Sie entweder eine normale Synchronisation ausführen oder die Download-Datei mit der `dbmlsync`-Option `-bg` erstellen. Dieses Beispiel veranschaulicht die Verwendung von `-bg`.

Der folgende Code definiert eine Skriptversion namens `filebased_init` für die konsolidierte Datenbank. Diese Skriptversion hat ein einziges `begin_publication`-Skript.

```
CALL ml_add_table_script(
  'filebased_init', 'T1', 'download_cursor',
  'SELECT pk, c1 FROM T1' );

CALL ml_add_connection_script(
  'filebased_init',
  'begin_publication',
  '{ call begin_pub(
    {ml s.generation_number},
    {ml s.username},
    {ml s.publication_name} ) }' );

COMMIT;
```

Die beiden folgenden Befehlszeilen erstellen eine Download-Ausgangsdatei, die die Skriptversion filebased\_init und die Option -bg verwendet.

```
dbmlsync -c "UID=DBA;PWD=sql;SERVER=tfdef_eng;DBF=tfdef.db"
-v+ -e "sv=filebased_init" -bc tfile1.df -be ok -bg
-ot tfdef1.txt

dbmlsync -c "UID=DBA;PWD=sql;SERVER=tremote_eng;DBF=tremote.db"
-v+ -ba tfile1.df -ot tremote.txt
```

### Dateibasierten Zeitstempel-Download veranschaulichen

Stellen Sie eine Verbindung zur konsolidierten Datenbank her und fügen Sie Daten ein, die durch einen dateibasierten Download synchronisiert werden, wie im folgenden Beispiel:

```
INSERT INTO T1(pk, c1) VALUES( 1, 1 );
INSERT INTO T1(pk, c1) VALUES( 2, 4 );
INSERT INTO T1(pk, c1) VALUES( 3, 9 );
commit;
```

Die folgende Befehlszeile erstellt eine Download-Datei, die die neuen Daten enthält.

```
dbmlsync -c
"UID=DBA;PWD=sql;SERVER=tfdef_eng;DBF=tfdef.db"
-v+ -e "sv=filebased" -bc tfile2.df -be ok -ot tfdef2.txt
```

Die folgende Befehlszeile übernimmt eine Download-Datei in die entfernte Datenbank.

```
dbmlsync -c "UID=DBA;PWD=sql;SERVER=tremote_eng;DBF=tremote.db"
-v+ -ba tfile2.df -ot tfdef3.txt
```

Die Änderungen werden nun in die entfernte Datenbank übernommen. Öffnen Sie Interactive SQL, verbinden Sie sich mit der entfernten Datenbank und führen Sie die folgende SQL-Anweisung aus, um zu überprüfen, ob die entfernte Datenbank die Daten enthält:

```
SELECT * FROM T1
```

### Zeitstempelbeispiel aufräumen

Die folgenden Befehle stoppen alle drei Datenbankserver und löschen dann die Dateien.

```
del tfile1.df
mlstop -h -w
dbstop -y -c "SERVER=tfdef_eng; UID=DBA; PWD=sql"
dbstop -y -c "SERVER=tcons_eng; UID=DBA; PWD=sql"
dbstop -y -c "SERVER=tremote_eng; UID=DBA; PWD=sql"
```

```
dberase -y tfdef.db  
dberase -y tcons.db  
dberase -y tremote.db
```

---

# MobiLink-Ereignisse

Dieser Abschnitt beschreibt, wie Sie Skripten für MobiLink-Ereignisse schreiben.

## Schreiben von Synchronisationsskripten

Der Synchronisationsprozess wird mit Synchronisationsskripten gesteuert, die in MobiLink-Systemtabellen der konsolidierten Datenbank gespeichert oder referenziert werden. Sie können Skripten in SQL, Java oder .NET schreiben.

Die **MobiLink-Synchronisationslogik** wird mit Synchronisationsskripten angegeben. Die Skripten definieren Folgendes:

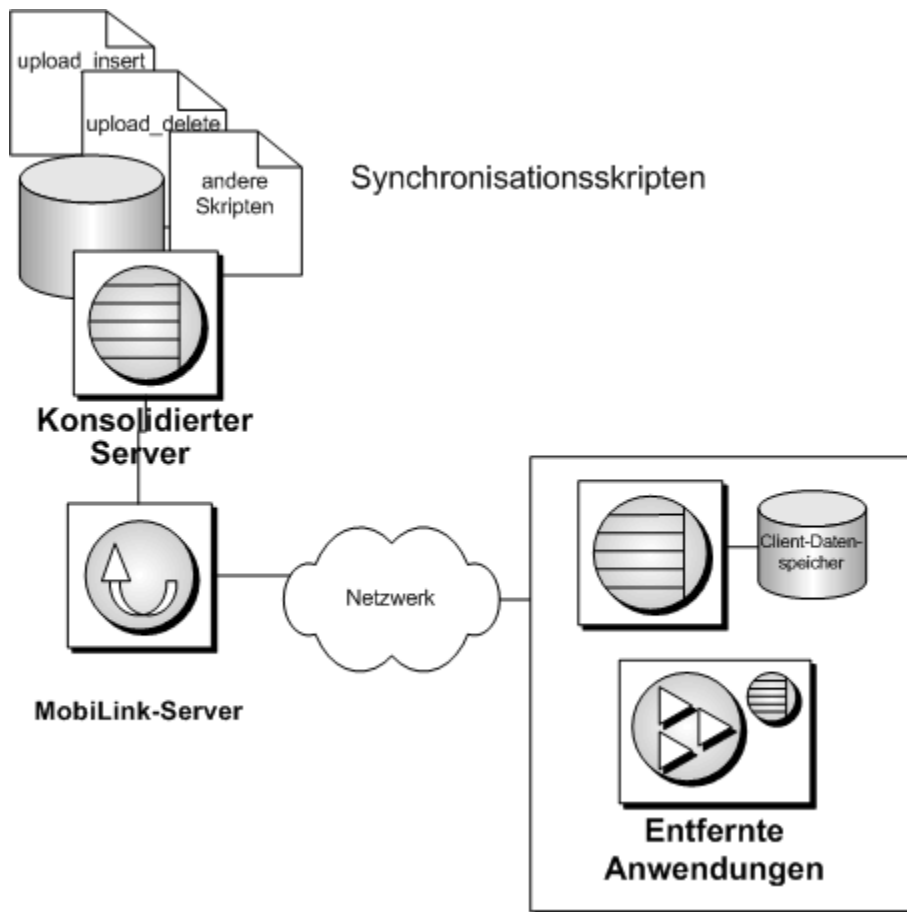
- Wie Daten, die aus der entfernten Datenbank heraufgeladen wurden, in die konsolidierte Datenbank übernommen werden.
- Welche Daten aus der konsolidierten Datenbank heruntergeladen werden sollen.
- wie die Authentifizierung während der Synchronisation ausgeführt wird (optional)

Skripten können individuelle Anweisungen sein oder gespeicherte Prozeduren aufrufen. Sie werden in Ihrer konsolidierten Datenbank gespeichert oder referenziert. Sie fügen Skripten mit Sybase Central oder mit Systemprozeduren in die konsolidierte Datenbank ein.

### **Vorsicht**

Ihre SQL-Synchronisationsskripten bzw. die Prozeduren oder Skripten, die von Ihren SQL-Synchronisationsskripten aufgerufen werden, dürfen keine impliziten oder expliziten COMMIT- oder ROLLBACK-Anweisungen enthalten. COMMIT- und ROLLBACK-Anweisungen innerhalb eines SQL-Skripts ändern den transaktionsbasierten Ablauf der Synchronisationsschritte. Falls Sie diese Anweisungen benutzen, kann MobiLink die Integrität der Daten bei einer Systemstörung nicht garantieren.

Während der Synchronisation liest der MobiLink-Server die Skripten, sofern sie nicht bereits geladen wurden, und führt sie dann in der konsolidierten Datenbank aus.



Der Synchronisationsprozess umfasst mehrere Schritte. Jeder Schritt wird durch ein eindeutiges Ereignis gekennzeichnet. Sie steuern den Synchronisationsprozess durch Synchronisationsskripten, die Sie einigen dieser Ereignisse zuordnen. Sie brauchen aber nur dann ein Skript zu schreiben, wenn bei einem bestimmten Ereignis eine bestimmte Aktion ausgeführt werden soll. Der MobiLink-Server führt die einzelnen Skripten aus, wenn das zugeordnete Ereignis ausgelöst wird. Wenn für ein bestimmtes Ereignis kein Skript definiert wird, geht der MobiLink-Server einfach zum nächsten Schritt über.

Ein Ereignis hat zum Beispiel den Namen `begin_upload_rows`. Sie schreiben ein Skript und ordnen es diesem Ereignis zu. Der MobiLink-Server liest dieses Skript, wenn er es zum ersten Mal benötigt, und führt es in der Uploadphase der Synchronisation aus. Wenn Sie kein Skript schreiben, geht der MobiLink-Server unmittelbar zum nächsten Schritt über und beginnt mit dem Verarbeiten der gesendeten Zeilen.

Einige als Tabellenskripten bezeichnete Skripten sind nicht nur einem Ereignis, sondern auch einer Tabelle in der entfernten Datenbank zugeordnet. Der MobiLink-Server führt einige Aufgaben tabellenweise aus. Eine dieser Aufgaben ist beispielsweise der Download von Zeilen. Sie können viele Skripten mit ein und demselben Ereignis verknüpfen, allerdings müssen sich diese Skripten dann auf unterschiedliche Anwendungstabellen beziehen. Eine andere Möglichkeit besteht darin, für einige Anwendungstabellen viele Skripten zu schreiben, für andere dagegen nur sehr wenige.

Einen Überblick über Ereignisse finden Sie unter „[Der Synchronisationsprozess](#)“ [[MobiLink - Erste Orientierung](#)].

Weitere Hinweise zu allen Skripten, die Sie erstellen können, finden Sie unter „[Synchronisationsergebnisse](#)“ auf [Seite 272](#).

Sie können Skripten in SQL, Java oder .NET schreiben. Dieser Abschnitt ist auf alle Arten von Skripten anzuwenden, konzentriert sich aber vor allem auf die Erstellung von Synchronisationsskripten in SQL.

Eine Beschreibung und einen Vergleich von SQL, Java und .NET finden Sie unter „[Optionen zum Erstellen serverseitiger Synchronisationslogik](#)“ [[MobiLink - Erste Orientierung](#)].

Hinweise zur Erstellung von Skripten in .NET finden Sie unter „[Synchronisationsskripten in .NET](#)“ auf [Seite 536](#).

Hinweise zur Erstellung von Skripten in Java finden Sie unter „[Schreiben eines Synchronisationsskripts in Java](#)“ auf [Seite 463](#).

Weitere Hinweise zur Implementierung von Synchronisationsskripten finden Sie unter „[Synchronisationsmethoden](#)“ auf [Seite 107](#).

## Einfaches Synchronisationsskript

MobiLink stellt viele Ereignisse bereit, die Sie nutzen können, doch Sie müssen nicht unbedingt für jedes Ereignis auch Skripten erstellen. In einem einfachen Synchronisationsmodell brauchen Sie nur wenige Skripten.

Durch den Download aller Zeilen aus der Tabelle in jede einzelne entfernte Datenbank wird die Tabelle ULProduct in der Beispieldatenbank CustDB synchronisiert. In diesem Fall sind auf den entfernten Datenbanken keine Einfügungen zulässig. Sie können diese einfache Form der Synchronisation mit zwei Skripten implementieren. In diesem Fall ist nur zwei Ereignissen ein Skript zugeordnet.

Das MobiLink-Ereignis download\_cursor steuert, welche Zeilen bei einer Synchronisation per Download übertragen werden sollen. Cursorskripten müssen eine SELECT-Anweisung enthalten. Mithilfe dieser Abfragen definiert der MobiLink-Server einen Cursor. Bei einem download\_cursor-Skript wählt der Cursor die Zeilen aus, die in eine bestimmte Tabelle der entfernten Datenbank geladen werden sollen.

In der Beispielanwendung CustDB gibt es ein einziges download\_cursor-Skript für die Tabelle ULProduct dieser Anwendung, das die folgende Abfrage enthält:

```
SELECT prod_id, price, prod_name
FROM ULProduct
```

Diese Abfrage erstellt eine Ergebnismenge. Die Zeilen dieser Ergebnismenge werden an den Client übertragen. In diesem Beispiel werden sämtliche Zeilen der Tabelle übertragen.

Der MobiLink-Server weiß, dass diese Zeilen in die Anwendungstabelle ULProduct geschrieben werden müssen, da dieses Skript bedingt durch die Art und Weise der Speicherung in der konsolidierten Datenbank sowohl mit dem download\_cursor-Ereignis als auch mit der Tabelle ULProduct verknüpft ist. Sybase Central stellt für solche Verknüpfungen bestimmte Funktionen zur Verfügung.

Das zweite erforderliche Ereignis ist `download_delete_cursor`, für das ein Skript, zusammen mit `download_cursor`, für jede herunterzuladende Tabelle definiert sein muss. Dieses einfache Beispiel verwendet nicht die Funktion "Löschungen herunterladen", sodass das Skript als `--{ml_ignore}` definiert ist.

Bei diesem Beispiel wählt die Abfrage Daten aus einer konsolidierten Tabelle aus, die ebenfalls den Namen `ULProduct` hat. Die Tabellennamen müssen aber nicht unbedingt übereinstimmen. Stattdessen könnten Sie Daten einer beliebigen Tabelle bzw. einer beliebigen Tabellenkombination der konsolidierten Datenbank in die Anwendungstabelle `ULProduct` laden, indem Sie die Abfrage entsprechend ändern.

Sie können natürlich auch komplexere Synchronisationsskripten schreiben. So können Sie zum Beispiel ein Skript erstellen, das nur gerade geänderte Zeilen einliest oder jeder entfernten Datenbank jeweils bestimmte Daten liefert.

## Skripten und der Synchronisationsprozess

Jedes Skript entspricht einem bestimmten Ereignis im Synchronisationsprozess. Sie brauchen aber nur dann ein Skript zu schreiben, wenn eine bestimmte Aktion ausgeführt werden soll. Nicht benötigte Ereignisse können undefiniert bleiben.

Die beiden entscheidenden Teile des Prozesses sind die Verarbeitung der hochgeladenen Daten und die Vorbereitung der herunterzuladenden Zeilen. Wenn Zeilen aus einer entfernten Tabelle hochgeladen werden, müssen Sie die entsprechenden Upload-Skripten definieren. Wenn für eine Tabelle Zeilen über SQL heruntergeladen werden sollen, müssen die Skripten `download_cursor` und `download_delete_cursor` beide definiert werden.

Der MobiLink-Server liest und bereitet jedes Skript einmal vor, wenn es zum ersten Mal gebraucht wird. Danach wird das Skript ausgeführt, sobald das Ereignis aufgerufen wird.

### Abfolge der Ereignisse

Weitere Hinweise zur vollständigen Sequenz der MobiLink-Ereignisse finden Sie unter [„Überblick über MobiLink-Ereignisse“ auf Seite 272](#).

Details zur Verarbeitung von Uploads finden Sie unter [„Skripten für den Upload von Zeilen“ auf Seite 262](#).

Details zur Verarbeitung von Downloads finden Sie unter [„Skripten für den Download von Zeilen“ auf Seite 265](#).

### Hinweise

- Die MobiLink-Technologie gestattet das gleichzeitige Synchronisieren mehrerer Clients. Dabei wird für jeden einzelnen Client eine eigene Verbindung mit der konsolidierten Datenbank hergestellt.
- Die Ereignisse `begin_connection` und `end_connection` sind von den einzelnen Synchronisationen unabhängig, da mit einer Verbindung viele Synchronisationsanforderungen verwaltet werden können. Diese Skripten haben keine Parameter. Sie sind Beispiele für Verbindungsskripten.

- Einige Ereignisse werden nur einmal für jede Synchronisation aufgerufen, unabhängig von der Anzahl der zu synchronisierenden Tabellen. Hierbei handelt es sich um Verbindungsskripten.

- Einige Ereignisse werden für jede zu synchronisierende Tabelle einmal aufgerufen. Die diesen Ereignissen zugeordneten Skripten werden als Tabellenskripten bezeichnet.

Obwohl jede Tabelle über eigene Tabellenskripten verfügen kann, können Sie auch Tabellenskripten schreiben, die von mehreren Tabellen gemeinsam genutzt werden. Dies ist allerdings nicht die gängige Vorgehensweise.

- Einige Ereignisse, wie etwa `begin_synchronization`, treten sowohl auf der Verbindungsebene als auch auf der Tabellenebene ein. Sie können für diese Ereignisse sowohl Verbindungs- als auch Tabellenskripten bereitstellen.

Weitere Hinweise, einschließlich detaillierter Informationen zu den einzelnen Skripten und ihren Parametern, finden Sie unter [„Synchronisationseignisse“ auf Seite 272](#).

## Skripttypen

Es gibt zwei Arten von Synchronisationsskripten:

- **Verbindungsskripten** Diese Skripten führen Aktionen aus, die verbindungs- bzw. synchronisationsspezifisch und dabei von den entfernten Tabellen unabhängig sind. Diese Skripten können zusammen mit anderen Skripten verwendet werden, um Ihre Geschäftslogik für die Synchronisation zu implementieren.
- **Tabellenskripten** Diese Skripten führen Aktionen für eine bestimmte Verbindung und eine bestimmte entfernte Tabelle aus. Diese Skripten werden zusammen mit anderen Skripten verwendet, um Ihre Geschäftslogik für die Synchronisation, einschließlich der Konfliktlösung, zu implementieren.

### Siehe auch

- [„Verbindungsskripten“ auf Seite 235](#)
- [„Tabellenskripten“ auf Seite 236](#)

## Verbindungsskripten

Verbindungsskripten steuern Ereignisse höherer Ebenen, die keiner bestimmten Tabelle zugeordnet sind. Sie verwenden diese Ereignisse, um globale Aufgaben auszuführen, die bei jeder Synchronisation erforderlich sind.

Verbindungsskripten steuern Aktionen für die Herstellung und Trennung von Verbindungen sowie Ereignisaktionen für die Synchronisation, z.B. den Beginn und das Ende des Upload- oder Download-Vorgangs. Zu einigen Verbindungsskripten gibt es zugehörige Tabellenskripten. Diese Verbindungsskripten werden immer aufgerufen, unabhängig von den zu synchronisierenden Tabellen.

Sie brauchen aber nur dann ein Verbindungsskript zu schreiben, wenn bei einem bestimmten Ereignis eine bestimmte Aktion ausgeführt werden soll. Sie müssen wahrscheinlich nur für wenige Ereignisse

Skripten erstellen. Die Standardaktion des MobiLink-Servers ist für alle Ereignisse, keine Aktion auszuführen. Für einfache Synchronisationsschemata werden keine Verbindungsskripten benötigt.

### Skriptversion ml\_global

Damit Sie nicht ein- und dasselbe Skript mehrmals definieren müssen, können Sie einmalig Verbindungsskripten festlegen und diese dann aus jeder beliebigen Skriptversion wiederverwenden. Hierzu definieren Sie eine Skriptversion namens ml\_global.

### Siehe auch

- [Skriptversion ml\\_global auf Seite 256](#)

## Tabellenskripten

Tabellenskripten gestatten Aktionen bei bestimmten Ereignissen, die sich auf die Synchronisation einer bestimmten Tabelle beziehen, wie etwa das Hochladen von Zeilen, das Lösen von Konflikten oder das Auswählen von herunterzuladenden Zeilen.

Die Synchronisationsskripten für eine bestimmte Tabelle können sich auf eine beliebige Tabelle bzw. eine beliebige Tabellenkombination in der konsolidierten Datenbank beziehen. Mit dieser Funktion können Sie eine bestimmte entfernte Tabelle mit den Daten aus einer bzw. mehreren Tabelle(n) der konsolidierten Datenbank anfüllen oder die aus einer einzelnen entfernten Tabelle eingelesenen Daten in der konsolidierten Datenbank speichern.

### Tabellennamen müssen nicht übereinstimmen

Die Namen von Tabellen in der entfernten Datenbank müssen nicht mit den Namen der Tabellen in der konsolidierten Datenbank übereinstimmen. Der MobiLink-Server überprüft den Namen der entfernten Tabelle in der Systemtabelle ml\_table und ermittelt, welche Skripten einer bestimmten Tabelle zugeordnet sind. Die Skripten selbst referenzieren die konsolidierten Tabellen Ihrer Wahl.

## Skriptparameter

Die meisten Synchronisationsskripten können Parameter vom MobiLink-Server erhalten. Ausführliche Informationen zu den Parametern, die Sie in den Skripten verwenden können, finden Sie unter [„Synchronisationsereignisse“ auf Seite 272](#).

Es gibt zwei Möglichkeiten, Parameter in SQL-Skripten anzugeben:

- Benannte Skriptparameter
- Fragezeichen (in SQL-Skripten veraltet)

### Benannte Skriptparameter

Benannte Parameter bieten folgende Vorteile gegenüber (veralteten) Fragezeichen:

- Mit benannten Parametern können Sie beliebige Teilmengen der verfügbaren Parameter in beliebiger Reihenfolge angeben.

- Mit Ausnahme der I/O-Parameter können Sie einen benannten Parameter mehrmals in einem Skript angeben.
- Wenn Sie benannte Parameter verwenden, können Sie die entfernte ID in Ihren Skripten angeben. Dies ist die einzige Möglichkeit, entfernte IDs in Skripten festzulegen.
- Sie können Ihre eigenen benannten Parameter erstellen. Siehe „[Benutzerdefinierte benannte Parameter](#)“ auf Seite 252.

Benannte Parameter und Fragezeichen dürfen nicht gemeinsam in einem Skript verwendet werden.

Es gibt vier Arten von benannten Parametern in MobiLink. Wenn Sie einen benannten Parameter angeben möchten, müssen Sie ihm seinen Typ wie folgt als Präfix voranstellen:

| Typen benannter Parameter                                                                                                                                                                                              | Präfix | Beispiele                                    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|----------------------------------------------|
| Systemparameter                                                                                                                                                                                                        | s.     | {ml s.remote_id}                             |
| Zeilenparameter. (Der Spaltenname. Falls der Spaltenname Leerzeichen enthält, setzen Sie ihn in Anführungszeichen oder eckige Klammern.)                                                                               | r.     | {ml r.cust_id}<br><br>{ml r."Spaltenname"}   |
| Alte Zeilenparameter. (Wird nur in uplo-ad_update-Skripten verwendet, um die Pre-Image-Spaltenwerte festzulegen. Falls der Spaltenname Leerzeichen enthält, setzen Sie ihn in Anführungszeichen oder eckige Klammern.) | o.     | {ml o.cust_name}<br><br>{ml o."Spaltenname"} |
| Authentifizierungsparameter. Siehe „ <a href="#">Authentifizierungsparameter</a> “ auf Seite 253.                                                                                                                      | a.     | {ml a.1}                                     |

| Typen benannter Parameter                                                                  | Präfix      | Beispiele                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------------------------------------------------------------------------------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Benutzerdefinierte Parameter. Siehe „Benutzerdefinierte benannte Parameter“ auf Seite 252. | u. oder ui. | <ul style="list-style-type: none"><li>• <b>{ml u.varname}</b> Zu verwenden, wenn der Parameter aktualisiert werden soll. Benutzerdefinierte Parameter mit dem Präfix u. sind für Ein- und Ausgabe.</li><li>• <b>{ml ui.varname}</b> Zu verwenden, wenn der Parameter nur referenziert werden soll. Benutzerdefinierte Parameter mit dem Präfix ui. sind nur für die Eingabe.</li></ul> |

Sie referenzieren einen Skriptparameter nach Namen, indem Sie den Parameter in geschweifte Klammern setzen und ihm das Präfix voranstellen wie in **{ml Parameter}**. Zum Beispiel, **{ml s.action\_code}**. Geschweifte Klammern entsprechen der ODBC-Konvention.

Der Einfachheit halber können Sie einen größeren Codeteil in geschweifte Klammern setzen, solange er keine Schemanamen enthält, die mit dem Namen eines MobiLink-Skriptparameters identisch sind. Beispielsweise sind die folgenden upload\_insert-Skripten gültig und äquivalent:

```
INSERT INTO t ( id, c0 ) VALUES( {ml r.id}, {ml r.c0} )

und

INSERT INTO t ( is, c0 ) VALUES({ml r.id, r.c0})

und

{ml INSERT INTO t ( id, c0 ) VALUES( r.id, r.c0 ) }
```

Durch Fragezeichen dargestellte Skriptparameter (veraltet für SQL)

**Hinweis**

Die Darstellung von Parametern durch Fragezeichen ist in SQL-Skripten veraltet. Es wird dringend empfohlen, stattdessen benannte Parameter zu verwenden. Siehe [Benannte Skriptparameter auf Seite 236](#) und „Benutzerdefinierte benannte Parameter“ auf Seite 252.

In ODBC werden Parameter durch Fragezeichen dargestellt. Wenn Sie Fragezeichen in Ihren SQL-Skripten in MobiLink verwenden möchten, schreiben Sie für jeden Parameter ein einzelnes Fragezeichen in Ihr Skript. Der MobiLink-Server ersetzt jedes Fragezeichen durch den Wert eines Parameters. Er ersetzt die Werte in der Reihenfolge, in der die Parameter in der Skriptdefinition erscheinen.

Einige Parameter sind optional. Ein Parameter ist nur dann optional, wenn keine weiteren Parameter angegeben werden. Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten. Die Reihenfolge der Parameter muss so sein, wie sie für jedes Ereignis angegeben wurde. Siehe „Synchronisationsereignisse“ auf Seite 272.

## Skriptparameter kommentieren

Die folgenden Kommentarformen werden erkannt:

- Präfix mit doppeltem Bindestrich (--)
- Präfix mit doppeltem Schrägstrich (//)
- Blockkommentar (/\* \*/)

Bei den ersten beiden Formen wird der Text bis zum Zeilenende ignoriert. Bei der dritten Form wird der gesamte Text zwischen dem Präfix /\* und dem Suffix \*/ ignoriert. Ein Blockkommentar kann nicht verschachtelt werden.

Andere herstellerspezifische Kommentartypen werden nicht erkannt und sollten nicht verwendet werden, um Referenzen zu einem benannten Parameter zu kommentieren.

## MobiLink-Systemparameter und -Ereignisse

| Systemparameter        | Ereignistyp            | Verfügbarkeit des Parameters in Ereignissen                                                                                                                                                                                   |
|------------------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| action_code            | Verbindung             | Der Parameter action_code ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>• handle_error</li> <li>• handle_odbc_error</li> <li>• report_error</li> <li>• report_odbc_error</li> </ul>      |
| authentication_message | Verbindung             | Der authentication_message-Parameter ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>• authenticate_parameters</li> <li>• authenticate_user</li> <li>• authenticate_user_hashed</li> </ul> |
| authentication_status  | Verbindung             | Der Parameter authentication_status ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>• authenticate_parameters</li> <li>• authenticate_user</li> <li>• authenticate_user_hashed</li> </ul>  |
| bytes                  | Verbindung und Tabelle | Der Parameter bytes ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>• download_statistics</li> <li>• upload_statistics</li> </ul>                                                          |

| Systemparameter    | Ereignistyp            | Verfügbarkeit des Parameters in Ereignissen                                                                                                                                                                                               |
|--------------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| conflicted_deletes | Verbindung und Tabelle | Der Parameter conflicted_deletes ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>• upload_statistics</li> </ul>                                                                                        |
| conflicted_inserts | Verbindung und Tabelle | Der Parameter conflicted_inserts ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>• upload_statistics</li> </ul>                                                                                        |
| conflicted_updates | Verbindung und Tabelle | Der Parameter conflicted_updates ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>• upload_statistics</li> </ul>                                                                                        |
| connection_retries | Verbindung             | Der Parameter conflicted_retries ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>• synchronization_statistics</li> </ul>                                                                               |
| deadlocks          | Verbindung und Tabelle | Der Parameter deadlocks ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>• synchronization_statistics (Verbindungsereignis)</li> <li>• upload_statistics (Verbindungs- und Tabellenereignis)</li> </ul> |
| deleted_rows       | Verbindung und Tabelle | Der Parameter deleted_rows ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>• download_statistics</li> <li>• upload_statistics</li> </ul>                                                               |
| error_code         | Verbindung             | Der Parameter error_code ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>• handle_error</li> <li>• modify_error_message</li> <li>• report_error</li> </ul>                                             |

| Systemparameter          | Ereignistyp            | Verfügbarkeit des Parameters in Ereignissen                                                                                                                                                                                                                       |
|--------------------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| error_message            | Verbindung             | <p>Der Parameter error_message ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• handle_error</li> <li>• handle_odbc_error</li> <li>• modify_error_message</li> <li>• report_error</li> <li>• report_odbc_error</li> </ul> |
| errors                   | Verbindung und Tabelle | <p>Der Parameter errors ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• download_statistics</li> <li>• synchronization_statistics</li> <li>• upload_statistics</li> </ul>                                                |
| event_name               | Verbindung und Tabelle | <p>Der Parameter event_name ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• time_statistics</li> </ul>                                                                                                                   |
| fetches_rows             | Verbindung und Tabelle | <p>Der Parameter fetches_rows ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• download_statistics</li> </ul>                                                                                                             |
| file_authentication_code | Verbindung             | <p>Der Parameter file_authentication_code ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• authenticate_file_transfer</li> <li>• authenticate_file_upload</li> </ul>                                                      |
| filename                 | Verbindung             | <p>Der Parameter filename ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• authenticate_file_transfer</li> <li>• authenticate_file_upload</li> </ul>                                                                      |
| file_size                | Verbindung             | <p>Der Parameter file_size ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• authenticate_file_upload</li> </ul>                                                                                                           |
| filtered_rows            | Verbindung und Tabelle | <p>Der Parameter filtered_rows ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• download_statistics</li> </ul>                                                                                                            |

| Systemparameter     | Ereignistyp            | Verfügbarkeit des Parameters in Ereignissen                                                                                                                                                                                                                                                                         |
|---------------------|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| generation_number   | Verbindung             | Der Parameter generation_number ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>begin_publication</li> <li>end_publication</li> </ul>                                                                                                                                            |
| hashed_new_password | Verbindung             | Der Parameter hashed_new_password ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>authenticate_user_hashed</li> </ul>                                                                                                                                                            |
| hashed_password     | Verbindung             | Der Parameter hashed_password ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>authenticate_user_hashed</li> </ul>                                                                                                                                                                |
| ignored_deletes     | Verbindung und Tabelle | Der Parameter ignored_deletes ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>upload_statistics</li> </ul>                                                                                                                                                                       |
| ignored_inserts     | Verbindung und Tabelle | Der Parameter ignored_inserts ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>upload_statistics</li> </ul>                                                                                                                                                                       |
| ignored_updates     | Verbindung und Tabelle | Der Parameter ignored_updates ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>upload_statistics</li> </ul>                                                                                                                                                                       |
| inserted_rows       | Verbindung und Tabelle | Der Parameter inserted_rows ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>upload_statistics</li> </ul>                                                                                                                                                                         |
| last_download       | Verbindung             | Der Parameter last_download ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>begin_download</li> <li>end_download</li> <li>modify_last_download_timestamp</li> <li>modify_next_last_download_timestamp</li> <li>nonblocking_download_ack</li> <li>prepare_for_download</li> </ul> |

| Systemparameter           | Ereignistyp            | Verfügbarkeit des Parameters in Ereignissen                                                                                                                                                                                                                                                                                                                                |
|---------------------------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| last_publication_download | Verbindung             | <p>Der Parameter last_publication_download ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• begin_publication</li> <li>• end_publication</li> <li>• publication_nonblocking_download_ack</li> </ul>                                                                                                                                |
| last_publication_upload   | Verbindung             | <p>Der Parameter last_publication_upload ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• begin_publication</li> <li>• end_publication</li> </ul>                                                                                                                                                                                  |
| last_table_download       | Tabelle                | <p>Der Parameter last_table_download ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• begin_download</li> <li>• begin_download_deletes</li> <li>• begin_download_rows</li> <li>• download_cursor</li> <li>• download_delete_cursor</li> <li>• end_download</li> <li>• end_download_deletes</li> <li>• end_download_rows</li> </ul> |
| maximum_time              | Verbindung und Tabelle | <p>Der Parameter maximum_time ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• time_statistics</li> </ul>                                                                                                                                                                                                                          |
| minimum_time              | Verbindung und Tabelle | <p>Der Parameter minimum_time ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• time_statistics</li> </ul>                                                                                                                                                                                                                          |
| new_password              | Verbindung             | <p>Der Parameter new_password ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• authenticate_user</li> </ul>                                                                                                                                                                                                                        |
| next_last_download        | Verbindung             | <p>Der Parameter next_last_download ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• generate_next_last_download_timestamp</li> <li>• modify_next_last_download_timestamp</li> </ul>                                                                                                                                               |

| Systemparameter  | Ereignistyp            | Verfügbarkeit des Parameters in Ereignissen                                                                                                                                                                           |
|------------------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| number_of_calls  | Verbindung und Tabelle | Der Parameter number_of_calls ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>time_statistics</li> </ul>                                                                           |
| odbc_state       | Verbindung             | Der Parameter odbc_state ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>handle_odbc_error</li> <li>report_odbc_error</li> </ul>                                                   |
| password         | Verbindung             | Der Parameter password ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>authenticate_user</li> </ul>                                                                                |
| publication_name | Verbindung             | Der Parameter publication_name ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>begin_publication</li> <li>end_publication</li> <li>publication_nonblocking_download_ack</li> </ul> |

| Systemparameter | Ereignistyp            | Verfügbarkeit des Parameters in Ereignissen                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| remote_id       | Verbindung und Tabelle | <p>Der Parameter remote_id ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• authenticate_parameters (Verbindungsereignis)</li> <li>• authenticate_user (Verbindungsereignis)</li> <li>• authenticate_user_hashed (Verbindungsereignis)</li> <li>• begin_download (Verbindungs- und Tabellenereignis)</li> <li>• begin_download_deletes (Tabellenereignis)</li> <li>• begin_download_rows (Tabellenereignis)</li> <li>• begin_publication (Verbindungsereignis)</li> <li>• begin_synchronization (Verbindungs- und Tabellenereignis)</li> <li>• begin_upload (Verbindungs- und Tabellenereignis)</li> <li>• begin_upload_deletes (Tabellenereignis)</li> <li>• begin_upload_rows (Tabellenereignis)</li> <li>• download_cursor (Tabellenereignis)</li> <li>• download_delete_cursor (Tabellenereignis)</li> <li>• download_statistics (Verbindungs- und Tabellenereignis)</li> <li>• end_download (Verbindungs- und Tabellenereignis)</li> <li>• end_download_deletes (Tabellenereignis)</li> <li>• end_download_rows (Tabellenereignis)</li> <li>• end_publication (Verbindungsereignis)</li> <li>• end_synchronization (Verbindungs- und Tabellenereignis)</li> <li>• end_upload (Verbindungs- und Tabellenereignis)</li> <li>• end_upload_deletes (Tabellenereignis)</li> <li>• end_upload_rows (Tabellenereignis)</li> <li>• generate_next_last_download_timestamp (Verbindung)</li> <li>• handle_error (Verbindungsereignis)</li> <li>• handle_odbc_error (Verbindungsereignis)</li> <li>• modify_error_message (Verbindungsereignis)</li> <li>• modify_last_download_timestamp (Verbindungsereignis)</li> <li>• modify_next_last_download_timestamp (Verbindungsereignis)</li> <li>• modify_user (Verbindungsereignis)</li> </ul> |

| Systemparameter | Ereignistyp | Verfügbarkeit des Parameters in Ereignissen                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 |             | <ul style="list-style-type: none"> <li>• nonblocking_download_ack (Verbindungsereignis)</li> <li>• prepare_for_download (Verbindungsereignis)</li> <li>• publication_nonblocking_download_ack (Verbindungsereignis)</li> <li>• report_error (Verbindungsereignis)</li> <li>• report_odbc_error (Verbindungsereignis)</li> <li>• resolve_conflict (Tabellenereignis)</li> <li>• synchronization_statistics (Verbindungs- und Tabellenereignis)</li> <li>• time_statistics (Verbindungs- und Tabellenereignis)</li> <li>• upload_delete (Tabellenereignis)</li> <li>• upload_fetch (Tabellenereignis)</li> <li>• upload_fetch_column_conflict (Tabellenereignis)</li> <li>• upload_insert (Tabellenereignis)</li> <li>• upload_new_row_insert (Tabellenereignis)</li> <li>• upload_old_row_insert (Tabellenereignis)</li> <li>• upload_statistics (Verbindungs- und Tabellenereignis)</li> <li>• upload_update (Tabellenereignis)</li> </ul> |
| remote_key      | Verbindung  | <p>Der Parameter remote_key ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• authenticate_file_transfer</li> <li>• authenticate_file_upload</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

| Systemparameter | Ereignistyp            | Verfügbarkeit des Parameters in Ereignissen                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| script_version  | Verbindung und Tabelle | <p>Der script_version-Parameter ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• authenticate_file_transfer (Verbindungsereignis)</li> <li>• authenticate_file_upload (Verbindungsereignis)</li> <li>• authenticate_parameters (Verbindungsereignis)</li> <li>• authenticate_user (Verbindungsereignis)</li> <li>• authenticate_user_hashed (Verbindungsereignis)</li> <li>• begin_download (Verbindungs- und Tabellenereignis)</li> <li>• begin_download_deletes (Tabellenereignis)</li> <li>• begin_download_rows (Tabellenereignis)</li> <li>• begin_publication (Verbindungsereignis)</li> <li>• begin_synchronization (Verbindungs- und Tabellenereignis)</li> <li>• begin_upload (Verbindungs- und Tabellenereignis)</li> <li>• begin_upload_deletes (Tabellenereignis)</li> <li>• begin_upload_rows (Tabellenereignis)</li> <li>• download_cursor (Tabellenereignis)</li> <li>• download_delete_cursor (Tabellenereignis)</li> <li>• download_statistics (Verbindungs- und Tabellenereignis)</li> <li>• end_download (Verbindungs- und Tabellenereignis)</li> <li>• end_download_deletes (Tabellenereignis)</li> <li>• end_download_rows (Tabellenereignis)</li> <li>• end_publication (Verbindungsereignis)</li> <li>• end_synchronization (Verbindungs- und Tabellenereignis)</li> <li>• end_upload (Verbindungs- und Tabellenereignis)</li> <li>• end_upload_deletes (Tabellenereignis)</li> <li>• end_upload_rows (Tabellenereignis)</li> <li>• generate_next_last_download_timestamp (Verbindung)</li> <li>• handle_DownloadData (Verbindungsereignis)</li> <li>• handle_error (Verbindungsereignis)</li> <li>• handle_odbc_error (Verbindungsereignis)</li> <li>• modify_error_message (Verbindungsereignis)</li> </ul> |

| Systemparameter | Ereignistyp | Verfügbarkeit des Parameters in Ereignissen                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 |             | <ul style="list-style-type: none"> <li>• modify_last_download_timestamp (Verbindungsereignis)</li> <li>• modify_next_last_download_timestamp (Verbindungsereignis)</li> <li>• modify_user (Verbindungsereignis)</li> <li>• nonblocking_download_ack (Verbindungsereignis)</li> <li>• prepare_for_download (Verbindungsereignis)</li> <li>• publication_nonblocking_download_ack (Verbindungsereignis)</li> <li>• report_error (Verbindungsereignis)</li> <li>• report_odbc_error (Verbindungsereignis)</li> <li>• resolve_conflict (Tabellenereignis)</li> <li>• synchronization_statistics (Verbindungs- und Tabellenereignis)</li> <li>• time_statistics (Verbindungs- und Tabellenereignis)</li> <li>• upload_delete (Tabellenereignis)</li> <li>• upload_fetch (Tabellenereignis)</li> <li>• upload_fetch_column_conflict (Tabellenereignis)</li> <li>• upload_insert (Tabellenereignis)</li> <li>• upload_new_row_insert (Tabellenereignis)</li> <li>• upload_old_row_insert (Tabellenereignis)</li> <li>• upload_statistics (Verbindungs- und Tabellenereignis)</li> <li>• upload_update (Tabellenereignis)</li> </ul> <div> <b>Hinweis</b><br/> Auf die Skriptversion kann aus Java- und .NET-Skripten heraus zugegriffen werden, indem die getVersion-Methode auf die DBConnectionContext-Klassen angewendet wird. Siehe <a href="#">DBConnectionContext.getVersion-Methode [MobiLink-Server Java]</a> auf Seite 482 und <a href="#">DBConnectionContext.GetVersion-Methode [MobiLink-Server .NET]</a> auf Seite 570. </div> |
| subdir          | Verbindung  | Der Parameter subdir ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>• authenticate_file_transfer</li> <li>• authenticate_file_upload</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

| Systemparameter     | Ereignistyp            | Verfügbarkeit des Parameters in Ereignissen                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| subscription_id     | Verbindung             | <p>Der Parameter subscription_id ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• begin_publication</li> <li>• end_publication</li> <li>• publication_nonblocking_download_ack</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| synchronization_ok  | Verbindung und Tabelle | <p>Der Parameter synchronization_ok ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• end_synchronization</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| synchronized_tables | Verbindung             | <p>Der Parameter synchronized_tables ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• synchronization_statistics</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| table               | Verbindung und Tabelle | <p>Der Parameter table ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• begin_download (Tabellenereignis)</li> <li>• begin_download_deletes (Tabellenereignis)</li> <li>• begin_download_rows (Tabellenereignis)</li> <li>• begin_synchronization (Tabellenereignis)</li> <li>• begin_upload (Tabellenereignis)</li> <li>• begin_upload_deletes (Tabellenereignis)</li> <li>• begin_upload_rows (Tabellenereignis)</li> <li>• download_statistics (Tabellenereignis)</li> <li>• end_download (Tabellenereignis)</li> <li>• end_download_deletes (Tabellenereignis)</li> <li>• end_download_rows (Tabellenereignis)</li> <li>• end_synchronization (Tabellenereignis)</li> <li>• end_upload (Tabellenereignis)</li> <li>• end_upload_deletes (Tabellenereignis)</li> <li>• end_upload_rows (Tabellenereignis)</li> <li>• handle_error (Verbindungsereignis)</li> <li>• handle_odbc_error (Verbindungsereignis)</li> <li>• report_error (Verbindungsereignis)</li> <li>• report_odbc_error (Verbindungsereignis)</li> <li>• resolve_conflict (Tabellenereignis)</li> <li>• synchronization_statistics (Tabellenereignis)</li> <li>• time_statistics (Tabellenereignis)</li> <li>• upload_statistics (Tabellenereignis)</li> </ul> |

| Systemparameter | Ereignistyp            | Verfügbarkeit des Parameters in Ereignissen                                                                                                  |
|-----------------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| total_time      | Verbindung und Tabelle | Der Parameter total_time ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>• time_statistics</li> </ul>     |
| updated_rows    | Verbindung und Tabelle | Der Parameter updated_rows ist in den folgenden Ereignissen verfügbar: <ul style="list-style-type: none"> <li>• upload_statistics</li> </ul> |

| Systemparameter | Ereignistyp            | Verfügbarkeit des Parameters in Ereignissen                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| username        | Verbindung und Tabelle | <p>Der Parameter username ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• authenticate_file_transfer (Verbindungsereignis)</li> <li>• authenticate_file_upload (Verbindungsereignis)</li> <li>• authenticate_parameters (Verbindungsereignis)</li> <li>• authenticate_user (Verbindungsereignis)</li> <li>• authenticate_user_hashed (Verbindungsereignis)</li> <li>• begin_download (Verbindungs- und Tabellenereignis)</li> <li>• begin_download_deletes (Tabellenereignis)</li> <li>• begin_download_rows (Tabellenereignis)</li> <li>• begin_publication (Verbindungsereignis)</li> <li>• begin_synchronization (Verbindungs- und Tabellenereignis)</li> <li>• begin_upload (Verbindungs- und Tabellenereignis)</li> <li>• begin_upload_deletes (Tabellenereignis)</li> <li>• begin_upload_rows (Tabellenereignis)</li> <li>• download_cursor (Tabellenereignis)</li> <li>• download_delete_cursor (Tabellenereignis)</li> <li>• download_statistics (Verbindungs- und Tabellenereignis)</li> <li>• end_download (Verbindungs- und Tabellenereignis)</li> <li>• end_download_deletes (Tabellenereignis)</li> <li>• end_download_rows (Tabellenereignis)</li> <li>• end_publication (Verbindungsereignis)</li> <li>• end_synchronization (Verbindungs- und Tabellenereignis)</li> <li>• end_upload (Verbindungs- und Tabellenereignis)</li> <li>• end_upload_deletes (Tabellenereignis)</li> <li>• end_upload_rows (Tabellenereignis)</li> <li>• generate_next_last_download_timestamp (Verbindungsereignis)</li> <li>• handle_error (Verbindungsereignis)</li> <li>• handle_odbc_error (Verbindungsfehler)</li> <li>• modify_error_message (Verbindungsfehler)</li> <li>• modify_last_download_timestamp (Verbindungsfehler)</li> </ul> |

| Systemparameter | Ereignistyp            | Verfügbarkeit des Parameters in Ereignissen                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 |                        | <ul style="list-style-type: none"> <li>• modify_next_last_download_timestamp (Verbindungsereignis)</li> <li>• modify_user (Verbindungsereignis)</li> <li>• nonblocking_download_ack (Verbindungsereignis)</li> <li>• prepare_for_download (Verbindungsereignis)</li> <li>• publication_nonblocking_download_ack (Verbindungsereignis)</li> <li>• report_error (Verbindungsereignis)</li> <li>• report_odbc_error (Verbindungsereignis)</li> <li>• resolve_conflict (Tabellenereignis)</li> <li>• synchronization_statistics (Verbindungs- und Tabellenereignis)</li> <li>• time_statistics (Verbindungs- und Tabellenereignis)</li> <li>• upload_delete (Tabellenereignis)</li> <li>• upload_fetch (Tabellenereignis)</li> <li>• upload_fetch_column_conflict (Tabellenereignis)</li> <li>• upload_insert (Tabellenereignis)</li> <li>• upload_new_row_insert (Tabellenereignis)</li> <li>• upload_old_row_insert (Tabellenereignis)</li> <li>• upload_statistics (Verbindungs- und Tabellenereignis)</li> <li>• upload_update (Tabellenereignis)</li> </ul> |
| warnings        | Verbindung und Tabelle | <p>Der Parameter warnings ist in den folgenden Ereignissen verfügbar:</p> <ul style="list-style-type: none"> <li>• download_statistics</li> <li>• synchronization_statistics</li> <li>• upload_statistics</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

## Benutzerdefinierte benannte Parameter

Sie können auch Ihre eigenen Parameter definieren. Diese sind besonders nützlich für RDBMS, die keine benutzerdefinierten Variablen erlauben.

Benutzerdefinierte Parameter werden festgelegt (und auf NULL gesetzt), wenn sie das erste Mal referenziert werden. Sie müssen mit ui und einem Punkt (ui.) beginnen, wenn der Parameter nur referenziert werden soll (nur für die Eingabe), und mit u und einem Punkt (u.), wenn der Parameter aktualisiert werden soll (Ein- und Ausgabe). Ein benutzerdefinierter Parameter bleibt für die Dauer einer Synchronisation bestehen. Er wird beim Start jeder Synchronisation auf NULL gesetzt.

Eine typische Anwendung eines benutzerdefinierten Parameters ist der Zugriff auf Statusinformationen, ohne diese in einer Tabelle speichern zu müssen (mit möglicherweise komplexen Joins).

## Beispiel

Angenommen, Sie erstellen eine gespeicherte Prozedur namens MyCustomProc, die eine Variable namens var1 auf custom\_value setzt:

```
CREATE PROCEDURE MyCustomProc(
  IN username VARCHAR(128), INOUT var1 VARCHAR(128)
)
begin
  SET var1 = 'custom_value';
end
```

Das folgende begin\_connection-Skript definiert den benutzerdefinierten Parameter var1 und setzt den Wert auf custom\_value:

```
CALL ml_add_connection_script (
  'version1',
  'begin_synchronization',
  '{call MyCustomProc( {ml s.username}, {ml u.var1} )}' );
```

Das folgende download\_cursor-Tabellenskript referenziert var1, dessen Wert custom\_value ist:

```
CALL ml_add_table_script (
  'version1',
  'MyTable',
  'download_cursor',
  'select pk,col1 from MyTable where u_name = {ml s.username} and
some_other_column = {ml ui.var1}' );
```

Angenommen, Sie haben eine andere gespeicherte Prozedur namens MyPFDPProc, die ihren ersten I/O-Parameter definiert. Das folgende Skript prepare\_for\_download ändert den Wert von var1 auf pfd\_value:

```
CALL ml_add_connection_script (
  'version1',
  'prepare_for_download',
  '{call MyPFDPProc( {ml u.var1} )}' );
```

Das folgende Skript begin\_download referenziert var1, dessen Wert nun pfd\_value ist:

```
CALL ml_add_connection_script (
  'version1',
  'begin_download',
  'insert into SomeTable values( {ml s.username}, {ml ui.var1} )' );
```

## Authentifizierungsparameter

In MobiLink-Skripten können Authentifizierungsparameter mithilfe von benannten Parametern angegeben werden. Wenn benannte Parameter verwendet werden, muss den Authentifizierungsparametern der Buchstabe a als Präfix vorangestellt werden, wie z.B. {ml a.1}. Die Parameter müssen Nummern sein, die mit 1 beginnen und deren Höchstwert 255 ist. Jeder Parameter kann maximal 4000 Byte lang sein. Die Werte werden von MobiLink-Clients gesendet.

Wenn Authentifizierungsparameter in den authenticate\_\*-Skripten verwendet werden, übergeben sie Authentifizierungsinformationen.

Authentifizierungsparameter können in allen anderen Ereignissen (außer begin\_connection und end\_connection) verwendet werden, um Informationen von MobiLink-Clients zu übergeben. Diese

Technik ist eine bequeme Methode, etwas durchzuführen, das andernfalls das Hochladen von Zeilen in eine Tabelle erfordert. Mit Authentifizierungsparametern stehen die Werte vor den Upload-Ereignissen der Tabelle zur Verfügung.

Bei entferntem SQL Anywhere übergeben Sie die Informationen mit der Option `dbmlsync -ap`. Bei entferntem UltraLite übergeben Sie die Informationen mit `auth_parms` und `num_auth_parms`.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- `dbmlsync`: „`dbmlsync-Option -ap`“ [*MobiLink - Clientadministration*]
- UltraLite: „Synchronisationsparameter Authentication Parameters“ [*UltraLite - Datenbankverwaltung*] und „Synchronisationsparameter Number of Authentication Parameters“ [*UltraLite - Datenbankverwaltung*]

### Beispiel

Für entfernte UltraLite-Datenbanken, werden die Parameter mit den Feldern `num_auth_parms` und `auth_parms` in der `ul_sync_info`-Struktur übergeben. `num_auth_parms` ist die Anzahl der Parameter, von 0 bis 255. `auth_parms` ist ein Zeiger auf ein Array von Zeichenfolgen. Während der Synchronisation werden die Authentifizierungsparameter auf die gleiche Weise verschleiert wie Kennwörter. Wenn `num_auth_parms` gleich 0 ist, setzen Sie `auth_parms` auf `NULL`. Nachfolgend finden Sie ein Beispiel für die Übergabe von Parametern in UltraLite:

```
ul_char * Params[ 3 ] = { UL_TEXT( "param1" ),
                          UL_TEXT( "param2" ), UL_TEXT( "param3" ) };

...
info.num_auth_parms = 3;
info.auth_parms = Params;
```

Bei entfernten SQL Anywhere-Datenbanken können Sie Authentifizierungsparameter mit der Option `dbmlsync -ap` in einer kommagetrennten Liste übergeben. Die folgende Befehlszeile übergibt z.B drei Parameter:

```
dbmlsync -ap "param1,param2,param3"
```

Auf dem Server referenzieren Sie die Authentifizierungsparameter in der Reihenfolge, in der sie gesendet wurden. In diesem Beispiel könnte das `authenticate_parameters`-Skript wie folgt aussehen:

```
CALL my_auth_parm (
  {ml s.authentication_status},
  {ml s.remote_id},
  {ml s.username},
  {ml a.1},
  {ml a.2},
  {ml a.3}
)
```

## Skriptversionen

Skripten werden in Gruppen sogenannter **Skriptversionen** unterteilt. Durch Angabe einer bestimmten Skriptversion können die MobiLink-Clients festlegen, mit welchen Synchronisationsskripten der Upload verarbeitet und der Download vorbereitet wird.

Weitere Hinweise zum Hinzufügen einer Skriptversion zu einer konsolidierten Datenbank finden Sie unter „[Skriptversionen zu konsolidierten Datenbanken hinzufügen](#)“ auf Seite 256.

### Skriptversionen verwenden

Mithilfe von Skriptversionen können Sie Ihre Skripten in Gruppen unterteilen, die Sie bei unterschiedlichen Voraussetzungen verwenden können. Diese Funktion gibt Ihnen eine höhere Flexibilität und ist bei folgenden Zielsetzungen besonders hilfreich:

- **Anwendungen anpassen** Mit unterschiedlichen Skriptengruppen können Sie die Daten unterschiedlicher entfernter Benutzertypen verarbeiten. So können Sie beispielsweise unterschiedliche Skriptengruppen schreiben, damit einmal die Manager und ein weiteres Mal andere Benutzer innerhalb einer Organisation eine Synchronisation vornehmen. Natürlich könnten Sie diese Funktionalität auch mit einem Skriptensatz erzielen, diese Skripten wären aber viel komplizierter.
- **Upgrade von Anwendungen** Wenn Sie eine Datenbankanwendung aktualisieren möchten, benötigen Sie unter Umständen neue Skripten, da die neue Version der Anwendung die Daten möglicherweise anders verwaltet. Neue Skripten sind fast immer dann erforderlich, wenn sich die entfernte Datenbank ändert. Es ist im Allgemeinen unmöglich, sämtliche Benutzer gleichzeitig zu aktualisieren. Da es möglich ist, sowohl alte als auch neue Skripten auf dem Server parallel zu verwalten, können alle Benutzer eine Synchronisation unabhängig davon durchführen, welche Version der Anwendung sie gerade verwenden.
- **Mehrere Anwendungen aufrechterhalten** Unter Umständen kann ein einzelner MobiLink-Server zwei völlig unterschiedliche Anwendungen synchronisieren. So verwenden beispielsweise einige Mitarbeiter eines Unternehmens eine Anwendung für den Vertrieb, während sich andere Mitarbeiter mit einer Anwendung für die Inventarverwaltung beschäftigen. Wenn diese beiden Anwendungen unterschiedliche Daten benötigen, können Sie zwei Versionen der Synchronisationsskripten erstellen, nämlich eine Skriptversion für jede Anwendung.
- **Eigenschaften für die Skriptversion festlegen** Sie können für Ihre Skriptversion Eigenschaften festlegen, die von Klassen in der .NET- oder Java-Synchronisationslogik referenziert werden können. Siehe „[ml\\_add\\_property-Systemprozedur](#)“ auf Seite 664.

### Skriptversionsnamen zuordnen

Bei dem Versionsnamen eines Skripts handelt es sich um eine Zeichenfolge. Diesen Namen geben Sie an, wenn Sie das Skript zu einer konsolidierten Datenbank hinzufügen möchten. Wenn Sie beispielsweise neue Skripten mithilfe der gespeicherten Prozeduren `ml_add_connection_script` und `ml_add_table_script` hinzufügen möchten, dann wird der Versionsname der Skripten als erster Parameter verwendet. Eine andere Möglichkeit besteht darin, dass Sie, wenn Sie die Skripten mit Sybase Central hinzufügen, zur Eingabe des Skriptversionsnamens aufgefordert werden.

Sie können für Skriptversionen die Namen **ml\_sis\_1** oder **ml\_qa\_1** nicht verwenden. Diese Namen werden nur intern von MobiLink benutzt.

#### Vorsicht

Grundsätzlich sollten die Namen Ihrer Skriptversionen nicht mit **ml\_** beginnen. Skriptversionen, die mit **ml\_** beginnen, sind der internen Verwendung vorbehalten.

## Skriptversion für eine Synchronisation festlegen

Wenn auf dem entfernten Computer bei der Initiierung der Synchronisation keine Skriptversion angegeben wird, schlägt die Synchronisation fehl. Siehe „[Skriptversionen und Subskriptionen](#)“ [[MobiLink - Clientadministration](#)].

## Skriptversion ml\_global

Sie können eine Skriptversion namens **ml\_global** erstellen, die sich von den anderen Skriptversionen unterscheidet. Wenn Sie eine Skriptversion namens ml\_global erstellen, definieren Sie sie einmalig. Die damit verknüpften Verbindungsskripten werden dann automatisch in allen Synchronisationen verwendet. ml\_global wird nie explizit als Skriptversion von einem Synchronisationsclient angegeben.

Wenn Sie ein Skript in der ml\_global-Skriptversion definieren und anschließend ein Skript für dasselbe Ereignis in der Skriptversion festlegen, die Sie für die Synchronisation angegeben haben, wird die angegebene Skriptversion verwendet. Skripten in der ml\_global-Skriptversion werden nur verwendet, wenn sie nicht in der primären Skriptversion definiert sind, die synchronisiert wird.

Die ml\_global-Skriptversion kann nur Verbindungsskripten enthalten. Sie ist optional und möglicherweise auch nicht hilfreich, wenn Sie nur eine Skriptversion verwenden.

## Skriptversionen zu konsolidierten Datenbanken hinzufügen

Eine Skriptversion kennzeichnet eine Gruppe von Skripten. Wenn Sie in Sybase Central arbeiten, müssen Sie Ihrer konsolidierten Datenbank einen Skriptversionsnamen hinzufügen, bevor Sie Verbindungsskripten hinzufügen können.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Kontext und Bemerkungen

Wenn Sie Skripten mit Systemprozeduren hinzufügen, wird mit dem Skript automatisch ein neuer Skriptversionsname hinzugefügt.

In Sybase Central darf nur eine Skriptversion pro Synchronisationsmodell verwendet werden. Sie erhält standardmäßig den Namen des Synchronisationsmodells.

Wenn Sie die Möglichkeit haben möchten, Schemaänderungen ohne Synchronisation vorzunehmen, müssen Sie mit SQL-Syntax eine Skriptversion zur Synchronisationssubskription hinzufügen.

Sie können eine Skriptversion mit dem gleichen Vorgang mithilfe von Systemprozeduren hinzufügen wie ein Verbindungs- oder Tabellenskript. Siehe [Systemprozeduren zum Hinzufügen oder Löschen von Skripten auf Seite 648](#).

Siehe „[Skriptversionen](#)“ auf Seite 254.

### Aufgabe

1. Klicken Sie im Menü **Ansicht** auf **Ordner**.
2. Erweitern Sie im linken Fensterausschnitt von Sybase Central Ihren MobiLink-Projektnamen und anschließend die konsolidierte Datenbank, mit der Sie arbeiten möchten. Ihre Verbindung mit der konsolidierten Datenbank basiert auf den Verbindungsinformationen, die bereitgestellt wurden, als Sie die konsolidierte Datenbank zu Ihrem Projekt hinzugefügt haben.
3. Klicken Sie auf den Ordner **Versionen** und klicken Sie auf **Datei » Neu » Version**.
4. Befolgen Sie die Anweisungen des **Assistenten zur Skriptversionserstellung**.

### Ergebnisse

Die Skriptversion wird erstellt.

## Skriptversionen aus konsolidierten Datenbanken entfernen

Gehen Sie folgendermaßen vor, um eine Skriptversion und ihre zugeordneten Skripten aus der konsolidierten Datenbank zu entfernen.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Aufgabe

1. Klicken Sie im Menü **Ansicht** auf **Ordner**.
2. Erweitern Sie im linken Fensterausschnitt von Sybase Central Ihren MobiLink-Projektnamen und anschließend die konsolidierte Datenbank, mit der Sie arbeiten möchten. Ihre Verbindung mit der konsolidierten Datenbank basiert auf den Verbindungsinformationen, die bereitgestellt wurden, als Sie die konsolidierte Datenbank zu Ihrem Projekt hinzugefügt haben.
3. Klicken Sie unter Ihrer konsolidierten Datenbank im linken Fensterausschnitt auf **Versionen**. Im rechten Fensterausschnitt wird eine Liste der Skriptversionen angezeigt.
4. Klicken Sie im rechten Fensterausschnitt mit der rechten Maustaste auf die Skriptversion, die Sie entfernen möchten, und wählen Sie **Löschen**.
5. Klicken Sie auf **Ja**.

#### Hinweis

Alle der Skriptversion zugeordneten Skripten werden ebenfalls gelöscht.

### Ergebnisse

Die Skriptversion und ihre zugeordneten Skripten werden aus der konsolidierten Datenbank entfernt.

## Erforderliche Skripten

Wenn Sie den MobiLink-Server ausführen, sind bestimmte Skripten erforderlich. Welche Skripten erforderlich sind, hängt davon ab, ob Sie eine Zweiweg-, reine Upload- oder eine reine Download-Synchronisation ausführen.

Bei einer bidirektionalen oder reinen Upload-Synchronisation erfordert MobiLink mindestens die folgenden Tabellenskripten:

- upload\_delete (bei Upload von gelöschten Zeilen mit SQL)
- upload\_insert (bei Upload von eingefügten Zeilen mit SQL)
- upload\_update (bei Upload von aktualisierten Zeilen mit SQL)
- Wenn Sie den Upload als direkte Zeilenbehandlung durchführen, erfordert MobiLink ein Skript für das Verbindungsereignis handle\_UploadData.

Bei einer bidirektionalen oder reinen Download-Synchronisation geht MobiLink davon aus, dass jede Tabelle in der Synchronisation sowohl ein download\_cursor-Skript als auch ein download\_delete\_cursor-Skript hat. Wenn Sie den Download als direkte Zeilenbehandlung durchführen, erfordert MobiLink die Angabe eines Verbindungsskripts handle\_DownloadData. Dieses Skript kann leer sein und Sie können den Download in jedem anderen Ereignis verarbeiten.

Alle erforderlichen Skripten müssen angegeben werden. Die Synchronisation wird abgebrochen, wenn ein erforderliches Skript fehlt. Wenn ein Skript ignoriert werden soll, verwenden Sie das Präfix --{ml\_ignore}. Siehe „[Ignorierte Skripten](#)“ auf Seite 262.

## Hinzufügungen und Löschungen von Skripten

Falls Sie den **Assistenten zum Erstellen eines Synchronisationsmodells** verwenden, werden der konsolidierten Datenbank automatisch Skripten hinzugefügt, sobald Sie das Deployment des Modells durchführen.

Wenn Sie Synchronisationsskripten außerhalb von Sybase Central erstellen, müssen Sie sie in die MobiLink-Systemtabellen in der konsolidierten Datenbank einfügen. Bei SQL-Skripten wird das gesamte Skript in der MobiLink-Systemtabelle gespeichert. Bei Java- oder .NET-Skripten wird der Methodename in der Systemtabelle registriert. Die Methode zum Speichern von Skripten und Methodennamen ist ähnlich.

Siehe „[Systemtabellen des MobiLink-Servers](#)“ auf Seite 4.

Wenn Sie Sybase Central benutzen, müssen Sie der Datenbank eine Skriptversion hinzufügen, bevor Sie einzelne Skripten hinzufügen können. Siehe „[Skriptversionen zu konsolidierten Datenbanken hinzufügen](#)“ auf Seite 256.

### Hinzufügen oder Löschen aller Typen von Skripten (Systemprozeduren)

Sie können Skripten mithilfe von gespeicherten Prozeduren, die beim Einrichten der konsolidierten Datenbank installiert werden, der konsolidierten Datenbank hinzufügen bzw. aus ihr löschen.

Weitere Hinweise zu den gespeicherten Prozeduren, mit denen Sie Skripten hinzufügen oder löschen können, finden Sie unter:

- „ml\_add\_connection\_script-Systemprozedur“ auf Seite 651
- „ml\_add\_table\_script-Systemprozedur“ auf Seite 668
- „ml\_add\_dnet\_connection\_script-Systemprozedur“ auf Seite 653
- „ml\_add\_dnet\_table\_script-Systemprozedur“ auf Seite 654
- „ml\_add\_java\_connection\_script-Systemprozedur“ auf Seite 655
- „ml\_add\_java\_table\_script-Systemprozedur“ auf Seite 656

## Verbindungsskripten hinzufügen

Gehen Sie folgendermaßen vor, um ein Verbindungsskript mit Sybase Central hinzuzufügen.

### Voraussetzungen

Wenn Sie Sybase Central benutzen, müssen Sie der Datenbank eine Skriptversion hinzufügen, bevor Sie einzelne Skripten hinzufügen können. Siehe [„Skriptversionen zu konsolidierten Datenbanken hinzufügen“ auf Seite 256](#).

### Aufgabe

1. Klicken Sie im Menü **Ansicht** auf **Ordner**.
2. Erweitern Sie im linken Fensterausschnitt von Sybase Central Ihren MobiLink-Projektnamen und anschließend die konsolidierte Datenbank, mit der Sie arbeiten möchten. Ihre Verbindung mit der konsolidierten Datenbank basiert auf den Verbindungsinformationen, die bereitgestellt wurden, als Sie die konsolidierte Datenbank zu Ihrem Projekt hinzugefügt haben.
3. Klicken Sie auf **Verbindungsskripten** und anschließend auf **Neu » Verbindungsskript**.
4. Befolgen Sie die Anweisungen des **Assistenten zum Erstellen von Verbindungsskripten**.

### Ergebnisse

Das Verbindungsskript wird erstellt.

## Verbindungsskripten löschen

Gehen Sie folgendermaßen vor, um ein Verbindungsskript mit Sybase Central zu löschen.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Aufgabe

1. Klicken Sie im Menü **Ansicht** auf **Ordner**.

2. Erweitern Sie im linken Fensterausschnitt von Sybase Central Ihren MobiLink-Projektnamen und anschließend die konsolidierte Datenbank, mit der Sie arbeiten möchten. Ihre Verbindung mit der konsolidierten Datenbank basiert auf den Verbindungsinformationen, die bereitgestellt wurden, als Sie die konsolidierte Datenbank zu Ihrem Projekt hinzugefügt haben.
3. Erweitern Sie **Verbindungsskripten**.
4. Rechtsklicken Sie auf ein Verbindungsskript und klicken Sie auf **Löschen**.
5. Klicken Sie auf **Ja**.

### Ergebnisse

Das Verbindungsskript wird gelöscht.

## Tabellenskripten hinzufügen

Gehen Sie folgendermaßen vor, um ein Tabellenskript mit Sybase Central hinzuzufügen.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Aufgabe

1. Klicken Sie im Menü **Ansicht** auf **Ordner**.
2. Erweitern Sie im linken Fensterausschnitt von Sybase Central Ihren MobiLink-Projektnamen und anschließend die konsolidierte Datenbank, mit der Sie arbeiten möchten. Ihre Verbindung mit der konsolidierten Datenbank basiert auf den Verbindungsinformationen, die bereitgestellt wurden, als Sie die konsolidierte Datenbank zu Ihrem Projekt hinzugefügt haben.
3. Erweitern Sie **Synchronisierte Tabellen**.
4. Rechtsklicken Sie auf die Tabelle und klicken Sie auf **Neu » Tabellenskript**.
5. Befolgen Sie die Anweisungen des **Assistenten zum Erstellen von Tabellenskripten**.

### Ergebnisse

Das Tabellenskript wird erstellt.

## Tabellenskripten löschen

Gehen Sie folgendermaßen vor, um ein Tabellenskript zu löschen.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

## Aufgabe

1. Klicken Sie im Menü **Ansicht** auf **Ordner**.
2. Erweitern Sie im linken Fensterausschnitt von Sybase Central Ihren MobiLink-Projektnamen und anschließend die konsolidierte Datenbank, mit der Sie arbeiten möchten. Ihre Verbindung mit der konsolidierten Datenbank basiert auf den Verbindungsinformationen, die bereitgestellt wurden, als Sie die konsolidierte Datenbank zu Ihrem Projekt hinzugefügt haben.
3. Erweitern Sie **Synchronisierte Tabellen**.
4. Erweitern Sie die Tabelle.
5. Rechtsklicken Sie auf die Tabelle und klicken Sie auf **Löschen**.
6. Klicken Sie auf **Ja**.

## Ergebnisse

Das Tabellenskript wird gelöscht.

## Skripten direkt einfügen

Es wird empfohlen, Skripten mithilfe von gespeicherten Prozeduren oder Sybase Central in die Systemtabellen einzufügen. In bestimmten seltenen Fällen müssen Sie jedoch eine INSERT-Anweisung verwenden, um die Skripten direkt einzufügen. Bestimmte Versionen einiger RDBMS-Systeme haben z.B. Längenbegrenzungen, die die Verwendung von gespeicherten Prozeduren erschweren.

Weitere Hinweise zu den MobiLink-Systemtabellen finden Sie unter [„Systemtabellen des MobiLink-Servers“ auf Seite 4](#).

Das Format der INSERT-Anweisungen, die für das direkte Einfügen von Skripten erforderlich sind, können im Quellcode für die gespeicherten Prozeduren `ml_add_connection_script` und `ml_add_table_script` eingesehen werden. Der Quellcode für diese gespeicherten Prozeduren befindet sich in den MobiLink-Setupskripten. Für jedes unterstützte RDBMS-System ist ein eigenes Setupskript vorhanden. Alle Setupskripten befinden sich im Verzeichnis `%SQLANY16%\MobiLink\Setup` und sind folgendermaßen benannt:

| Konsolidierte Datenbank    | Setupdatei         |
|----------------------------|--------------------|
| Adaptive Server Enterprise | <i>syncase.sql</i> |
| IBM DB2 LUW                | <i>syncdb2.sql</i> |
| Microsoft SQL Server       | <i>syncmss.sql</i> |
| MySQL                      | <i>syncmys.sql</i> |
| Oracle                     | <i>syncora.sql</i> |

| Konsolidierte Datenbank | Setupdatei        |
|-------------------------|-------------------|
| SQL Anywhere            | <i>syncsa.sql</i> |

## Ignorierte Skripten

Wenn ein Upload-Datenstrom Einfüge-, Aktualisierungs- oder Löschdaten für eine Tabelle enthält, für die in der konsolidierten Datenbank kein `upload_insert`-, `upload_update`- oder `upload_delete`-Skript vorhanden ist, oder wenn kein Download-Skript (`download_cursor`- oder `download_delete_cursor`-Skript) für die Tabelle vorhanden ist, gibt der MobiLink-Server Warnungen wegen der fehlenden Skripten aus und bricht die Synchronisation ab.

Die Warnungen können mit der Befehlsoption `-zwd` des MobiLink-Servers unterdrückt werden. Diese Option unterdrückt jedoch auch die Warnungen für alle Synchronisationstabellen.

Der MobiLink-Server behandelt alle Verbindungs- und Tabellenskripten, die das Präfix `--{ml_ignore}` enthalten, unterschiedlich. Er erkennt diese Skripten als absichtlich ignorierte Skripten. Wenn ein Upload-Datenstrom Einfüge-, Aktualisierungs- oder Löschdaten für eine Synchronisationstabelle enthält, für die ein `upload_insert`-, `upload_update`- oder `upload_delete`-Skript mit dem Präfix `--{ml_ignore}` vorhanden ist, werden diese Skripten vom MobiLink-Server nicht in der konsolidierten Datenbank ausgeführt. Stattdessen wird die Synchronisation fortgesetzt, ohne Fehlermeldungen oder Warnungen anzuzeigen. Die hochgeladenen Zeilen werden ignoriert.

Wenn eine Tabelle heruntergeladen wird, müssen die Skripten `download_cursor` und `download_delete_cursor` beide definiert werden. Um zu verhindern, dass Zeilen heruntergeladen werden, definieren Sie ggf. eines dieser Skripten oder beide als `--{ml_ignore}`.

## Skripten für den Upload von Zeilen

Um den MobiLink-Server anzuweisen, wie die von den entfernten Datenbanken empfangenen Upload-Daten verarbeitet werden sollen, definieren Sie Upload-Skripten. Sie schreiben jeweils separate Skripten für Zeilen, die in der entfernten Datenbank aktualisiert, eingefügt bzw. gelöscht werden sollen. Eine einfache Implementierung würde die entsprechenden Aktionen (Aktualisieren, Einfügen, Löschen) in der konsolidierten Datenbank ausführen.

Der MobiLink-Server liest die Upload-Daten in einer einzelnen Transaktion ein. Weitere Hinweise zum Upload-Vorgang finden Sie unter [„Ereignisse beim Upload“ auf Seite 279](#).

Techniken für den Upload von Zeilen in der .NET-Synchronisationslogik finden Sie unter [„Zeilen-Uploads und -Downloads“ auf Seite 548](#).

### Hinweise

- Die Skripten `begin_upload` und `end_upload` für die einzelnen entfernten Tabellen enthalten Programmlogik, die von den einzelnen zu aktualisierenden Zeilen unabhängig ist.

- Der Upload besteht aus einzelnen Einfügungen, Änderungen und Löschungen von Zeilen. Diese Aktionen werden normalerweise mit den Skripten `upload_insert`, `upload_update` und `upload_delete` ausgeführt.
- Um den Datenstrom für SQL Anywhere-Clients vorzubereiten, muss das Dienstprogramm `dbmlsync` Zugang zu allen Transaktionslogs haben, die seit der letzten erfolgreichen Synchronisation geschrieben wurden. Siehe „Transaktionslogdateien“ [*MobiLink - Clientadministration*].

## upload\_insert-Skripten

Der MobiLink-Server benutzt dieses Ereignis während der Verarbeitung des Uploads, um die in die entfernte Datenbank eingefügten Zeilen zu verarbeiten.

Die folgende INSERT-Anweisung wird in einem `upload_insert`-Skript verwendet.

```
INSERT INTO emp ( emp_id, emp_name )
VALUES ( { ml r.emp_id }, { ml r.emp_name } );
```

### Hinweise

- Wenn Fragezeichen anstelle von benannten Parametern als Platzhalter verwendet werden, akzeptieren die Ereignisse `upload_new_row_insert` und `upload_old_row_insert` zusätzlich die Parameter `remote_id` und `user_name`. Diese Parameter müssen sich vor der vollständigen Spaltenliste der Tabelle befinden.

### Siehe auch

- „Skripten für den Upload von Zeilen“ auf Seite 262
- „`upload_insert` (Tabellenereignis)“ auf Seite 441

## upload\_update-Skripten

Der MobiLink-Server benutzt dieses Ereignis während der Verarbeitung des Uploads zur Verarbeitung der in der entfernten Datenbank aktualisierten Zeilen. Die folgende UPDATE-Anweisung kann als `upload_update`-Skript für die Tabelle `emp` verwendet werden.

```
UPDATE emp
SET emp_name = {ml r.emp_name}
WHERE emp_id = {ml o.emp_id};
```

### Hinweise

- Wenn Fragezeichen anstelle von benannten Parametern als Platzhalter verwendet werden, kann die Anzahl der Parameter gleich einem der folgenden Werte sein (die Verwendung von Fragezeichen in SQL-Skripten ist veraltet):
  - Anzahl von Nicht-Primärschlüsselspalten + Primärschlüsselspalten
  - $2 * (\text{Anzahl von Nicht-Primärschlüsselspalten} + \text{Primärschlüsselspalten})$

Die Spaltenreihenfolge muss aus Nicht-Primärschlüsselspalten gefolgt von einem der Folgenden bestehen:

- Die Primärschlüsselspalten
- Alle Spalten

### Siehe auch

- „Skripten für den Upload von Zeilen“ auf Seite 262
- „upload\_update (Tabellenereignis)“ auf Seite 458

## upload\_delete-Skripten

Der MobiLink-Server benutzt dieses Ereignis während der Verarbeitung des Uploads zur Verarbeitung der aus der entfernten Datenbank gelöschten Zeilen. Die folgende Anweisung veranschaulicht, wie die upload\_delete-Anweisung verwendet wird.

```
DELETE FROM emp
WHERE emp_id = {ml r.emp_id};
```

### Hinweise

- Wenn Fragezeichen anstelle von benannten Parametern als Platzhalter verwendet werden, muss die Anzahl der Parameter gleich einem der folgenden Werte sein (die Verwendung von Fragezeichen in SQL-Skripten ist veraltet):
  - Anzahl von Primärschlüsselspalten
  - Anzahl aller Spalten

### Siehe auch

- „Skripten für den Upload von Zeilen“ auf Seite 262
- „upload\_delete (Tabellenereignis)“ auf Seite 435

## upload\_fetch-Skripten

Bei einem upload\_fetch-Skript handelt es sich um eine SELECT-Anweisung, die in der Tabelle der konsolidierten Datenbank einen Cursor definiert. Dieser Cursor wird verwendet, um die alten Werte der aktualisierten Zeilen, wie sie aus der entfernten Datenbank empfangen wurden, mit dem entsprechenden Wert in der konsolidierten Datenbank zu vergleichen. Auf diese Weise erkennt das Skript upload\_fetch-Konflikte während der Verarbeitung von Aktualisierungen.

Angenommen, eine synchronisierte Tabelle ist folgendermaßen definiert:

```
CREATE TABLE uf_example (
  pk1 integer NOT NULL,
  pk2 integer NOT NULL,
  val varchar(200),
  PRIMARY KEY( pk1, pk2 ));
```

Für diese Tabelle lautet dann ein mögliches upload\_fetch-Skript folgendermaßen:

```
SELECT pk1, pk2, val
FROM uf_example
WHERE pk1 = {ml r.pk1} and pk2 = {ml r.pk2}
```

Siehe „[upload\\_fetch \(Tabellenereignis\)](#)“ auf Seite 437.

Für den MobiLink-Server muss die WHERE-Klausel der Abfrage im upload\_fetch-Skript eine bestimmte Zeile in der konsolidierten Datenbank angeben, die auf Konflikte geprüft werden soll.

## Skripten für den Download von Zeilen

Es gibt zwei Skripten, die während der Downloadtransaktion für die Verarbeitung der einzelnen Tabellen verwendet werden können. Hierbei handelt es sich um das Skript download\_cursor, das Einfügungen und Aktualisierungen ausführt, und um das Skript download\_delete\_cursor, mit dem Löschvorgänge vorgenommen werden.

Diese Skripten sind entweder SELECT-Anweisungen oder Aufrufe von Prozeduren, die Ergebnismengen zurückgeben. Der MobiLink-Server führt ein Upload der Ergebnismenge des Skripts in die entfernte Datenbank durch. Anhand der Ergebnismenge des Skripts download\_cursor fügt der MobiLink-Client automatisch Zeilen ein bzw. aktualisiert sie. Grundlage für die Löschvorgänge ist das Ereignis download\_delete\_cursor.

Weitere Hinweise zur Verwendung von gespeicherten Prozeduren finden Sie unter „[Ergebnismengen aus Aufrufen an gespeicherte Prozeduren](#)“ auf Seite 138.

Der MobiLink-Server liest die Download-Daten in einer einzelnen Transaktion ein. Hinweise zum Download-Vorgang finden Sie unter „[Ereignisse beim Download](#)“ auf Seite 281.

### Hinweise

- Wie der Upload beginnt und endet auch der Download mit Verbindungsereignissen. Andere Ereignisse finden auf Tabellenebene statt.
- Die Skripten begin\_download und end\_download für die einzelnen entfernten Tabellen enthalten eine Logik, die von den einzelnen zu aktualisierenden Zeilen unabhängig ist.
- Der Download unterscheidet nicht zwischen Einfügungen und Änderungen. Das Skript, das dem Ereignis download\_cursor zugeordnet ist, ist eine SELECT-Anweisung, die die herunterzuladenden Zeilen festlegt. Der Client ermittelt, ob die Zeile vorhanden ist, und führt dann den entsprechenden Einfüge- oder Aktualisierungsvorgang aus.
- Bei zeitstempelbasierten Downloads können Sie den Parameter last\_table\_download so festlegen, dass nur Änderungen heruntergeladen werden, die seit der letzten Synchronisation vorgenommen wurden. Die SQL-Skripten download\_cursor oder download\_delete\_cursor könnten beispielsweise folgende Zeile enthalten:

```
WHERE Customer.last_modified >= {ml s.last_table_download}
```

Siehe „[Zeitpunkt des letzten Downloads in Skripten](#)“ auf Seite 109.

- Unter Umständen löscht der Client am Ende des Download-Prozesses automatisch einige Zeilen, um Verletzungen der referenziellen Integrität zu vermeiden.
- Wenn Sie die Einstellung von SendDownloadAck in ON ändern, wird die Download-Transaktion festgeschrieben, aber die Bestätigungsskripten werden erst ausgeführt, wenn die Bestätigung empfangen wird.

Standardmäßig hat SendDownloadAck die Einstellung OFF.

Siehe „Erweiterte Option SendDownloadAck (sa)“ [[MobiLink - Clientadministration](#)]  
„Synchronisationsparameter Send Download Acknowledgement“ [[UltraLite - Datenbankverwaltung](#)]  
„nonblocking\_download\_ack (Verbindungsereignis)“ auf Seite 401 und  
„publication\_nonblocking\_download\_ack (Verbindungsereignis)“ auf Seite 407.

### **Vorsicht**

Synchronisieren Sie keine Schattentabellen, die bei vorhergehenden Deployments erstellt wurden (z.B. dürfen Tabellen, die auf `_mod` oder `_del` enden, nicht synchronisiert werden). Diese Tabellen werden nur von der konsolidierten Datenbank benötigt, um geänderte oder gelöschte Zeilen zu protokollieren.

Siehe „Referenzielle Integrität und Synchronisation“ [[MobiLink - Erste Orientierung](#)].

## download\_cursor-Skripten

Sie schreiben download\_cursor-Skripten, um Daten aus der konsolidierten Datenbank in eine entfernte Datenbank herunterzuladen. Entsprechend schreiben Sie download\_delete\_cursor-Skripten für den Download von zu löschenden Zeilen aus der entfernten Datenbank. Sie müssen für jede Tabelle in der entfernten Datenbank, für die Sie ein Download von Änderungen durchführen wollen, beide Skripten schreiben. Es steht Ihnen frei, auch andere Skripten zu verwenden, um den Download-Prozess anzupassen.

- Jedes download\_cursor-Skript muss eine SELECT-Anweisung oder einen Prozeduraufruf mit einer SELECT-Anweisung enthalten.
- Wenn Sie keine Zeilen herunterladen möchten, definieren Sie das Skript als `--{ml_ignore}`. Alternativ dazu können Sie mit der Systemprozedur `ml_add_missing_dnld_scripts` fehlende Download-Skripten als ignoriert definieren.
- Das download\_cursor-Skript muss alle Spalten auswählen, die mit den Spalten in der entsprechenden Tabelle der entfernten Datenbank übereinstimmen. Die Spalten in der konsolidierten Datenbank können andere Namen haben als die entsprechenden Spalten in der entfernten Datenbank. Ihre Datentypen müssen jedoch kompatibel sein.

### **Siehe auch**

- „`ml_add_missing_dnld_scripts`-Systemprozedur“ auf Seite 660

### **Beispiel**

Das folgende Skript kann als download\_cursor-Skript für eine entfernte Tabelle dienen, die Mitarbeiterdaten enthält. Das Skript liest die Daten aller Mitarbeiter ein.

```
SELECT emp_id, emp_fname, emp_lname
FROM employee;
```

Der MobiLink-Server übergibt einigen Skripten spezifische Parameter. Der MobiLink-Server ersetzt den Wert des Parameters, bevor er die Anweisung in der konsolidierten Datenbank ausführt. Die Verwendung von Fragezeichen ist in SQL-Skripten veraltet. Aus dem folgenden Skript wird ersichtlich, wie Sie benannte Parameter verwenden können.

```
CALL ml_add_table_script(
    'Lab',
    'ULOrder',
    'download_cursor',
    'SELECT o.order_id, o.cust_id, o.prod_id, o.emp_id, o.disc, o.quant,
o.notes, o.status
    FROM ULOrder o
    WHERE o.last_modified >= {ml s.last_table_download}
    AND o.emp_name = {ml s.username}' )
```

## Hinweise

- Alle Cursorskripten müssen die Spalten genau in der Reihenfolge auswählen, in der diese in der entfernten Datenbank definiert sind. Wenn die Spaltennamen bzw. die Tabellenstruktur in der konsolidierten Datenbank anders sind, dann müssen die Spalten in der für die entfernte Datenbank bzw. die Referenzdatenbank korrekten Reihenfolge ausgewählt werden. Die Spalten werden den Spalten der entfernten Datenbank in der Reihenfolge der SELECT-Anweisung zugeordnet.
- Zeilenwerte können aus einer einzelnen Tabelle oder aus einem Join mehrerer Tabellen ausgewählt werden.
- Auch braucht die entfernte Tabelle nicht denselben Namen zu haben wie die Tabelle in der konsolidierten Datenbank. Der Name der entfernten Tabelle muss dabei nicht unbedingt im Skript enthalten sein. Der Name der entfernten Tabelle wird durch einen Eintrag in der MobiLink-Systemtabelle ml\_table festgelegt. In Sybase Central können Sie die entfernten Tabellen zusammen mit den dazugehörigen Skripten anzeigen.

## Siehe auch

- [„download\\_cursor \(Tabellenereignis\)“ auf Seite 329](#)
- [„Partitionierte Zeilen in entfernten Datenbanken“ auf Seite 113](#)
- [„download\\_delete\\_cursor-Skripten“ auf Seite 267](#)

## download\_delete\_cursor-Skripten

Sie schreiben download\_delete\_cursor-Skripten, um Zeilen aus einer entfernten Datenbank zu löschen. Sie müssen für jede Tabelle in der entfernten Datenbank, die in den Download einbezogen ist, eines dieser Skripten schreiben. Wenn Sie keine Zeilen löschen möchten, definieren Sie die einzelnen Skripten als -- {ml\_ignore}. Alternativ dazu können Sie mit der Systemprozedur ml\_add\_missing\_dnld\_scripts fehlende Download-Skripten als ignoriert definieren. Siehe [„ml\\_add\\_missing\\_dnld\\_scripts-Systemprozedur“ auf Seite 660](#).

Es ist nicht möglich, Zeilen aus der konsolidierten Datenbank einfach zu löschen, damit sie in den entfernten Datenbanken nicht mehr enthalten sind. Sie müssen die Primärschlüssel für gelöschte Zeilen

berücksichtigen und diese Primärschlüssel mit Ihrem `download_delete_cursor` auswählen. Für die Erreichung dieses Ziels gibt es zwei Methoden:

- **Logische Löschungen** Löschen Sie die Zeile nicht physisch in der konsolidierten Datenbank. Verwenden Sie stattdessen eine Statusspalte, die protokolliert, ob Zeilen gültig sind. Dies vereinfacht den `download_delete_cursor`. Der `download_cursor` und andere Anwendungen müssen jedoch möglicherweise geändert werden, um die Statusspalte zu erkennen und zu verwenden. Wenn Sie eine Spalte mit der letzten Änderung verwenden, die den Löschzeitpunkt enthält, und außerdem den letzten Download-Zeitpunkt für die einzelnen entfernten Datenbanken protokollieren, können Sie die Zeile physisch löschen, wenn alle Downloadzeiten der entfernten Datenbanken nach dem Zeitpunkt der Löschung liegen.
- **Schattentabelle** Für jede Tabelle, für die Sie Löschungen verfolgen wollen, erstellen Sie eine Schattentabelle mit zwei Spalten, die den Primärschlüssel für die Tabelle und einen Zeitstempel enthalten. Erstellen Sie einen Trigger, der den Primärschlüssel und den Zeitstempel in die Schattentabelle einfügt, wenn eine Zeile gelöscht wird. Ihr `download_delete_cursor` kann dann auf diese Schattentabelle zugreifen. Wie bei logischen Löschungen können Sie die Zeile aus der Schattentabelle löschen, wenn alle entfernten Datenbanken die entsprechenden Daten eingelesen haben.

Der MobiLink-Server löscht Zeilen aus der entfernten Datenbank, indem er Primärschlüsselwerte aus der konsolidierten Datenbank auswählt und sie an die entfernte Datenbank weitergibt. Wenn die Werte mit den Werten eines Primärschlüssels in der entfernten Datenbank übereinstimmen, wird die betreffende Zeile gelöscht.

- Jedes `download_delete_cursor`-Skript, mit dem gelöschte Zeilen heruntergeladen werden sollen, muss eine `SELECT`-Anweisung oder einen Aufruf einer gespeicherten Prozedur enthalten, die eine Ergebnismenge zurückgibt. Der MobiLink-Server verwendet diese Anweisung, um einen Cursor in der konsolidierten Datenbank zu definieren.
- Wenn Sie immer wollen, dass ein `download_delete_cursor`-Skript keine Zeilen auswählt, definieren Sie das Skript als `--{ml_ignore}`.
- Diese Anweisung muss alle Spalten auswählen, die mit den Primärschlüsselspalten in der entsprechenden Tabelle der entfernten Datenbank übereinstimmen. Die Spalten in der konsolidierten Datenbank können andere Namen haben als die entsprechenden Spalten in der entfernten Datenbank. Ihre Datentypen müssen jedoch kompatibel sein.
- Die Werte müssen dabei genau in der Reihenfolge ausgewählt werden, in der die entsprechenden Spalten der entfernten Datenbank definiert sind. Es handelt sich dabei um die Reihenfolge, in der die Spalten in der `CREATE TABLE`-Anweisung aufgeführt sind, mit der die Tabelle erstellt wurde, und nicht um die Reihenfolge in der Anweisung zum Definieren des Primärschlüssels.
- Wenn Sie einen übergeordneten Datensatz in der entfernten Datenbank mit `download_delete_cursor` löschen, werden die untergeordneten Datensätze automatisch mitgelöscht. Dies gilt nicht für BlackBerry-Geräte.

Weitere Hinweise zum Löschen von untergeordneten Datensätzen finden Sie unter „[Referenzielle Integrität und Synchronisation](#)“ [*MobiLink - Erste Orientierung*].

## Alle Zeilen in einer Tabelle löschen

Wenn MobiLink ein `download_delete_cursor`-Skript für eine Zeile findet, die nur NULL-Werte enthält, werden alle Daten in der entfernten Tabelle gelöscht. Die Anzahl der NULL-Werte im `download_delete_cursor`-Skript kann der Anzahl der Primärschlüsselspalten oder der Gesamtzahl der Spalten in der Tabelle entsprechen.

Das folgende `download_delete_cursor`-SQL-Skript löscht z.B. alle Zeilen in einer Tabelle, in der zwei Primärschlüsselspalten enthalten sind. Dieses Beispiel ist für SQL Anywhere-, Adaptive Server Enterprise- und Microsoft SQL Server-Datenbanken geeignet.

```
SELECT NULL, NULL
```

In konsolidierten IBM DB2 LUW- und Oracle-Datenbanken müssen Sie eine Durchgangstabelle festlegen, um NULL auszuwählen. Bei IBM DB2 LUW 9.5 und 9.7 können Sie die folgende Syntax verwenden:

```
SELECT CAST( NULL AS INTEGER ), CAST( NULL AS INTEGER ) FROM SYSIBM.SYSDUMMY1
```

Bei konsolidierten Oracle-Datenbanken können Sie folgende Syntax verwenden:

```
SELECT NULL, NULL FROM DUAL
```

## Siehe auch

Weitere Hinweise zur Verwendung von `download_delete_cursor`-Skripten finden Sie unter:

- „`download_cursor` (Tabellenereignis)“ auf Seite 329
- „`download_delete_cursor` (Tabellenereignis)“ auf Seite 331
- „Löschungen“ auf Seite 132
- „Synchronisation von Löschungen vorübergehend stoppen“ auf Seite 133
- „STOP SYNCHRONIZATION DELETE-Anweisung [MobiLink]“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]
- „Partitionierte Zeilen in entfernten Datenbanken“ auf Seite 113
- „Snapshot-Synchronisation“ auf Seite 111

## Beispiele

Folgendes Beispiel ist ein `download_delete_cursor`-Skript für eine entfernte Tabelle, die Mitarbeiterdaten enthält. Der MobiLink-Server verwendet diese SQL-Anweisung, um den Löschcursor festzulegen. Dieses Skript löscht Informationen über Bestellungen, die zur Zeit der Ausführung des Skripts sowohl in der konsolidierten als auch in der entfernten Datenbank vorhanden sind.

```
SELECT order_id
FROM UOrder
```

Das `download_delete_cursor`-Skript akzeptiert die Parameter `last_table_download` und `username`. Aus dem folgenden Skript ist ersichtlich, wie die einzelnen Parameter eingesetzt werden können, um die Auswahl einzugrenzen.

```
SELECT order_id
FROM UOrder
WHERE last_modified >= {ml s.last_table_download}
AND status = 'Approved'
AND user_name = {ml s.username}
```

Eine andere Strategie besteht darin, der Clientanwendung zu gestatten, selbst die Zeilen zu löschen. Allerdings ist dies nur dann möglich, wenn die nicht benötigten Zeilen durch eine Regel bestimmt werden. Zeilen können zum Beispiel einen Zeitstempel mit einem Ablaufdatum enthalten. Bevor Sie die Zeilen in der entfernten Datenbank löschen, verwenden Sie die Anweisung STOP SYNCHRONIZATION DELETE, um zu verhindern, dass diese Löschvorgänge bei der nächsten Synchronisation übertragen werden. Führen Sie START SYNCHRONIZATION DELETE sofort anschließend aus, wenn andere Löschvorgänge normal synchronisiert werden sollen.

### Hinweise

- Das download\_delete\_cursor-Skript muss Primärschlüsselspalten in derselben Reihenfolge enthalten, in der sie in der entfernten Datenbank festgelegt sind.
- Sie können die in allen MobiLink-Clients integrierte Prüfung der referenziellen Integrität verwenden, um Zeilen effizient zu löschen, indem Sie nur die übergeordneten Zeilen löschen. Siehe „[Referenzielle Integrität und Synchronisation](#)“ [*MobiLink - Erste Orientierung*].

## Skripten für die Fehlerbehandlung

Ein Fehler in einem Synchronisationsskript tritt auf, wenn ein Vorgang im Skript während der Ausführung durch den MobiLink-Server fehlschlägt. Bei SQL-Skripten gibt das DBMS einen SQLCODE und eine Fehlermeldung an den MobiLink-Server zurück, um die Fehlerart anzuzeigen. Jedes DBMS einer konsolidierten Datenbank verfügt über eigene SQLCODE-Werte und Meldungen. Standardmäßig setzt der MobiLink-Server die Transaktion in der konsolidierten Datenbank zurück, protokolliert den Fehler und bricht die Synchronisation ab.

Wenn während des Aufrufs eines SQL-Datenskripts ein Fehler auftritt, startet der MobiLink-Server das Ereignis handle\_error oder handle\_odbc\_error. Wenn diese Fehlerbehandlungsskripten definiert sind, ruft der MobiLink-Server sie auf und übergibt mehrere Parameter mit Informationen über die Art und den Kontext des Fehlers. Ein Parameter ist ein Ausgabewert namens action\_code, der den MobiLink-Server anweist, wie er auf den Fehler reagieren soll. Der Parameter action\_code weist den MobiLink-Server an, entweder den Fehler zu ignorieren oder die Synchronisation abubrechen.

Die Fehlerbehandlungsskripten werden **nicht** für alle SQL-Fehler aufgerufen. Nur Datenskripten verursachen das Aufrufen der Fehlerbehandlungsskripten. Wenn Fehler in Nicht-Datenskripten auftreten, setzt der MobiLink-Server die Transaktion in der konsolidierten Datenbank zurück, protokolliert den Fehler und bricht die Synchronisation ab.

Falls das DBMS Ihrer konsolidierten Datenbank die Verarbeitungsroutine bei Ausnahmefehlern unterstützt, sollten Sie in Betracht ziehen, diese statt der Fehlerbehandlungsskripten zu verwenden, besonders wenn Sie bestimmte Fehler in Datenskripten ignorieren müssen. Die Verarbeitungsroutine bei Ausnahmefehlern liefert fast immer eine bessere Performance als die Fehlerbehandlungsskripten.

Wenn ein handle\_error- oder handle\_odbc\_error-Skript selbst einen Fehler verursacht, setzt der MobiLink-Server die Transaktion in der konsolidierten Datenbank zurück, protokolliert den Fehler und bricht die Synchronisation ab.

### Maßnahmen für die Fehlerbehandlung

Folgende Aktionen sollten Sie in ein Fehlerbehandlungsskript aufnehmen:

- Ignorieren des Fehlers, aber Protokollieren in einer Audit-Tabelle.
- Anweisung an den MobiLink-Server die Synchronisation zurückzusetzen.
- Senden einer E-Mail-Alarmnachricht.

### **Mehrere Fehler in einer einzelnen SQL-Anweisung behandeln**

ODBC lässt mehrere Fehler pro SQL-Anweisung zu, und einige RDBMS-Systeme nutzen diese Funktion. Microsoft SQL Server kann zum Beispiel zwei Fehler für eine einzige Anweisung melden. Der erste ist der tatsächliche Fehler, während der zweite gewöhnlich eine Meldung ist, die darüber informiert, warum die aktuelle Anweisung beendet wurde.

Wenn von einer SQL-Anweisung mehrere Fehler verursacht werden, wird das Skript `handle_error` einmal pro Fehler aufgerufen. Der MobiLink-Server verwendet den härtesten Aktionscode (also den numerisch höchsten), um die vorzunehmende Aktion zu bestimmen. Dies gilt auch für das `handle_error`-Skript.

Verursacht das Skript `handle_error` selbst einen SQL-Fehler, wird der Standardaktionscode (3000) angenommen.

### **Siehe auch**

- „`handle_error` (Verbindungsereignis)“ auf Seite 375
- „`handle_odbc_error` (Verbindungsereignis)“ auf Seite 379
- „`report_error` (Verbindungsereignis)“ auf Seite 410
- „`report_odbc_error` (Verbindungsereignis)“ auf Seite 414

## **Fehlermeldungen**

Da Fehler standardmäßig ein Zurücksetzen in der konsolidierten Datenbank verursachen, ist es aufgrund dieses Vorgangs schwierig, innerhalb der konsolidierten Datenbank ein Protokoll mit Fehlern und den dazugehörigen Lösungen zu erstellen. Mithilfe der Ereignisse `report_error` und `report_odbc_error` können Sie eine korrekte Aufzeichnung der Fehler in benutzerdefinierten Skripten erstellen, da diese auf einer anderen Datenbankverbindung aufgerufen werden als die Synchronisation. Diese Fehlerberichtsskripten werden sofort nach den Fehlerbehandlungsskripten aufgerufen. Direkt im Anschluss erfolgt eine Festschreibung.

Die Fehlerberichtsskripten werden für alle SQL-Fehler in benutzerdefinierten Skripten aufgerufen. Wenn Fehler in benutzerdefinierten Skripten auftreten, setzt der MobiLink-Server die Transaktion in der konsolidierten Datenbank zurück, protokolliert den Fehler und bricht die Synchronisation ab.

Falls das DBMS Ihrer konsolidierten Datenbank ein Out-of-Band-Verfahren (Verfahren außerhalb der aktuellen Datenbankverbindung) für die Berichterstellung über Aktivitäten aus SQL unterstützt, sollten Sie in Betracht ziehen, dieses Verfahren statt der von MobiLink definierten Fehlerberichtsskripten zu verwenden.

Siehe:

- „handle\_error (Verbindungsereignis)“ auf Seite 375
- „handle\_odbc\_error (Verbindungsereignis)“ auf Seite 379
- „report\_error (Verbindungsereignis)“ auf Seite 410
- „report\_odbc\_error (Verbindungsereignis)“ auf Seite 414

### Beispiel

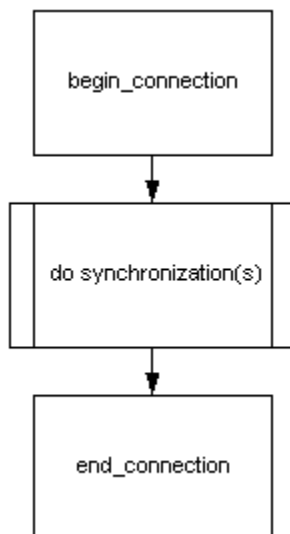
Das folgende report\_error-Skript besteht nur aus einer INSERT-Anweisung und fügt die Skriptparameter zusammen mit dem Datum und der Uhrzeit in eine Tabelle ein. Das Skript schreibt diese Änderung allerdings nicht fest, da dies automatisch vom MobiLink-Server übernommen wird.

```
INSERT INTO errors
VALUES(
    CURRENT DATE,
    {ml s.action_code},
    {ml s.error_code},
    {ml s.error_message},
    {ml s.username},
    {ml s.table} );
```

## Synchronisationsereignisse

### Überblick über MobiLink-Ereignisse

Wenn eine Synchronisationsanforderung abgesetzt wird und der MobiLink-Server entscheidet, dass eine neue Verbindung zur konsolidierten Datenbank erstellt werden muss, wird das begin\_connection-Ereignis ausgelöst und die Synchronisation beginnt.



Nach der Synchronisation wird die Verbindung zur konsolidierten Datenbank in einen Verbindungspool gesetzt und MobiLink wartet erneut auf eine Synchronisationsanforderung. Wenn jedoch eine weitere

Synchronisationsanforderung für dieselbe Version eintrifft, wickelt MobiLink die nächste Synchronisationsanforderung mit derselben Verbindung ab. Bevor eine Verbindung aus dem Verbindungspool gelöscht wird, muss das Ereignis `end_connection` ausgelöst werden.

Jede Synchronisation umfasst viele Ereignisse. Die meisten Ereignisse werden von der Transaktion organisiert, die sie enthält.

### Transaktionen

Bei jeder Synchronisation kann es zu folgenden Transaktionen kommen.

- Authentifizierung
- Synchronisation beginnen
- Upload
- Download vorbereiten
- Download
- Synchronisation beenden
- Nicht blockierende Downloadbestätigung

Zudem sind zwei Verbindungstransaktionen zulässig. Nach dem Herstellen der Verbindung zur konsolidierten Datenbank wird sofort eine `begin_connection`-Transaktion ausgeführt und beim Trennen der Verbindung eine `end_connection`-Transaktion.

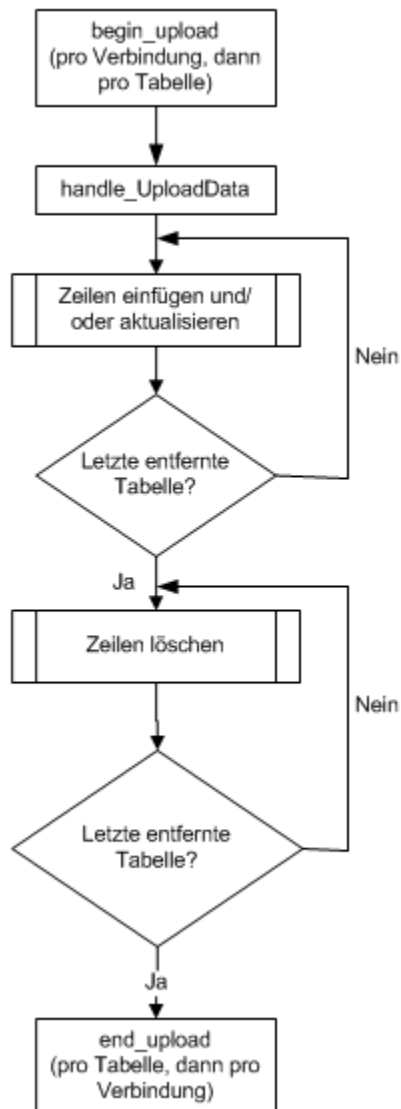
Die wichtigsten Phasen einer Synchronisation sind die Upload- und Downloadtransaktionen. Die in Upload- und Downloadtransaktionen enthaltenen Ereignisse werden im Folgenden beschrieben.

### Die Upload-Transaktion

Die Uploadtransaktion übernimmt Änderungen, die per Upload von einer entfernten Datenbank übermittelt werden.

Das Ereignis `begin_upload` kennzeichnet den Anfang der Uploadtransaktion. Die Uploadtransaktion ist ein zweiteiliger Prozess. Zunächst werden Einfügungen und Aktualisierungen für alle entfernten Tabellen per Upload übermittelt, dann die Löschungen für alle entfernten Tabellen.

#### Uploadtransaktion



Das Ereignis end\_upload kennzeichnet das Ende der Uploadtransaktion.

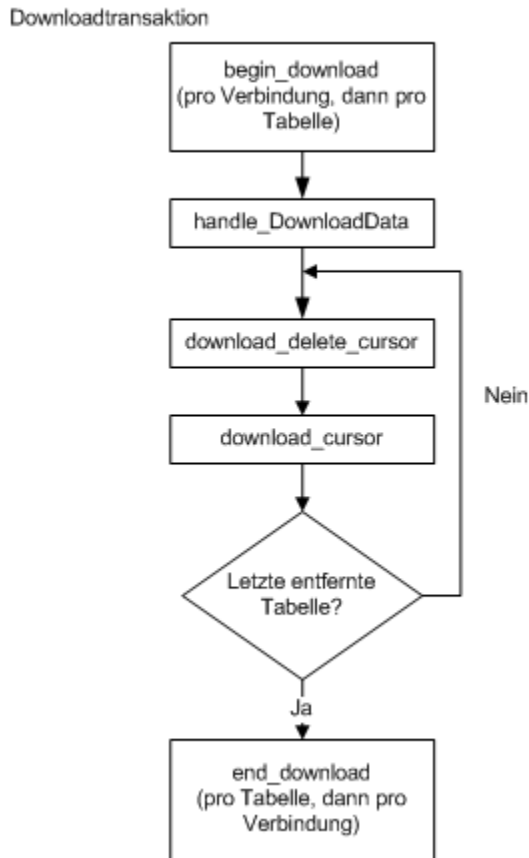
Mit der Option dbmsync -tu können Sie mehrere Upload-Transaktionen angeben.

Siehe „[Skripten für den Upload von Zeilen](#)“ auf Seite 262.

#### Die Download-Transaktion

Die Downloadtransaktion ruft Zeilen aus der konsolidierten Datenbank ab, Sie beginnt mit dem begin\_download-Ereignis.

Die Downloadtransaktion ist ein zweiteiliger Prozess. Für jede Tabelle erfolgt erst der Download der Löschungen und dann der aktualisierten und eingefügten Zeilen. Das end\_download-Ereignis beendet die Downloadtransaktion.



Siehe „Skripten für den Download von Zeilen“ auf Seite 265.

### Die nicht blockierende Downloadbestätigungstransaktion

Die nicht blockierende Downloadbestätigungstransaktion wird nur durchgeführt, wenn eine Downloadbestätigung empfangen wird. Diese Transaktion hat zwei Funktionen. Die Skripten `publication_nonblocking_download_ack` und `nonblocking_download_ack` werden in dieser Transaktion ausgeführt. Sie vereinfachen die Protokollierung des Downloadstatus. Außerdem werden die Download-Zeitstempel in den MobiLink-Systemtabellen bei dieser Transaktion aktualisiert.

Diese Transaktion wird möglicherweise nicht auf derselben Datenbankverbindung ausgeführt wie die anderen Ereignisse für die Zielsynchronisation, sodass in dieser Transaktion keine Variablen auf Verbindungsebene referenziert werden können.

### Ereignisüberblick in Pseudocode

Der folgende Pseudocode bietet einen Überblick über die Abfolge, in der Ereignisse und die gleichnamigen Skripten aufgerufen werden. In dieser Darstellung des MobiLink-Ereignismodells wird davon ausgegangen, dass eine volle Synchronisation (nicht nur für Upload oder nur für Download) ohne Fehler durchgeführt wird.

### Hinweise

- Normalerweise besteht die Standardaktion darin, nichts zu tun, wenn Sie kein Skript für ein bestimmtes Ereignis definiert haben.
- Die Ereignisse `begin_connection` und `end_connection` sind **Ereignisse auf Verbindungsebene**. Sie sind unabhängig von allen Synchronisationen und haben keine Parameter.
- Einige Ereignisse werden einmal pro Synchronisation für jede zu synchronisierende Tabelle aufgerufen. Die diesen Ereignissen zugeordneten Skripten werden als **Skripten auf Verbindungsebene** bezeichnet.

Jede Tabelle kann über eigene Tabellenskripten verfügen; Sie können aber auch Tabellenskripten schreiben, die von mehreren Tabellen gemeinsam genutzt werden.

- Einige Ereignisse, wie etwa `begin_synchronization`, treten sowohl auf der Verbindungsebene als auch auf der Tabellenebene ein. Sie können für diese Ereignisse sowohl Verbindungs- als auch Tabellenskripten bereitstellen.
- Die COMMIT-Anweisung illustriert, wie der Synchronisationsprozess in unterschiedliche Transaktionen aufgeteilt wird.

#### Vorsicht

Ihre SQL-Synchronisationsskripten bzw. die Prozeduren oder Skripten, die von Ihren SQL-Synchronisationsskripten aufgerufen werden, dürfen keine impliziten oder expliziten COMMIT- oder ROLLBACK-Anweisungen enthalten. COMMIT- und ROLLBACK-Anweisungen innerhalb eines SQL-Skripts ändern den transaktionsbasierten Ablauf der Synchronisationsschritte. Falls Sie diese Anweisungen benutzen, kann MobiLink die Integrität der Daten bei einer Systemstörung nicht garantieren.

## Komplettes MobiLink-Ereignismodell

-----  
MobiLink complete event model.

#### Legend:

- `// This is a comment.`
- `<name>`  
The pseudocode for `<name>` is listed separately in a later section, under a banner:  
-----  
name  
-----
- `VariableName <- value`  
Assign the given value to the given variable name.  
Variable names are in mixed case.

```

- event_name
  If you have defined a script for the given event name,
  it is invoked.
-----

CONNECT to consolidated database
begin_connection_autocommit
begin_connection
COMMIT
for each synchronization request with
  the same script version {
  <synchronize>
}
end_connection
COMMIT
DISCONNECT from consolidated database

-----

synchronize
-----

<authenticate>
<begin_synchronization>
<upload>
<prepare_for_download>
<download>
<end_synchronization>

-----

authenticate
-----

Status <- 1000
UseDefaultAuthentication <- TRUE
if( authenticate_user script is defined ) {
  UseDefaultAuthentication <- FALSE
  TempStatus <- authenticate_user
  if( TempStatus > Status ) {
    Status <- TempStatus
  }
}
if( authenticate_user_hashed script is defined ) {
  UseDefaultAuthentication <- FALSE
  TempStatus <- authenticate_user_hashed
  if( TempStatus > Status ) {
    Status <- TempStatus
  }
}
if( authenticate_parameters script is defined )
{
  TempStatus <- authenticate_parameters
  if( TempStatus > Status ) {
    Status <- TempStatus
  }
}
if( UseDefaultAuthentication ) {
  if( the user exists in the ml_user table ) {
    if( ml_user.hashed_password column is not NULL ) {
      if( password matches ml_user.hashed_password ) {
        Status <- 1000
      } else {
        Status <- 4000
      }
    } else {
      Status <- 1000
    }
  }
}

```

```
    }
  } else if( -zu+ was on the command line ) {
    Status <- 1000
  } else {
    Status <- 4000
  }
}
if( Status >= 3000 ) {
  // Abort the synchronization.
} else {
  // UserName defaults to MobiLink user name
  // sent from the remote.
  if( modify_user script is defined ) {
    UserName <- modify_user
    // The new value of UserName is later passed to
    // all scripts that expect the MobiLink user name.
  }
}
COMMIT

-----
begin_synchronization
-----

begin_synchronization // Connection event.
for each table being synchronized {
  begin_synchronization // Call the table level script.
}
for each publication being synchronized {
  begin_publication
}
COMMIT

-----
end_synchronization
-----

for each publication being synchronized {
  if( begin_publication script was processed ) {
    end_publication
  }
}
for each table being synchronized {
  if( begin_synchronization table script was processed ) {
    end_synchronization // Table event.
  }
}
if( begin_synchronization connection script was processed ) {
  end_synchronization // Connection event.
}
for each table being synchronized {
  synchronization_statistics // Table event.
}
synchronization_statistics // Connection event.
for each table being synchronized {
  time_statistics // Table event.
}
time_statistics // Connection event.

COMMIT
```

Details zur Verarbeitung von Uploads finden Sie unter [„Ereignisse beim Upload“](#) auf Seite 279.

Details zur Verarbeitung von Downloads finden Sie unter [„Ereignisse beim Download“ auf Seite 281](#).

## Ereignisse beim Upload

Der folgende Pseudocode veranschaulicht, wie Upload-Ereignisse und Upload-Skripten angestoßen werden. Der Pseudocode verwendet die folgenden Konventionen:

- **Ein Skript ist als echtes Skript definiert** Das bedeutet, dass das Skript als echtes Skript definiert ist, das während der Synchronisation in der konsolidierten Datenbank ausgeführt wird.
- **Ein Skript ist als ignoriertes Skript definiert** Das bedeutet, dass das Skript mit "--{ml\_ignore}" als ignoriertes Skript definiert ist.
- **Ein Skript ist definiert** Das bedeutet, dass das Skript als echtes oder ignoriertes Skript definiert ist.
- **Ein Skript ist nicht definiert** Das bedeutet, dass für das Ereignis überhaupt keine Skripten definiert sind. Sie müssen es als ignoriertes Skript definieren, wenn es sich um ein erforderliches Skript handelt, Sie es aber nicht verwenden möchten.

Diese Ereignisse finden Sie an der Upload-Stelle im vollständigen Ereignismodell ausgeführt. Siehe [„Überblick über MobiLink-Ereignisse“ auf Seite 272](#).

### Überblick über den Upload

```
-----
upload
-----

begin_upload // Connection event
for each table being synchronized {
    begin_upload // Table event
}
handle_UploadData
for each table being synchronized {
    begin_upload_rows
    for each uploaded INSERT or UPDATE for this table {
        if( INSERT ) {
            <upload_inserted_row>
        }
        if( UPDATE ) {
            <upload_updated_row>
        }
    }
    end_upload_rows
}
for each table being synchronized IN REVERSE ORDER {
    begin_upload_deletes
    for each uploaded DELETE for this table {
        <upload_deleted_row>
    }
    end_upload_deletes
}
For each table being synchronized {
    if( begin_upload table script was processed ) {
        end_upload // Table event
    }
}
```

```

}
if( begin_upload connection script was processed ) {
    end_upload // Connection event
}
for each table being synchronized {
    upload_statistics // Table event.
}
upload_statistics // Connection event.

COMMIT

```

## Upload-Einfügungen

```

-----
<upload_inserted_row>
-----
// NOTES:
// - Only table scripts for the current table are involved.

if( upload_insert script is real ) {
    upload_insert
} else if( handle_uploadData script is real or
    upload_insert script is defined as an ignored script ) {
    // Ignore the insert. (Only ignored in SQL, possibly handled by
    handle_uploadData.)
} else {
    error
}

```

## Upload-Aktualisierungen

```

-----
upload_updated_row
-----
// NOTES:
// - Only table scripts for the current table are involved.
// - Both the old (original) and new rows are uploaded for
//   each update.

ConflictsAreExpected <- (
    upload_new_row_insert script is defined or
    upload_old_row_insert script is defined )
Conflicted <- FALSE

if( upload_update script is real ) {
    if( upload_fetch or upload_fetch_column_conflict script is real ) {
        if( ConflictsAreExpected ) {
            FETCH using upload_fetch INTO current_row
            if( current_row <> old row ) {
                Conflicted <- TRUE
            } else {
                upload_update
            }
        } else {
            error
        }
    } else if( upload_fetch and upload_fetch_column_conflict scripts are not
    defined ) {
        if( ConflictsAreExpected ) {
            error
        } else {
            // No conflict detection and resolution by the MobiLink server
            // The upload_update script should handle conflict detection and
            resolution
        }
    }
}

```

```

        upload_update
    }
    } else {
        // the upload_fetch script cannot defined as an ignored script
        error
    }
    } else if( handle_uploadData script is defined or upload_update script is
defined as an ignored script ) {
        // Ignore the upload update (Only ignored in SQL, possibly handled by
handle_uploadData.)
    } else {
        error
    }
}
if( Conflicted ) {
    if( upload_old_row_insert script is real ) {
        upload_old_row_insert
    } else if( upload_old_row_insert script is defined as ignored script ) {
        // Ignore the old value
    } else {
        error
    }
    if( upload_new_row_insert script is real ) {
        upload_new_row_insert
    } else if( upload_new_row_insert script is defined as ignored script ) {
        // Ignore the new value
    } else {
        error
    }
    if( no error ) {
        resolve_conflict
    }
}
}

```

## Upload-Löschungen

```

-----
upload_deleted_row
-----
// NOTES:
// - Only table scripts for the current table are involved.

if( upload_delete is real ) {
    upload_delete
} else if( handle_UploadData script is real or
upload_delete script is defined as an ignored script ) {
    // Ignore this delete. (Only ignored in SQL, possibly handled by
handle_uploadData.)
} else {
    error
}

```

## Ereignisse beim Download

Der folgende Pseudocode bietet einen Überblick über die Abfolge, in der Download-Ereignisse und die gleichnamigen Skripten aufgerufen werden.

Diese Ereignisse finden Sie an der Download-Stelle im vollständigen Ereignismodell ausgeführt (siehe [„Überblick über MobiLink-Ereignisse“ auf Seite 272](#)).

```

-----
prepare_for_download

```

```

-----
generate_next_last_download_timestamp
modify_last_download_timestamp
fetch the next download timestamp from consolidated
prepare_for_download

-----
download
-----

begin_download // Connection event.
for each table being synchronized {
    begin_download // Table event.
}
    handle_DownloadData
    for each table being synchronized {
        begin_download_deletes
        for each row in download_delete_cursor {
            if( all primary key columns are NULL ) {
                send TRUNCATE to remote
            } else {
                send DELETE to remote
            }
        }
        end_download_deletes
        begin_download_rows
        for each row in download_cursor {
            send INSERT ON EXISTING UPDATE to remote
        }
        end_download_rows
    }
    modify_next_last_download_timestamp
    for each table being synchronized {
        if( begin_download table script was processed ) {
            end_download // Table event
        }
    }
    if( begin_download connect script was processed ) {
        end_download // Connection event
    }
    for each table being synchronized {
        download_statistics // Table event.
    }
    download_statistics // Connection event.

COMMIT

```

## Hinweise

- Der Download-Datenstrom unterscheidet nicht zwischen Einfügungen und Änderungen. Das Skript, das dem download\_cursor-Ereignis zugeordnet ist, ist eine SELECT-Anweisung, die die herunterzuladenden Zeilen festlegt. Der Client ermittelt, ob die Zeile vorhanden ist, und führt dann den entsprechenden Einfüge- oder Aktualisierungsvorgang aus.
- Am Ende des Download-Prozesses löscht der Client automatisch Zeilen, die die referenzielle Integrität verletzen.

Siehe „Referenzielle Integrität und Synchronisation“ [[MobiLink - Erste Orientierung](#)].

## Datenskripten

Skripten, die Zeilendaten direkt verarbeiten, heißen Datenskripten. Alle anderen Skripten sind Nicht-Datenskripten. Die Unterscheidung zwischen Datenskripten und Nicht-Datenskript ist manchmal wichtig. Beispielsweise können nur Datenskripten die benannten Parameter für Spaltenwerte referenzieren.

Mit den nachfolgenden Ereignissen sind Datenskripten verknüpft:

- „download\_cursor (Tabellenereignis)“
- „download\_delete\_cursor (Tabellenereignis)“
- „handle\_UploadData (Verbindungsereignis)“
- „handle\_DownloadData (Verbindungsereignis)“
- „upload\_delete (Tabellenereignis)“
- „upload\_fetch (Tabellenereignis)“
- „upload\_fetch\_column\_conflict (Tabellenereignis)“
- „upload\_insert (Tabellenereignis)“
- „upload\_new\_row\_insert (Tabellenereignis)“
- „upload\_old\_row\_insert (Tabellenereignis)“
- „upload\_update (Tabellenereignis)“

### Java- und .NET-Datenskripten mit SQL-Rückgabe (entfernt)

Ab Version 16 wurde die Möglichkeit, mit Java- und .NET-Skriptenlogik Zeichenfolgen zurückzugeben, die vom MobiLink-Server als SQL-Skripten interpretiert werden, in allen Skripten entfernt. Wenn Ihre Skripten Änderungen in der konsolidierten Datenbank veranlassen müssen, sollten sie dies direkt aus Java oder .NET tun.

Nachfolgend finden Sie ein Beispiel dafür, wie ein Skript aktualisiert werden kann. Das erste Beispiel verwendet SQL, das zweite Beispiel nicht.

```
public String beginDownloadConnection(
    Timestamp ts,
    String user )
    throws java.sql.SQLException
{
    doSomeWork( ts, user );
    return( "CALL do_some_sql( {ml s.last_download}, {ml s.username} )" );
}

public void beginDownloadConnection(
    Timestamp ts,
    String user )
    throws java.sql.SQLException
{
    doSomeWork( ts, user );
}
```

```
Connection conn = DBConnectionContext.getConnection();
PreparedStatement stmt = conn.prepareStatement( "CALL
do_some_sql( ?,? )" );
stmt.setTimestamp( 1, ts );
stmt.setString( 2, user );
stmt.executeUpdate();
}
```

**authenticate\_file\_transfer (Verbindungsereignis)**

Implementiert benutzerdefinierte Authentifizierung für Dateiübertragungen mit dem Dienstprogramm mlfiletransfer oder der Methode MLFileDownload.

**Parameter**

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „SQL-Java-Datentypen“ auf Seite 466 und „SQL-.NET-Datentypen“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reihenfolge (veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.file_authentication_code     | INTEGER. Erforderlich. Dies ist ein INOUT-Parameter. Er zeigt den Gesamterfolg der Authentifizierung an.<br><br>Wenn dieser Wert im Bereich von 1000 bis 1999 liegt, ist die Datenübertragung erlaubt. Wenn dieser Wert im Bereich von 2000 bis 2999 liegt, ist die Datenübertragung nicht erlaubt.                                                                                                                                                                                                                                                                                                       | 1                              |
| s.filename                     | VARCHAR(128). Erforderlich. Dieser INOUT-Parameter ist der Name der übertragenen Datei, die authentifiziert werden soll. Geben Sie keine Pfadangaben mit an und verwenden Sie nicht Auslassungszeichen (drei Punkte), Komma, Schrägstrich (/) oder Backslash (\). Die Datei muss sich entweder im Stammübertragungsverzeichnis befinden, das Sie mit der mlsrv16-Option -ftr oder -ftru angegeben haben, oder in einem der automatisch erstellten Unterverzeichnisse. Falls dies nicht explizit festgelegt wurde, ist der Standardwert der Dateiname, der dem MobiLink-Server vom Client übergeben wurde. | 2                              |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                            | Reihenfolge (veraltet für SQL) |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                                                                                                                                                | 3                              |
| s.subdir                       | VARCHAR(128). Dieser optionale INOUT-Parameter setzt das Unterverzeichnis für den Speicherort der zu übertragenden Dateien. Um das Stammverzeichnis zu verwenden, setzen Sie diese Option auf Null. Diese Option darf nicht Auslassungszeichen (drei Punkte), Komma, Schrägstrich (/) oder Backslash (\) enthalten. Wenn dies nicht explizit festgelegt wurde, wird standardmäßig remote_key verwendet. | Nicht anwendbar                |
| s.remote_key                   | VARCHAR(128). Optionaler IN-Parameter für die Angabe eines entfernten Schlüssels für die Dateiübertragung.                                                                                                                                                                                                                                                                                              | Nicht anwendbar                |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben.                                                                                                                            | Nicht anwendbar                |

### Bemerkungen

Der MobiLink-Server führt dieses Ereignis aus, bevor eine Dateiübertragung mit dem Dienstprogramm mlfiletransfer oder der Methode MLFileDownload zugelassen wird. Er wird ausgeführt, nachdem der Benutzer unter Verwendung einer regulären Authentifizierung authentifiziert wurde. Wenn dieses Skript nicht definiert ist, ist die Dateiübertragung zulässig.

Die Methode MLFileDownload kann nur von UltraLite-Clients benutzt werden.

### Siehe auch

- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „mlsrv16-Option -ftr “ auf Seite 63
- „mlsrv16-Option -ftru “ auf Seite 64
- „MobiLink-Dienstprogramm für die Dateiübertragung (mlfiletransfer)“ [*MobiLink - Clientadministration*]
- MLFileDownload-Methode [UltraLite Embedded SQL] [*UltraLite - C- und C++-Programmierung*]
- UltraLite: „MobiLink-Dateiübertragungen“ [*UltraLite - Datenbankverwaltung*]
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

## authenticate\_file\_upload (Verbindungsereignis)

Implementiert benutzerdefinierte Authentifizierung für Dateiübertragungen mit dem Dienstprogramm mlfiletransfer oder der Methode MLFileUpload.

## Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Reihenfolge (veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.file_authentication_code     | INTEGER. Erforderlich. Dies ist ein INOUT-Parameter. Er zeigt den Gesamterfolg der Authentifizierung an.<br><br>Wenn dieser Wert im Bereich von 1000 bis 1999 liegt, ist die Datenübertragung erlaubt. Wenn dieser Wert im Bereich von 2000 bis 2999 liegt, ist die Datenübertragung nicht erlaubt.                                                                                                                                                                                                                                                                                                       | 1                              |
| s.filename                     | VARCHAR(128). Erforderlich. Dieser INOUT-Parameter ist der Name der übertragenen Datei, die authentifiziert werden soll. Geben Sie keine Pfadangaben mit an und verwenden Sie nicht Auslassungszeichen (drei Punkte), Komma, Schrägstrich (/) oder Backslash (\). Die Datei muss sich entweder im Stammübertragungsverzeichnis befinden, das Sie mit der mlsrv16-Option -ftr oder -ftru angegeben haben, oder in einem der automatisch erstellten Unterverzeichnisse. Falls dies nicht explizit festgelegt wurde, ist der Standardwert der Dateiname, der dem MobiLink-Server vom Client übergeben wurde. | 2                              |
| s.file_size                    | INTEGER. Dieser optionale IN-Parameter kann verwendet werden, um die Dateigröße für den Upload zu begrenzen.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 3                              |
| s.subdir                       | VARCHAR(128). Dieser optionale INOUT-Parameter setzt das Unterverzeichnis für den Speicherort der zu übertragenden Dateien. Um das Stammverzeichnis zu verwenden, setzen Sie diese Option auf Null. Diese Option darf nicht Auslassungszeichen (drei Punkte), Komma, Schrägstrich (/) oder Backslash (\) enthalten. Wenn dies nicht explizit festgelegt wurde, wird standardmäßig remote_key verwendet.                                                                                                                                                                                                   | Nicht anwendbar                |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.remote_key                   | VARCHAR(128). Optionaler IN-Parameter für die Angabe eines entfernten Schlüssels für die Dateiübertragung.                                                                                                                                                                   | Nicht anwendbar                |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

### Bemerkungen

Der MobiLink-Server führt dieses Ereignis aus, bevor eine Dateiübertragung mit dem Dienstprogramm mlfiletransfer oder der Methode MLFileUpload zugelassen wird. Er wird ausgeführt, nachdem der Benutzer unter Verwendung einer regulären Authentifizierung authentifiziert wurde. Wenn dieses Skript nicht definiert ist, ist die Dateiübertragung zulässig.

Die Methode MLFileUpload kann nur von UltraLite-Clients benutzt werden.

### Siehe auch

- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „mlsrv16-Option -ftr “ auf Seite 63
- „mlsrv16-Option -ftru “ auf Seite 64
- „MobiLink-Dienstprogramm für die Dateiübertragung (mlfiletransfer)“ [*MobiLink - Clientadministration*]
- MLFileUpload-Methode [UltraLite Embedded SQL] [*UltraLite - C- und C++-Programmierung*]
- UltraLite: „MobiLink-Dateiübertragungen“ [*UltraLite - Datenbankverwaltung*]
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

## authenticate\_parameters (Verbindungsereignis)

Empfängt Werte von der entfernten Datenbank, die für eine Authentifizierung über Benutzer-ID bzw. Kennwort hinaus verwendet werden können. Mit den Werten können Sie ebenfalls jede Synchronisation beliebig anpassen.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine

nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.authentication_status        | INTEGER. Dies ist ein INOUT-Parameter.                                                                                                                                                                                                                                       | 1                              |
| s.authentication_message       | VARCHAR(1024). Dies ist ein INOUT-Parameter. Stellt eine Authentifizierungsnachricht bereit.                                                                                                                                                                                 | Nicht anwendbar                |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                              |
| a.N (mindestens einer)         | VARCHAR(4000). Benannte Parameter können z.B. a . 1 a . 2 sein.                                                                                                                                                                                                              | 3 oder höher                   |

## Parameterbeschreibung

- **authentication\_status** Der Parameter authentication\_status ist erforderlich. Zeigt den gesamten Erfolg der Authentifizierung an und kann einen der folgenden Werte annehmen:

| Zurückgegebener Wert | authentication_status | Beschreibung                                                        |
|----------------------|-----------------------|---------------------------------------------------------------------|
| V <= 1999            | 1000                  | Authentifizierung erfolgreich                                       |
| 1000 <= V <= 2999    | 2000                  | Authentifizierung erfolgreich, aber das Kennwort läuft in Kürze ab. |
| 2000 <= V <= 3999    | 3000                  | Authentifizierung fehlgeschlagen, da das Kennwort abgelaufen ist.   |
| 4000 <= V <= 4999    | 4000                  | Authentifizierung fehlgeschlagen.                                   |

| Zurückgegebener Wert | authentication_status | Beschreibung                                                                                                                                     |
|----------------------|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 5000 <= V <= 5999    | 5000                  | Authentifizierung nicht möglich, da die entfernte ID bereits verwendet wird. Versuchen Sie die Synchronisation später nochmals.                  |
| 6000 <= V            | 4000                  | Wenn der zurückgegebene Wert größer als 5999 ist, interpretiert ihn MobiLink als einen Rückgabewert von 4000 (Authentifizierung fehlgeschlagen). |

- **authentication\_message** Dieser optionale Parameter stellt eine Authentifizierungsnachricht bereit.

Dieser benannte Parameter wird auf NULL initialisiert, bevor er zum ersten Mal von einem Benutzerauthentifizierungsskript verwendet wird. Die zurückgegebene Nachricht wird anschließend an das nächste Benutzerauthentifizierungsskript übergeben, sofern das Skript diesen benannten Parameter verwendet. Die letzte Nachricht wird in den Zeichensatz der entfernten Datenbank konvertiert.

Wenn während der Ausführung der Benutzerauthentifizierungsskripten kein Fehler aufgetreten ist, wird diese Nachricht anschließend vor dem Verarbeiten des Upload-Datenstroms vom MobiLink-Server an den Client gesendet, und zwar unabhängig vom Benutzerauthentifizierungsstatus.

Diese Nachricht wird an den Client gesendet, auch wenn die Benutzerauthentifizierung fehlgeschlagen ist.

- **remote\_ID** Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.

Siehe „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [[MobiLink - Clientadministration](#)].

- **script\_version** Dieser optionale Parameter gibt an, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben.
- **username** Dieser Parameter ist der MobiLink-Benutzername. VARCHAR(128).
- **remote a.N** Der n-te Authentifizierungsparameter, der vom entfernten Client gesendet wurde.

## Bemerkungen

Die Anzahl der entfernten Parameter muss der vom authenticate\_parameters-Skript erwarteten Anzahl entsprechen, da sonst ein Fehler gemeldet wird. Ein Fehler tritt auch auf, wenn vom Client Parameter gesendet werden und kein Skript für dieses Ereignis vorliegt.

Sie können Zeichenfolgen (oder Parameter in der Form von Zeichenfolgen) sowohl von SQL Anywhere- als auch von UltraLite-Clients senden. Dadurch kann die Authentifizierung über eine Benutzer-ID und ein Kennwort hinausgehen. Außerdem bedeutet dies, dass Sie während der Authentifizierung Ihre

Synchronisation basierend auf dem Wert von Parametern anpassen und dies vor der Synchronisation tun können. Diese Parameter können auch von Synchronisationsskripten referenziert werden.

Der MobiLink-Server führt dieses Ereignis beim Start jeder einzelnen Synchronisation aus. Es wird in derselben Transaktion ausgeführt wie das `authenticate_user`-Ereignis.

Sie können dieses Ereignis verwenden, um das integrierte MobiLink-Authentifizierungsverfahren durch ein angepasstes Verfahren zu ersetzen. Als Grund dafür ist denkbar, dass Sie möglicherweise das Authentifizierungsverfahren Ihres DBMS verwenden oder Funktionen implementieren wollen, die im integrierten MobiLink-Verfahren nicht vorhanden sind.

Wenn die Skripten `authenticate_user` oder `authenticate_user_hashed` aufgerufen werden und einen Fehler zurückgeben, wird dieses Ereignis nicht aufgerufen.

SQL-Skripten für das Ereignis `authenticate_parameters` müssen als gespeicherte Prozedur implementiert werden.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „Authentifizierungsparameter“ auf Seite 253
- „MobiLink-Benutzer“ [*MobiLink - Clientadministration*]
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Angepasste Benutzerauthentifizierung“ [*MobiLink - Clientadministration*]
- „authenticate\_user (Verbindungsereignis)“ auf Seite 291
- „authenticate\_user\_hashed (Verbindungsereignis)“ auf Seite 296
- „begin\_synchronization (Verbindungsereignis)“ auf Seite 313
- `dbmlsync: „dbmlsync-Option -ap“` [*MobiLink - Clientadministration*]
- UltraLite: „Synchronisationsparameter Authentication Parameters“ [*UltraLite - Datenbankverwaltung*] und „Synchronisationsparameter Number of Authentication Parameters“ [*UltraLite - Datenbankverwaltung*]

### Beispiele

Bei entfernten UltraLite-Datenbanken können Sie die Parameter mit den Feldern `num_auth_parms` und `auth_parms` in der `ul_sync_info`-Struktur übergeben. `num_auth_parms` ist die Anzahl der Parameter, von 0 bis 255. `auth_parms` ist ein Zeiger auf ein Array von Zeichenfolgen. Damit die Zeichenfolgen nicht als normaler Text angezeigt werden können, werden sie auf dieselbe Weise verschleiert wie Kennwörter. Wenn `num_auth_parms` gleich 0 ist, setzen Sie `auth_parms` auf NULL. Nachstehend finden Sie ein Beispiel für die Übergabe von Parametern in UltraLite:

```
ul_char * Params[ 3 ] = { UL_TEXT( "param1" ),
    UL_TEXT( "param2" ), UL_TEXT( "param3" ) };

...
info.num_auth_parms = 3;
info.auth_parms = Params;
```

Bei entfernten SQL Anywhere-Datenbanken können Sie Parameter mit der `dbmlsync-Option -ap` in eine kommasetrennten Liste übergeben. Die folgende Befehlszeile übergibt z.B drei Parameter:

```
dbmlsync -ap "param1,param2,param3"
```

In diesem Beispiel könnte das `authenticate_parameters`-Skript wie folgt aussehen:

```
CALL my_auth_parm (
  {ml s.authentication_status},
  {ml s.remote_id},
  {ml s.username},
  {ml a.1},
  {ml a.2},
  {ml a.3}
)
```

## authenticate\_user (Verbindungsereignis)

Angepasste Benutzerauthentifizierung implementieren.

### Parameter

In der folgenden Tabelle gibt die Beschreibung den SQL-Datentyp an. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge (veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.authentication_status        | INTEGER. Dies ist ein INOUT-Parameter.                                                                                                  | 1                              |
| s.authentication_message       | VARCHAR(1024). Dies ist ein INOUT-Parameter. Stellt eine Authentifizierungsnachricht bereit.                                            | Nicht anwendbar                |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                | 2                              |
| s.password                     | VARCHAR(128). Das Kennwort für Authentifizierungszwecke. Wenn der Benutzer kein Kennwort eingibt, ist sein Wert NULL.                   | 3                              |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.new_password                 | VARCHAR(128). Das neue Kennwort, sofern es zum Zurücksetzen des Kennworts verwendet wird. Wenn der Benutzer sein Kennwort nicht ändert, ist dieser Wert NULL.                                                                                                                | 4                              |
| s.new_remote_id                | VARCHAR(128). Die entfernte MobiLink-ID, wenn die entfernte ID in der konsolidierten Datenbank neu ist. Wenn die entfernte ID nicht neu ist, ist der Wert NULL.                                                                                                              |                                |
| s.new_username                 | VARCHAR(128). Der MobiLink-Benutzername, wenn der Benutzername in der konsolidierten Datenbank neu ist. Wenn der Benutzername nicht neu ist, ist der Wert NULL.                                                                                                              |                                |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

### Standardaktion

Integriertes MobiLink-Benutzerauthentifizierungsverfahren verwenden.

### Bemerkungen

Der MobiLink-Server führt dieses Ereignis beim Start jeder einzelnen Synchronisation aus. Es wird in einer Transaktion vor der Transaktion begin\_synchronization ausgeführt.

Sie können dieses Ereignis verwenden, um das integrierte MobiLink-Authentifizierungsverfahren durch ein angepasstes Verfahren zu ersetzen. Als Grund dafür ist denkbar, dass Sie möglicherweise das Authentifizierungsverfahren Ihres DBMS verwenden oder Funktionen implementieren wollen, die im integrierten MobiLink-Verfahren nicht vorhanden sind, z.B. ein Kennwortablaufdatum oder eine Mindestlänge für das Kennwort.

Im Ereignis authenticate\_user werden folgende Parameter verwendet:

- **authentication\_status** Der Parameter authentication\_status ist erforderlich. Zeigt den gesamten Erfolg der Authentifizierung an und kann einen der folgenden Werte annehmen:

| Zurückgegebener Wert | authentication_status | Beschreibung                                                        |
|----------------------|-----------------------|---------------------------------------------------------------------|
| V <= 1999            | 1000                  | Authentifizierung erfolgreich                                       |
| 2000 <= V <= 2999    | 2000                  | Authentifizierung erfolgreich, aber das Kennwort läuft in Kürze ab. |

| Zurückgegebener Wert | authentication_status | Beschreibung                                                                                                                    |
|----------------------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------|
| 3000 <= V <= 3999    | 3000                  | Authentifizierung fehlgeschlagen: Kennwort abgelaufen.                                                                          |
| 4000 <= V <= 4999    | 4000                  | Authentifizierung fehlgeschlagen.                                                                                               |
| 5000 <= V <= 5999    | 5000                  | Authentifizierung nicht möglich, da die entfernte ID bereits verwendet wird. Versuchen Sie die Synchronisation später nochmals. |
| 6000 <= V            | 4000                  | Wenn der zurückgegebene Wert größer als 5999 ist, interpretiert ihn MobiLink als einen Rückgabewert von 4000.                   |

Der Wert wird an den Client gesendet und kann daher verwendet werden, um das Authentifizierungsverhalten auf Clientseite anzupassen.

Siehe:

- „Authentication Value-Synchronisationsparameter“ [[UltraLite - Datenbankverwaltung](#)]
- „sp\_hook\_dbmlsync\_upload\_end“ [[MobiLink - Clientadministration](#)]

- **authentication\_message** Dieser optionale Parameter stellt eine Authentifizierungsnachricht bereit.

Dieser benannte Parameter wird auf NULL initialisiert, bevor er zum ersten Mal von einem Benutzerauthentifizierungsskript verwendet wird. Die zurückgegebene Nachricht wird anschließend an das nächste Benutzerauthentifizierungsskript übergeben, sofern das Skript diesen benannten Parameter verwendet. Die letzte Nachricht wird in den Zeichensatz der entfernten Datenbank konvertiert.

Wenn während der Ausführung der Benutzerauthentifizierungsskripten kein Fehler aufgetreten ist, wird diese Nachricht anschließend vor dem Verarbeiten des Upload-Datenstroms vom MobiLink-Server an den Client gesendet, und zwar unabhängig vom Benutzerauthentifizierungsstatus.

Diese Nachricht wird an den Client gesendet, auch wenn die Benutzerauthentifizierung fehlgeschlagen ist.

- **username** Dieser optionale Parameter ist der MobiLink-Benutzername.

Siehe „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [[MobiLink - Clientadministration](#)].

- **remote\_id** Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.
- **password** Dieser optionale Parameter zeigt das Kennwort für Authentifizierungszwecke an. Wenn der Benutzer kein Kennwort eingibt, ist dies NULL.

- **new\_password** Dieser optionale Parameter gibt ein neues Kennwort an. Wenn der Benutzer sein Kennwort nicht ändert, ist dieser Parameter NULL.
- **new\_remote\_id** Dieser optionale Parameter gibt eine neue entfernte ID an. Wenn die entfernte ID nicht neu ist, ist dies NULL.
- **new\_username** Dieser optionale Parameter gibt einen neuen Benutzernamen an. Wenn der Benutzername nicht neu ist, ist dies NULL.
- **script\_version** Dieser optionale Parameter gibt an, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben.

SQL-Skripten für das Ereignis `authenticate_user` müssen als gespeicherte Prozedur implementiert werden.

Wenn die beiden Authentifizierungsskripten definiert sind und beide Skripten unterschiedliche `authentication_status`-Codes zurückgeben, wird der höhere verwendet.

Das Skript `authenticate_user` wird in einer Transaktion zusammen mit allen Authentifikationskripten ausgeführt. Diese Transaktion schreibt immer fest.

Es gibt vordefinierte Skripten, die Sie für das `authenticate_user`-Ereignis verwenden können, um die Authentifizierung mit LDAP-, IMAP- und POP3-Servern zu vereinfachen.

Siehe „Authentifizierung bei externen Servern“ [[MobiLink - Clientadministration](#)].

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „MobiLink-Benutzer“ [[MobiLink - Clientadministration](#)]
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [[MobiLink - Clientadministration](#)]
- „Angepasste Benutzerauthentifizierung“ [[MobiLink - Clientadministration](#)]
- „Authentifizierung bei externen Servern“ [[MobiLink - Clientadministration](#)]
- „authenticate\_user\_hashed (Verbindungsereignis)“ auf Seite 296
- „authenticate\_parameters (Verbindungsereignis)“ auf Seite 287
- „modify\_user (Verbindungsereignis)“ auf Seite 398
- „begin\_synchronization (Verbindungsereignis)“ auf Seite 313

### SQL-Beispiel

Ein typisches `authenticate_user`-Skript ist ein Aufruf einer gespeicherten Prozedur. Die Reihenfolge der Parameter im Aufruf muss mit der oben angegebenen Reihenfolge übereinstimmen. Das folgende Beispiel weist das Ereignis mithilfe von `ml_add_connection_script` der gespeicherten Prozedur `my_auth` zu.

```
CALL ml_add_connection_script(  
  'ver1', 'authenticate_user', 'call my_auth ( {ml s.authentication_status},  
  {ml s.username} )'  
)
```

Die folgende gespeicherte Prozedur von SQL Anywhere benutzt nur den Benutzernamen für die Authentifizierung. Sie prüft kein Kennwort. Die Prozedur stellt nur sicher, dass der angegebene Benutzername eine der Mitarbeiter-IDs in der Tabelle `ULEmployee` ist.

```

CREATE PROCEDURE my_auth( inout @auth_status int, in @user_name
varchar(128) )
BEGIN
  IF EXISTS
    ( SELECT * FROM ulemmployee
      WHERE emp_id = @user_name )
  THEN
    MESSAGE 'OK' type info to client;
    SET @auth_status = 1000;
  ELSE
    MESSAGE 'Not OK' type info to client;
    SET @auth_status = 4000;
  END IF
END

```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `authenticateUser` als das Skript für das Ereignis `authenticate_user`, wenn die Skriptversion `ver1` synchronisiert wird. Diese Syntax gilt für konsolidierte SQL Anywhere-Datenbanken.

```

CALL ml_add_java_connection_script(
  'ver1', 'authenticate_user',
  'ExamplePackage.ExampleClass.authenticateUser'
)

```

Im Folgenden sehen Sie die Java-Beispielmethode `authenticateUser`. Sie ruft Java-Methoden auf, die das Benutzerkennwort prüfen und gegebenenfalls ändern.

```

public void authenticateUser(
  ianywhere.ml.script.InOutInteger authStatus,
  String user,
  String pwd,
  String newPwd )
  throws java.sql.SQLException {
  // A real authenticate_user handler would
  // handle more authentication code states.
  _curUser = user;
  if( checkPwd( user, pwd ) ) {
    // Authentication successful.
    if( newPwd != null ) {
      // Password is being changed.
      if( changePwd( user, pwd, newPwd ) ) {
        // Authentication OK and password change OK.
        // Use custom code.
        authStatus.setValue( 1001 );
      } else {
        // Authentication OK but password
        // change failed. Use custom code.
        java.lang.System.err.println( "user: "
          + user + " pwd change failed!" );
        authStatus.setValue( 1002 );
      }
    } else {
      authStatus.setValue( 1000 );
    }
  } else {
    // Authentication failed.
    authStatus.setValue( 4000 );
  }
}

```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens AuthUser als das Skript für das Verbindungsereignis authenticate\_user, wenn die Skriptversion ver1 synchronisiert wird. Diese Syntax gilt für konsolidierte SQL Anywhere-Datenbanken.

```
CALL ml_add_dnet_connection_script(  
    'ver1', 'authenticate_user',  
    'TestScripts.Test.AuthUser'  
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode AuthUser. Sie ruft .NET-Methoden auf, die das Benutzerkennwort prüfen und gegebenenfalls ändern.

```
namespace TestScripts {  
    public class Test {  
        string _curUser = null;  
        public void AuthUser(  
            ref int authStatus,  
            string user,  
            string pwd,  
            string newPwd ) {  
            // A real authenticate_user handler would  
            // handle more authentication code states.  
            _curUser = user;  
            if( CheckPwd( user, pwd ) ) {  
                // Authentication successful.  
                if( newPwd != null ) {  
                    // Password is being changed.  
                    if( ChangePwd( user, pwd, newPwd ) ) {  
                        // Authentication OK and password change OK.  
                        // Use custom code.  
                        authStatus = 1001;  
                    } else {  
                        // Authentication OK but password  
                        // change failed. Use custom code.  
                        System.Console.WriteLine( "user: "  
                            + user + " pwd change failed!" );  
                        authStatus = 1002;  
                    }  
                } else {  
                    authStatus = 1000 ;  
                }  
            } else {  
                // Authentication failed.  
                authStatus = 4000;  
            }  
        }  
    }  
}
```

Ein ausführlicheres Beispiel für ein authenticate\_user-Skript in C# in .NET finden Sie unter „[.NET-Synchronisationsbeispiel](#)“ auf Seite 550.

## authenticate\_user\_hashed (Verbindungsereignis)

Angepasstes Benutzerauthentifizierungsverfahren implementieren

## Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.authentication_status        | INTEGER. Dies ist ein INOUT-Parameter.                                                                                                                                                                                                                                       | 1                              |
| s.authentication_message       | VARCHAR(1024). Dies ist ein INOUT-Parameter. Stellt eine Authentifizierungsnachricht bereit.                                                                                                                                                                                 | Nicht anwendbar                |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                              |
| s.hashed_password              | BINARY(32). Wenn der Benutzer kein Kennwort eingibt, ist sein Wert NULL.                                                                                                                                                                                                     | 3                              |
| s.hashed_new_password          | BINARY(32). Wenn mit diesem Ereignis nicht das Benutzerkennwort geändert wird, ist dieser Wert NULL.                                                                                                                                                                         | 4                              |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

## Standardaktion

Integriertes MobiLink-Benutzerauthentifizierungsverfahren verwenden.

## Bemerkungen

Dieses Ereignis ist dasselbe wie `authenticate_user`, abgesehen von den Kennwörtern, die dasselbe Hash-Format haben wie diejenigen aus der Spalte `ml_user.hashed_password`. Durch die Übergabe der Kennwörter im Hash-Format wird die Sicherheit erhöht.

Ein Einweg-Hash wird benutzt. Ein Einweg-Hash übernimmt ein Kennwort und konvertiert es in eine Bytesequenz, die (im Wesentlichen) für jedes mögliche Kennwort einmalig ist. Mit dem Einweg-Hash kann die Kennwortauthentifizierung vorgenommen werden, ohne dass das tatsächliche Kennwort in der konsolidierten Datenbank gespeichert werden muss.

Aufgrund von inkrementellen Verbesserungen an der Qualität der Hashes in allen MobiLink-Versionen kann dieses Skript während einer Authentifizierungssequenz für einen Benutzer mehrmals aufgerufen werden.

Wenn die beiden Authentifizierungsskripten `authenticate_user` und `authenticate_user_hashed` definiert sind und beide Skripten unterschiedliche `authentication_status`-Codes zurückgeben, wird der höhere Wert verwendet.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „MobiLink-Benutzer“ [*MobiLink - Clientadministration*]
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Angepasste Benutzerauthentifizierung“ [*MobiLink - Clientadministration*]
- „authenticate\_user (Verbindungsereignis)“ auf Seite 291
- „authenticate\_parameters (Verbindungsereignis)“ auf Seite 287

### SQL-Beispiel

Ein typisches `authenticate_user_hashed`-Skript ist ein Aufruf einer gespeicherten Prozedur. Die Reihenfolge der Parameter im Aufruf muss mit der oben angegebenen Reihenfolge übereinstimmen. Das folgende Beispiel ruft `ml_add_connection_script` auf, um das Ereignis der gespeicherten Prozedur `my_auth` zuzuweisen.

```
CALL ml_add_connection_script(
  'ver1', 'authenticate_user_hashed',
  'call my_auth (
    {ml s.authentication_status},
    {ml s.username},
    {ml s.hashed_password})'
)
```

Die nachstehende gespeicherte Prozedur in SQL Anywhere benutzt den Benutzernamen und das Kennwort für die Authentifizierung. Die Prozedur stellt nur sicher, dass der angegebene Benutzername eine der Mitarbeiter-IDs in der Tabelle `UEmployee` ist. Die Prozedur geht davon aus, dass die Tabelle `Employee` eine Spalte `binary(20)` namens `hashed_pwd` besitzt.

```
CREATE PROCEDURE my_auth(
  inout @authentication_status integer,
  in @user_name varchar(128),
  in @hpwd binary(32) )
BEGIN
  IF EXISTS
  ( SELECT * FROM uemployee
    WHERE emp_id = @user_name
      and hashed_pwd = @hpwd )
  THEN
    message 'OK' type info to client;
    RETURN 1000;
```

```

ELSE
    message 'Not OK' type info to client;
    RETURN 4000;
END IF
END

```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `authUserHashed` als das Skript für das Ereignis `authenticate_user_hashed`, wenn die Skriptversion `ver1` synchronisiert wird.

```

CALL ml_add_java_connection_script(
    'ver1', 'authenticate_user_hashed',
    'ExamplePackage.ExampleClass.authUserHashed')

```

Im Folgenden sehen Sie die Java-Beispielmethode `authUserHashed`. Sie ruft Java-Methoden auf, die das Benutzerkennwort prüfen und gegebenenfalls ändern.

```

public void authUserHashed(
    ianywhere.ml.script.InOutInteger authStatus,
    String user,
    byte pwd[],
    byte newPwd[] )
    throws java.sql.SQLException {
    // A real authenticate_user_hashed handler
    // would handle more auth code states.
    _curUser = user;
    if( checkPwdHashed( user, pwd ) ) {
        // Authorization successful.
        if( newPwd != null ) {
            // Password is being changed.
            if( changePwdHashed( user, pwd, newPwd ) ) {
                // Authorization OK and password change OK.
                // Use custom code.
                authStatus.setValue( 1001 );
            } else {
                // Auth OK but password change failed.
                // Use custom code
                java.lang.System.err.println( "user: " + user
                    + " pwd change failed!" );
                authStatus.setValue( 1002 );
            }
        } else {
            authStatus.setValue( 1000 );
        }
    } else {
        // Authorization failed.
        authStatus.setValue( 4000 );
    }
}

```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `AuthUserHashed` als das Skript für das Verbindungsereignis `authenticate_user_hashed`, wenn die Skriptversion `ver1` synchronisiert wird. Diese Syntax gilt für konsolidierte SQL Anywhere-Datenbanken.

```

CALL ml_add_dnet_connection_script(
    'ver1',
    'authenticate_user_hashed',

```

```
        'TestScripts.Test.AuthUserHashed'  
    )
```

Im Folgenden sehen Sie die .NET-Beispielmethode AuthUserHashed.

```
namespace TestScripts {  
    public class Test {  
        string _curUser = null;  
        public void AuthUserHashed(  
            ref int authStatus,  
            string user,  
            byte[] pwdHash,  
            byte[] newPwdHash ) {  
            // A real authenticate_user_hashed handler  
            // would handle more auth code states.  
            _curUser = user;  
            if( CheckPwdHashed( user, pwdHash ) ) {  
                // Authorization successful.  
                if( newPwdHash != null ) {  
                    // Password is being changed.  
                    if( ChangePwdHashed( user, pwdHash, newPwdHash ) ) {  
                        // Authorization OK and password change OK.  
                        // Use custom code.  
                        authStatus = 1001;  
                    } else {  
                        // Auth OK but password change failed.  
                        // Use custom code  
                        System.Console.WriteLine( "user: " + user  
                            + " pwd change failed!" );  
                        authStatus = 1002;  
                    }  
                } else {  
                    authStatus = 1000;  
                }  
            } else {  
                // Authorization failed.  
                authStatus = 4000;  
            }  
        }  
    }  
}
```

## begin\_connection (Verbindungseignis)

Wird aufgerufen, während der MobiLink-Server eine Verbindung mit dem Server der konsolidierten Datenbank herstellt.

### Parameter

Keine.

### Standardaktion

Keine.

### Bemerkungen

Die MobiLink-Synchronisation öffnet Verbindungen zur konsolidierten Datenbank bei Bedarf, wenn Anforderungen eingehen. Wenn ein MobiLink-Client eine Verbindung mit dem MobiLink-Server herstellt, weist der MobiLink-Server vorübergehend eine Verbindung mit dem Server der konsolidierten Datenbank zu, über die alle Datenbankaktivitäten für diese Synchronisation ausgeführt werden. Dieses

Ereignis wird unter Umständen nicht aufgerufen, wenn der MobiLink-Server eine Verbindung aus dem Verbindungspool verwendet.

#### Hinweis

Dieses Skript wird nicht allgemein in Java oder .NET verwendet, da anstelle von Datenbankvariablen gewöhnlich member-Variablen in dieser Klasseninstanz verwendet und diese Mitglieder im Konstruktor vorbereitet werden.

#### Siehe auch

- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „end\_connection (Verbindungsereignis)“ auf Seite 340
- „mlsrv16-Option -cn “ auf Seite 56
- „mlsrv16-Option -w “ auf Seite 90

#### SQL-Beispiel

Das folgende SQL-Skript ist für eine konsolidierte SQL Anywhere-Datenbank geeignet. Es werden zwei Variable erstellt, eine für den last\_download timestamp und eine für die Mitarbeiter-ID.

```
CALL ml_add_connection_script(
    'custdb',
    'begin_connection',
    'create variable @LastDownload timestamp;
    create variable @EmployeeID integer;')
```

## begin\_connection\_autocommit (Verbindungsereignis)

Wird zu dem Zeitpunkt aufgerufen, zu dem der MobiLink-Server eine Verbindung mit dem Server der konsolidierten Datenbank herstellt, sodass Sie vorübergehend ein Skript ausführen können, wenn Autocommit aktiviert ist.

#### Parameter

Keine.

#### Standardaktion

Autocommit ist deaktiviert.

#### Bemerkungen

Wenn der MobiLink-Server eine Verbindung mit der konsolidierten Datenbank aufnimmt, schaltet er Autocommit aus, damit er die Uploads und Downloads bei einem Fehler zurücksetzen kann. So wird die Integrität Ihrer Daten gewahrt.

Wenn Sie allerdings eine konsolidierte Adaptive Server Enterprise-Datenbank verwenden, können Sie DDL-Funktionen wie z.B. die Erstellung der temporären Tabellen erst durchführen, wenn Autocommit aktiviert ist. Wenn Sie eine konsolidierte Adaptive Server Enterprise-Datenbank verwenden, führen Sie Ihre DDL-Befehle im Ereignis begin\_connection\_autocommit aus. Wenn das Ereignis abgeschlossen ist, wird Autocommit deaktiviert.

Begin\_connection\_autocommit-Skripten müssen so geschrieben sein, dass sie wiederholbar sind. Dies ist aus folgendem Grund erforderlich: Wenn ein Fehler oder ein Deadlock eintritt, muss der MobiLink-Server in der Lage sein, das Skript nochmals zu versuchen (da er es nicht zurücksetzen kann).

Dieses Ereignis wird nur ausgeführt, wenn ein Skript für das Ereignis definiert wurde.

**Siehe auch**

- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)

# **begin\_download (Verbindungsereignis)**

Verarbeitet alle Anweisungen, unmittelbar bevor der MobiLink-Server mit der Vorbereitung des Downloads beginnt.

**Parameter**

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe [„SQL-Java-Datentypen“ auf Seite 466](#) und [„SQL-.NET-Datentypen“ auf Seite 541](#).

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.last_download                | TIMESTAMP. Die Zeit des ältesten Downloads einer synchronisierten Tabelle.                                                                                                                                                                                                   | 1                              |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                              |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

**Standardaktion**

Keine.

**Bemerkungen**

Der MobiLink-Server führt dieses Ereignis als ersten Schritt bei der Verarbeitung von Download-Daten aus. Download-Daten werden in einer einzigen Transaktion verarbeitet. Die Ausführung dieses Ereignisses ist die erste Aktion in dieser Transaktion.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „end\_download (Verbindungsereignis)“ auf Seite 342
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109

**SQL-Beispiel**

Das folgende Beispiel ruft `ml_add_connection_script` auf, um das Ereignis der gespeicherten Prozedur `SetDownloadParameters` zuzuweisen.

```
CALL ml_add_connection_script (
  'Lab',
  'begin_download',
  'CALL SetDownloadParameters( {ml s.last_table_download}, {ml
s.username} )' )
```

**Java-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `beginDownloadConnection` als das Skript für das Verbindungsereignis `begin_download`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_connection_script(
  'example_ver',
  'begin_download',
  'ExamplePackage.ExampleClass.beginDownloadConnection' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `beginDownloadConnection`. Sie ruft eine Java-Methode (`prepDeleteTables`) auf, die die Löschtabellen über eine JDBC-Verbindung vorbereitet, die zu einem früheren Zeitpunkt festgelegt wurde.

```
public void beginDownloadConnection(
  Timestamp ts,
  String user )
  throws java.sql.SQLException {
  prepDeleteTables ( _syncConn, ts, user );
}
```

**.NET-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `BeginDownload` als das Skript für das Verbindungsereignis `begin_download`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
  'ver1',
  'begin_download',
  'TestScripts.Test.BeginDownload'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode BeginDownload. Sie ruft eine .NET-Methode (prepDeleteTables) auf, die die Löschtabellen über eine Datenbankverbindung vorbereitet, die zu einem früheren Zeitpunkt festgelegt wurde.

```
public void BeginDownload(
    DateTime timestamp,
    string user ) {
    prepDeleteTables ( _syncConn, ts, user );
}
```

## begin\_download (Tabellenereignis)

Verarbeitet Anweisungen im Zusammenhang mit einer bestimmten Tabelle, bevor die Download-Einfügungen, -Aktualisierungen und -Löschungen vorbereitet werden.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe [„SQL-Java-Datentypen“ auf Seite 466](#) und [„SQL-.NET-Datentypen“ auf Seite 541](#).

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge (veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.last_table_download          | TIMESTAMP. Zeitpunkt des letzten Downloads der Tabelle.                                                                                 | 1                              |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                | 2                              |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                          | 3                              |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

### Standardaktion

Keine.

### Bemerkungen

Der MobiLink-Server führt dieses Ereignis als ersten Schritt bei der Verarbeitung von Download-Daten für eine bestimmte Tabelle aus. Die Download-Daten werden in einer eigenen Transaktion vorbereitet. Die Ausführung dieses Ereignisses ist die erste tabellenspezifische Aktion in der Download-Transaktion.

Sie können für jede Tabelle in der entfernten Datenbank ein begin\_download-Skript benutzen. Das Skript wird nur aufgerufen, wenn die Tabelle synchronisiert wird.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „end\_download (Tabellenereignis)“ auf Seite 344
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109

### SQL-Beispiel

Der folgende Aufruf der MobiLink-Systemprozedur ml\_add\_table\_script ruft die Prozedur BeginTableDownload auf. Diese Syntax gilt für konsolidierte SQL Anywhere 16-Datenbanken.

```
CALL ml_add_table_script(
    'version1',
    'Leads',
    'begin_download',
    'CALL BeginTableDownload(
        {ml s.username},
        {ml s.table} )' );
```

Mit den folgenden SQL-Anweisungen erstellen Sie die Prozedur BeginTableDownload. Der Download-Versuch wird in einer Tabelle erfasst.

```
CREATE PROCEDURE BeginTableDownload(
    MLUser varchar(128),
    TableName varchar(128) )
BEGIN
    INSERT INTO DownloadAttempts ( MLUser, TableName, LastDownload );
END
```

## Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `beginDownloadTable` als das Skript für das Tabellenereignis `begin_download`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_table_script(
    'ver1',
    'table1',
    'begin_download',
    'ExamplePackage.ExampleClass.beginDownloadTable'
)
```

Im Folgenden sehen Sie die Java-Beispielmethode `beginDownloadTable`. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog geschrieben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
package ExamplePackage;
public class ExampleClass {
    String _curUser = null;
    public void beginDownloadTable(
        String user,
        String table ) {
        java.lang.System.out.println("Beginning to process download for: " +
        table);
    }
}
```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `BeginTableDownload` als das Skript für das Tabellenereignis `begin_download`, wenn die Skriptversion `ver1` und die Tabelle `table1` synchronisiert werden.

```
CALL ml_add_dnet_table_script(
    'ver1',
    'table1',
    'begin_download',
    'TestScripts.Test.BeginTableDownload'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode `BeginTableDownload`. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog geschrieben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
namespace TestScripts {
public class Test {
    string _curUser = null;
    public void BeginTableDownload(
        string user,
        string table ) {
        System.Console.WriteLine("Beginning to process download for: " + table);
    }
}}
```

## begin\_download\_deletes (Tabellenereignis)

Verarbeitet Anweisungen im Zusammenhang mit einer angegebenen Tabelle, unmittelbar bevor in der entfernten Datenbank eine Liste mit in dieser Tabelle zu löschenden Zeilen abgerufen wird.

## Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.last_table_download          | TIMESTAMP. Zeitpunkt des letzten Downloads der Tabelle.                                                                                                                                                                                                                      | 1                              |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                              |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                                                                                                                                                               | 3                              |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

## Standardaktion

Keine.

## Bemerkungen

Dieses Ereignis wird unmittelbar vor dem Abrufen einer Liste der aus der angegebenen Tabelle in der entfernten Datenbank zu löschenden Zeilen ausgeführt.

**Hinweis**

Für jede Download-Tabelle werden die Ereignisse begin\_download\_deletes, download\_delete\_cursor und end\_download\_deletes nacheinander aufgerufen. Sie sollten in Betracht ziehen, die gesamte Download-Lösch-Logik für eine Tabelle in einem Ereignis download\_delete\_cursor in Form einer einzelnen gespeicherten Prozedur zu implementieren, die eine Ergebnismenge mit allen aus der entfernten Tabelle zu löschenden Zeilen zurückgibt. Die verringerte Anzahl der Skriptaufrufe führt möglicherweise zu einer verbesserten Download-Performance.

Sie können für jede Tabelle in der entfernten Datenbank ein begin\_download\_deletes-Skript benutzen.

Siehe auch

- [„Skriptparameter“ auf Seite 236](#)
- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)
- [„begin\\_download\\_rows \(Tabellenereignis\)“ auf Seite 308](#)
- [„end\\_download\\_rows \(Tabellenereignis\)“ auf Seite 348](#)
- [„Entfernte IDs und MobiLink-Benutzernamen in Skripten“ \[\*MobiLink - Clientadministration\*\]](#)
- [„Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109](#)

begin\_download\_rows (Tabellenereignis)

Verarbeitet Anweisungen im Zusammenhang mit einer angegebenen Tabelle, unmittelbar bevor in der entfernten Datenbank eine Liste mit in dieser Tabelle einzufügenden oder zu aktualisierenden Zeilen abgerufen wird.

Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe [„SQL-Java-Datentypen“ auf Seite 466](#) und [„SQL-.NET-Datentypen“ auf Seite 541](#).

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                            | Reihenfolge (veraltet für SQL) |
|--------------------------------|---------------------------------------------------------|--------------------------------|
| s.last_table_download          | TIMESTAMP. Zeitpunkt des letzten Downloads der Tabelle. | 1                              |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                              |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                                                                                                                                                               | 3                              |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

**Standardaktion**

Keine.

**Bemerkungen**

Dieses Ereignis wird unmittelbar vor dem Abrufen der Zeilen ausgeführt, die in die benannte Tabelle in der entfernten Datenbank eingefügt oder dort aktualisiert werden sollen.

**Hinweis**

Für jede Download-Tabelle werden die Ereignisse begin\_download\_deletes, download\_delete\_cursor und end\_download\_deletes nacheinander aufgerufen. Sie sollten in Betracht ziehen, die gesamte Download-Lösch-Logik für eine Tabelle in einem Ereignis download\_delete\_cursor in Form einer einzelnen gespeicherten Prozedur zu implementieren, die eine Ergebnismenge mit allen aus der entfernten Tabelle zu löschenden Zeilen zurückgibt. Die verringerte Anzahl der Skriptaufrufe führt möglicherweise zu einer verbesserten Download-Performance.

Für jede Tabelle in der entfernten Datenbank können Sie ein begin\_download\_rows-Skript benutzen.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „begin\_download\_deletes (Tabellenereignis)“ auf Seite 306
- „end\_download\_deletes (Tabellenereignis)“ auf Seite 346
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109

## begin\_publication (Verbindungsereignis)

Bietet nützliche Informationen über die synchronisierten Publikationen. Dieses Skript kann auch bei der Verwaltung von Generierungsnummern für dateibasierte Downloads verwendet werden.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                       | Reihenfolge (veraltet für SQL) |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.generation_number            | INTEGER. Dies ist ein INOUT-Parameter. Wenn Ihr Deployment keine dateibasierten Downloads verwendet, kann dieser Parameter ignoriert werden. Standardwert ist "1". | 1                              |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                            | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                           | 2                              |
| s.publication_name             | VARCHAR(128). Der Name der Publikation                                                                                                                             | 3                              |
| s.last_publication_upload      | TIMESTAMP. Zeit des letzten Uploads dieser Publikation.                                                                                                            | 4                              |
| s.last_publication_download    | TIMESTAMP. Zeit des letzten Downloads dieser Publikation.                                                                                                          | 5                              |
| s.subscription_id              | VARCHAR(128). Die entfernte Subskriptions-ID.                                                                                                                      | 6                              |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

### Standardaktion

Die Standard-Generierungsnummer ist 1. Wenn für dieses Ereignis kein Skript definiert ist, ist die an die entfernte Datenbank gesendete Generierungsnummer immer 1.

### Bemerkungen

Mit diesem Ereignis können Sie Synchronisationslogik basierend auf den gerade synchronisierten Publikationen entwerfen. Dieses Ereignis wird in derselben Transaktion wie das Ereignis `begin_synchronization` und nach dem Ereignis `begin_synchronization` aufgerufen. Es wird einmal pro synchronisierter Publikation aufgerufen.

Ein möglicher Einsatzbereich für dieses Ereignis besteht darin, den Umfang des Download auf der Basis der benutzten Publikation festzulegen. Als Beispiel können Sie eine Tabelle verwenden, die zu einer Prioritätspublikation (`PriorityPub`) und zu einer Publikation für alle Tabellen (`AllTablesPub`) gehört. Ein Skript für das Ereignis `begin_publication` könnte die Publikationsnamen in einer Java-Klasse oder einer SQL-Variablen bzw. einem Paket speichern. Download-Skripten könnten dann unterschiedlich vorgehen, je nachdem ob die zu synchronisierende Publikation `PriorityPub` oder `AllTablesPub` ist.

Wenn eine entfernte UltraLite-Datenbank mit `UL_SYNC_ALL` synchronisiert, wird dieses Ereignis einmal mit dem Publikationsnamen "unknown" aufgerufen.

### Generationsnummer

Der Parameter `generation_number` ist speziell für dateibasierte Downloads vorgesehen. Der Ausgabewert für die Generierungsnummer wird vom `begin_publication`-Skript an das `end_publication`-Skript übergeben. Die Bedeutung von `generation_number` hängt davon ab, ob die aktuelle Synchronisation benutzt wird, um eine Download-Datei zu erstellen oder ob die aktuelle Synchronisation ein Upload benötigt.

Bei dateibasierten Downloads werden Generierungsnummern benutzt, um ein Upload vor dem Download zu erzwingen. Während die Generierungsnummern unverändert bleiben, können entfernte Datenbanken viele dateibasierte Downloads ohne Upload verarbeiten. Die Nummer wird in der Download-Datei gespeichert. Während einer Synchronisation, zu der ein Upload gehört, wird eine Generierungsnummer für jede Subskription bei einer Publikation ausgegeben. Die Nummern werden in der Uploadbestätigung an die entfernte Datenbank übergeben und in `SYSSYNC.generation_number` gespeichert.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „end\_publication (Verbindungsereignis)“ auf Seite 350
- „MobiLink - dateibasierte Downloads“ auf Seite 213
- „MobiLink-Generierungsnummern“ auf Seite 220
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109

### SQL-Beispiel

Sie können die Informationen für jede synchronisierte Publikation aufzeichnen. Das folgende Beispiel ruft `ml_add_connection_script` auf, um das Ereignis der gespeicherten Prozedur `RecordPubSync` zuzuweisen.

```
CALL ml_add_connection_script(  
  'version1',  
  'begin_publication',  
  '{CALL RecordPubSync(  
    {ml s.generation_number},  
    {ml s.username},  
    {ml s.publication_name},  
    {ml s.last_publication_upload},  
    {ml s.last_publication_download},  
    {ml s.subscription_id} )}' );
```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `beginPublication` als das Skript für das Verbindungsereignis `begin_publication`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_connection_script(  
  'ver1',  
  'begin_publication',  
  'ExamplePackage.ExampleClass.beginPublication' );
```

Im Folgenden sehen Sie die Java-Beispielmethode `beginPublication`. Sie speichert den Namen jeder Publikation für die spätere Verwendung.

```
package ExamplePackage;  
public class ExampleClass  
{  
    java.util.ArrayList<String> _publicationNames;  
    int _numPublications = 0;  
  
    public void beginPublication( ianywhere.ml.script.InOutInteger  
        generation_number,  
                                String user,  
                                String pub_name )  
    {  
        _numPublications++;  
        _publicationNames.add( pub_name );  
    }  
}
```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens BeginPub als das Skript für das Verbindungsereignis begin\_publication, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
    'ver1',
    'begin_publication',
    'TestScripts.Test.BeginPub'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode BeginPub. Sie speichert den Namen jeder Publikation für die spätere Verwendung.

```
using System.Collections.Generic;
namespace TestScripts
{
    class Test
    {
        List<string> _publicationNames = new List<string>();
        int _numPublications = 0;

        public void BeginPub( ref int generation_number,
                               string user,
                               string pub_name )
        {
            _numPublications++;
            _publicationNames.Add( pub_name );
        }
    }
}
```

## begin\_synchronization (Verbindungsereignis)

Verarbeitet Anweisungen in Vorbereitung auf den Synchronisationsprozess.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 1                                 |
| s.new_remote_id                | VARCHAR(128). Die entfernte MobiLink-ID, wenn die entfernte ID in der konsolidierten Datenbank neu ist. Wenn die entfernte ID nicht neu ist, ist der Wert NULL.                                                                                                              |                                   |
| s.new_username                 | VARCHAR(128). Der MobiLink-Benutzername, wenn der Benutzername in der konsolidierten Datenbank neu ist. Wenn der Benutzername nicht neu ist, ist der Wert NULL.                                                                                                              |                                   |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

**Standardaktion**

Keine.

**Bemerkungen**

Der MobiLink-Server führt dieses Ereignis aus, nachdem er vom MobiLink-Client alle für den Beginn der Synchronisation erforderlichen Daten empfangen hat.

Das Skript begin\_synchronization ist bei der Pflege von Statistiken hilfreich. Das Skript end\_synchronization wird nämlich auch dann aufgerufen, wenn kein Fehler oder Konflikt vorliegt. Während also die Uploadtransaktion zurückgesetzt wird, werden Statistikdaten beibehalten und gepflegt.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „end\_synchronization (Verbindungsereignis)“ auf Seite 353
- „begin\_synchronization (Tabelleneignis)“ auf Seite 316
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

**SQL-Beispiel**

Es ist sinnvoll, den username-Wert in einer temporären Tabelle oder Variablen zu speichern, falls Sie diesen Wert in nachfolgenden Skripten referenzieren möchten.

```
CALL ml_add_connection_script (
  'version1',
  'begin_synchronization',
  'set @EmployeeID = {ml s.username}' );
```

**Java-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens beginSynchronizationConnection als das Skript für das Verbindungsereignis begin\_synchronization, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_connection_script(
  'ver1',
  'begin_synchronization',
  'ExamplePackage.ExampleClass.beginSynchronizationConnection'
)
```

Im Folgenden sehen Sie die Java-Beispielmethode beginSynchronizationConnection. Sie speichert den Namen des Synchronisationsbenutzers für die spätere Verwendung.

```
package ExamplePackage;
public class ExampleClass {
    String _curUser = null;
    public void beginSynchronizationConnection(
        String user ) {
        _curUser = user;
    }
}
```

**.NET-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens BeginSync als das Skript für das Verbindungsereignis begin\_synchronization, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_dnet_connection_script( 'ver1',
  'begin_synchronization',
  'TestScripts.Test.BeginSync'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode BeginSync. Sie speichert den Namen des Synchronisationsbenutzers für die spätere Verwendung.

```
namespace TestScripts {
public class Test {
    string _curUser = null;
}
```

```
public void BeginSync(  
    string user ) {  
    _curUser = user;  
    }  
}
```

## begin\_synchronization (Tabellenereignis)

Verarbeitet zu Beginn der Synchronisation Anweisungen im Zusammenhang mit einer bestimmten Tabelle.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                    | Reihenfolge (veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                         | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                        | 1                              |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                                                  | 2                              |
| s.new_remote_id                | VARCHAR(128). Die entfernte MobiLink-ID, wenn die entfernte ID in der konsolidierten Datenbank neu ist. Wenn die entfernte ID nicht neu ist, ist der Wert NULL. |                                |
| s.new_username                 | VARCHAR(128). Der MobiLink-Benutzername, wenn der Benutzername in der konsolidierten Datenbank neu ist. Wenn der Benutzername nicht neu ist, ist der Wert NULL. |                                |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

### Standardaktion

Keine.

### Bemerkungen

Der MobiLink-Server führt dieses Ereignis aus, nachdem er vom MobiLink-Client alle für den Beginn der Synchronisation erforderlichen Daten empfangen hat.

Sie können für jede Tabelle in der entfernten Datenbank ein begin\_synchronization-Skript benutzen. Das Ereignis wird nur aufgerufen, wenn die Tabelle synchronisiert wird.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „end\_synchronization (Tabellenereignis)“ auf Seite 355
- „begin\_synchronization (Verbindungsereignis)“ auf Seite 313
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

### SQL-Beispiel

Das Tabellenereignis begin\_synchronization wird benutzt, um die Synchronisation einer bestimmten Tabelle einzurichten. Das folgende SQL-Skript registriert ein Skript, das eine temporäre Tabelle zum Speichern von Zeilen während der Synchronisation erstellt. Diese Syntax gilt für eine konsolidierte SQL Anywhere-Datenbank.

```
CALL ml_add_table_script(
  'ver1',
  'sales_order',
  'begin_synchronization',
  'CREATE TABLE #sales_order (
    id          integer NOT NULL default autoincrement,
    cust_id     integer NOT NULL,
    order_date  date NOT NULL,
    fin_code_id char(2) NULL,
    region      char(7) NULL,
    sales_rep   integer NOT NULL,
    PRIMARY KEY (id),
  )' )
```

## Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `beginSynchronizationTable` als das Skript für das Tabellenereignis `begin_synchronization`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_table_script(
  'ver1',
  'table1',
  'begin_synchronization',
  'ExamplePackage.ExampleClass.beginSynchronizationTable' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `beginSynchronizationTable`. Sie fügt den aktuellen Tabellennamen zu einer Liste der Tabellennamen in dieser Instanz hinzu.

```
package ExamplePackage;
import java.util.ArrayList;
import java.sql.Timestamp;
class ExampleClass
{
    ArrayList<String> _tableList;
    String _curTable;
    public void beginSynchronizationTable( String user,
   String table )
    {
        _curTable = table;
        _tableList.add( table );
    }

    public void endTableDownload( Timestamp ts,
                                  String user,
                                  String table )
    {
        _curTable = null;
    }
}
```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `BeginTableSync` als das Skript für das Tabellenereignis `begin_synchronisation`, wenn die Skriptversion `ver1` und die Tabelle `table1` synchronisiert werden.

```
CALL ml_add_dnet_table_script (
  'ver1',
  'table1',
  'begin_synchronization',
  'TestScripts.Test.BeginTableSync' )
```

Im Folgenden sehen Sie die .NET-Beispielmethode `BeginTableSync`. Sie fügt den aktuellen Tabellennamen zu einer Liste der Tabellennamen in dieser Instanz hinzu.

```
using System.Collections.Generic;
using System;
namespace TestScripts
{
    class Test
    {
        List<string> _tableList = new List<string>();
        string _curTable = "";
    }
}
```

```

public void BeginSynchronizationTable( string    user,
                                     string    table )
{
    _curTable = table;
    _tableList.Add( table );
}
public void EndTableDownload( DateTime    timestamp,
                             string    user,
                             string    table )
{
    _curTable = null;
}
}

```

## begin\_upload (Verbindungseignis)

Führt Anweisungen aus, unmittelbar bevor der MobiLink-Server beginnt, den Upload-Datenstrom von Einfügungen, Aktualisierungen und Löschungen zu verarbeiten.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge (veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                | 1                              |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

**Standardaktion**

Keine.

**Bemerkungen**

Der MobiLink-Server führt dieses Ereignis als ersten Schritt bei der Verarbeitung von hochgeladenen Zeilen aus. Hochgeladene Zeilen werden in einer einzigen Transaktion verarbeitet. Die Ausführung dieses Ereignisses ist die erste Aktion in dieser Transaktion.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „end\_upload (Verbindungsereignis)“ auf Seite 358
- „begin\_upload (Tabellenereignis)“ auf Seite 321
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

**SQL-Beispiel**

Das Verbindungsereignis begin\_upload wird benutzt, um alle Schritte auszuführen, die vor dem Upload von Zeilen erforderlich sind. Das nachstehende SQL-Skript erstellt eine temporäre Tabelle zur Speicherung der alten und neuen Zeilenwerte zur Konfliktverarbeitung der Tabelle sales\_order. Dieses Beispiel eignet sich für eine konsolidierte SQL Anywhere-Datenbank.

```
CALL ml_add_connection_script(  
  'version1',  
  'begin_upload',  
  'CREATE TABLE #sales_order_conflicts (  
    id          integer NOT NULL default autoincrement,  
    cust_id     integer NOT NULL,  
    order_date  date NOT NULL,  
    fin_code_id char(2) NULL,  
    region      char(7) NULL,  
    sales_rep   integer NOT NULL,  
    new_value   char(1) NOT NULL,  
    PRIMARY KEY (id) )' )
```

## Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `beginUploadConnection` als das Skript für das Verbindungsereignis `begin_upload`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_connection_script(
  'ver1',
  'begin_upload',
  'ExamplePackage.ExampleClass.beginUploadConnection ' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `beginUploadConnection`. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog geschrieben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
package ExamplePackage;
public class ExampleClass {
    String _curUser = null;
    public void beginUploadConnection( String user ) {
        java.lang.System.out.println(
            "Starting upload for user: " + user );
    }
}
```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `BeginUpload` als das Skript für das Verbindungsereignis `begin_upload`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
  'ver1',
  'begin_upload',
  'TestScripts.Test.BeginUpload'
)
```

Das folgende C#-Beispiel speichert den aktuellen Benutzernamen für die Verwendung in einem späteren Ereignis.

```
namespace TestScripts {
public class Test {
    string _curUser = null;
    public void BeginUpload( string curUser ) {
        _curUser = curUser;
    }
}}
```

## begin\_upload (Tabellenereignis)

Führt Anweisungen im Zusammenhang mit einer bestimmten Tabelle aus, unmittelbar bevor der MobiLink-Server beginnt, den Upload-Datenstrom von Einfügungen, Aktualisierungen und Löschungen zu verarbeiten.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 1                                 |
| s.table                        | VARCHAR(128). Der Tabellenname.                                                                                                                                                                                                                                              | 2                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

### Standardaktion

Keine.

### Bemerkungen

Der MobiLink-Server führt dieses Ereignis als ersten Schritt bei der Verarbeitung von hochgeladenen Zeilen aus. Hochgeladene Zeilen werden in einer separaten Transaktion verarbeitet. Die Ausführung dieses Ereignisses ist die erste tabellenspezifische Aktion in dieser Transaktion.

Sie können für jede Tabelle in der entfernten Datenbank ein begin\_upload-Skript benutzen. Das Skript wird nur dann aufgerufen, wenn die Tabelle tatsächlich synchronisiert wird.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „end\_upload (Tabellenereignis)“ auf Seite 361
- „begin\_upload (Verbindungsereignis)“ auf Seite 319
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

## SQL-Beispiel

Wenn Sie Zeilen von einer entfernten Datenbank übertragen, sollten Sie die Änderungen in eine Zwischentabelle einfügen und die Änderungen manuell vornehmen. In diesem Ereignis können Sie Zeilen aus einer globalen temporären Tabelle löschen, um den Empfang der neuen Zeilen vorzubereiten.

```
CALL ml_add_table_script(
  'version1',
  'Leads',
  'begin_upload',
  'DELETE FROM T_Leads' )
```

## Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens beginUploadTable als das Skript für das Tabellenereignis begin\_upload, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_table_script(
  'ver1',
  'table1',
  'begin_upload',
  'ExamplePackage.ExampleClass.beginUploadTable'
)
```

Im Folgenden sehen Sie die Java-Beispielmethode beginUploadTable. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog geschrieben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
package ExamplePackage;
public class ExampleClass {
    String _curUser = null;
    public void beginUploadTable(
        String user,
        String table ) {
        java.lang.System.out.println("Beginning to process upload for: " + table);
    }
}
```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens BeginTableUpload als das Skript für das Tabellenereignis begin\_upload, wenn die Skriptversion ver1 und die Tabelle table1 synchronisiert werden.

```
CALL ml_add_dnet_table_script(
  'ver1',
  'table1',
  'begin_upload',
  'TestScripts.Test.BeginTableUpload'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode BeginTableUpload. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog geschrieben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
namespace TestScripts {
    public class Test {
        string _curUser = null;
        public void BeginTableUpload(
```

```
string user,
string table ) {
System.Console.WriteLine("Beginning to process upload for: " + table);
}}}
```

begin\_upload\_deletes (Tabellenereignis)

Verarbeitet Anweisungen im Zusammenhang mit einer angegebenen Tabelle, unmittelbar bevor in der entfernten Datenbank gelöschte Zeilen aus dieser Tabelle übertragen werden.

Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „SQL-Java-Datentypen“ auf Seite 466 und „SQL-.NET-Datentypen“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 1                              |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                                                                                                                                                               | 2                              |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

Standardaktion

Keine.

## Bemerkungen

Dieses Ereignis wird unmittelbar vor der Übernahme der Änderungen ausgeführt, die sich aus den gelöschten Zeilen aus der benannten entfernten Tabelle ergeben.

Sie können für jede Tabelle in der entfernten Datenbank ein `begin_upload_deletes`-Skript benutzen. Das Skript wird nur dann aufgerufen, wenn die Tabelle tatsächlich synchronisiert wird.

## Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „end\_upload\_deletes (Tabellenereignis)“ auf Seite 363
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

## SQL-Beispiel

Das Tabellenereignis `begin_upload_deletes` wird benutzt, um alle erforderlichen Schritte nach dem Upload von Einfügungen und Aktualisierungen für eine bestimmte Tabelle auszuführen, bevor Löschungen für diese Tabelle hochgeladen werden. Das nachstehende SQL-Skript erstellt eine temporäre Datei für das vorübergehende Speichern von Löschungen während des Uploads: Diese Syntax gilt für eine konsolidierte SQL Anywhere-Datenbank.

```
CALL ml_add_table_script(
  'ver1',
  'sales_order',
  'begin_upload_deletes',
  'CREATE TABLE #sales_order_deletes (
    id          integer NOT NULL default autoincrement,
    cust_id     integer NOT NULL,
    order_date  date NOT NULL,
    fin_code_id char(2) NULL,
    region      char(7) NULL,
    sales_rep   integer NOT NULL,
    PRIMARY KEY (id) )' )
```

## Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `beginUploadDeletes` als das Skript für das Tabellenereignis `begin_upload_deletes`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_table_script(
  'ver1',
  'table1',
  'begin_upload_deletes',
  'ExamplePackage.ExampleClass.beginUploadDeletes' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `beginUploadDeletes`. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog geschrieben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
package ExamplePackage;
public class ExampleClass {
    String _curUser = null;
    public void beginUploadDeletes(
        String user,
        String table )
```

```
throws java.sql.SQLException {
java.lang.System.out.println(
"Starting upload deletes for table: " + table );
}}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens BeginUploadDeletes als das Skript für das Tabellenereignis begin\_upload\_deletes, wenn die Skriptversion ver1 und die Tabelle table1 synchronisiert werden.

```
CALL ml_add_dnet_table_script(
'ver1',
'table1',
'begin_upload_deletes',
'TestScripts.Test.BeginUploadDeletes'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode BeginUploadDeletes. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog geschrieben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
namespace TestScripts {
public class Test {
    string _curUser = null;
    public void BeginUploadDeletes(
        string user,
        string table ) {
        System.Console.WriteLine(
            "Starting upload deletes for table: " + table );
    }
}
```

## begin\_upload\_rows (Tabellenereignis)

Verarbeitet Anweisungen im Zusammenhang mit einer angegebenen Tabelle, unmittelbar bevor in der entfernten Datenbank Einfügungen und Aktualisierungen aus dieser Tabelle übertragen werden.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 1                                 |
| s.table                        | VARCHAR(128). Der Tabellenname.                                                                                                                                                                                                                                              | 2                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

**Standardaktion**

Keine.

**Bemerkungen**

Dieses Ereignis wird unmittelbar vor der Übernahme der Änderungen ausgeführt, die sich aus Einfügungen und Löschungen in der im zweiten Parameter genannten entfernten Tabelle ergeben.

Sie können für jede Tabelle in der entfernten Datenbank ein begin\_upload\_rows-Skript benutzen. Das Skript wird nur dann aufgerufen, wenn die Tabelle tatsächlich synchronisiert wird.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „end\_upload\_rows (Tabellenereignis)“ auf Seite 366
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [[MobiLink - Clientadministration](#)]

**SQL-Beispiel**

Das Tabellenereignis begin\_upload\_rows wird benutzt, um alle erforderlichen Schritte vor dem Upload von Einfügungen und Aktualisierungen für eine bestimmte Tabelle auszuführen. Das nachstehende Skript ruft eine gespeicherte Prozedur auf, die die konsolidierte Datenbank für Einfügungen und Aktualisierungen in die Tabelle Inventory vorbereitet:

```
CALL ml_add_table_script(
  'MyCorp 1.0',
  'Inventory',
```

```
'begin_upload_rows',  
'CALL PrepareForUpserts()' )
```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `beginUploadRows` als das Skript für das Tabellenereignis `begin_upload_rows`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_table_script(  
  'ver1',  
  'table1',  
  'begin_upload_rows',  
  'ExamplePackage.ExampleClass.beginUploadRows' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `beginUploadRows`. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog geschrieben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
package ExamplePackage;  
public class ExampleClass {  
  String _curUser = null;  
  public void beginUploadRows(  
    String user,  
    String table )  
    throws java.sql.SQLException {  
    java.lang.System.out.println(  
      "Starting upload rows for table: " +  
      table + " and user: " + user );  
  }  
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `BeginUploadRows` als das Skript für das Tabellenereignis `begin_upload_rows`, wenn die Skriptversion `ver1` und die Tabelle `table1` synchronisiert werden.

```
CALL ml_add_dnet_table_script(  
  'ver1',  
  'table1',  
  'begin_upload_rows',  
  'TestScripts.Test.BeginUploadRows'  
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode `BeginUploadRows`. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog geschrieben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
namespace TestScripts {  
  public class Test  
  {  
    string _curUser = null;  
    public void BeginUploadRows(  
      string user,  
      string table ) {  
      System.Console.WriteLine(  
        "Starting upload rows for table: " +  
        table + " and user: " + user );  
    }  
  }  
}
```

## download\_cursor (Tabellenereignis)

Ein Datenskript, das einen Cursor für die Auswahl der Zeilen definiert, die heruntergeladen und in der angegebenen Tabelle in der entfernten Datenbank eingefügt oder aktualisiert werden.

### Parameter

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.last_table_download          | TIMESTAMP. Zeitpunkt des letzten Downloads der Tabelle.                                                                                                                                                                                                                      | 1                                 |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

### Standardaktion

Keine.

### Bemerkungen

Der MobiLink-Server öffnet mithilfe des Skripts einen schreibgeschützten Cursor, mit dem eine Liste von Zeilen abgerufen wird, die in die entfernte Datenbank heruntergeladen werden sollen.

Sie können für jede Tabelle in der entfernten Datenbank ein download\_cursor-Skript benutzen.

Um die Performance bei der Downloadphase der Synchronisation auf UltraLite-Clients zu optimieren, wenn der Bereich der Primärschlüsselwerte außerhalb der aktuellen Zeilen auf dem Device liegt, sollten

Sie die Zeilen im Download-Cursor nach dem Primärschlüssel sortieren. Von dieser Optimierung können z.B. Download-Vorgänge von großen Referenztabellen profitieren.

Jedes download\_cursor-Skript muss eine SELECT-Anweisung oder einen Aufruf einer Prozedur enthalten, die eine Ergebnismenge zurückgibt. Der MobiLink-Server verwendet diese Anweisung, um einen Cursor in der konsolidierten Datenbank zu definieren.

Das Skript muss alle Spalten auswählen, die mit den Spalten in der entsprechenden Tabelle der entfernten Datenbank übereinstimmen. Die Spalten in der konsolidierten Datenbank können andere Namen haben als die entsprechenden Spalten in der entfernten Datenbank. Ihre Datentypen müssen jedoch kompatibel sein.

Die Spalten müssen in der Reihenfolge ausgewählt werden, in der die entsprechenden Spalten in der entfernten Datenbank definiert sind.

Um den Download unnötiger Zeilen zu vermeiden, sollten Sie zeitstempelbasierte Downloads in Betracht ziehen. Wenn Sie zeitstempelbasierte Downloads verwenden, fügen Sie eine Zeile ähnlich der folgenden in die WHERE-Klausel Ihres download\_cursor-Skripts ein:

```
AND last_modified >= {ml s.last_table_download}
```

Dieses Skript muss in SQL implementiert werden. Hinweise zur Java- oder .NET-Verarbeitung von Zeilen finden Sie unter [„Direkte Zeilenbehandlung“ auf Seite 627](#).

Wenn Sie READPAST-Tabellen-Hints in download\_cursor-Skripten verwenden möchten, da Sie viele Aktualisierungen ausführen, die die Download-Leistung beeinträchtigen, sollten Sie für Downloads stattdessen die Snapshot-Isolation verwenden. Der READPAST-Tabellen-Hint kann Probleme verursachen, wenn er in download\_cursor-Skripten verwendet wird. Bei der Verwendung von zeitstempelbasierten Downloads kann der READPAST-Hint dazu führen, dass Zeilen fehlen und dass kein Download einer Zeile in eine entfernte Datenbank möglich ist. Beispiel:

- Eine Zeile wird der konsolidierten Datenbank hinzugefügt und festgeschrieben. Die Zeile enthält eine last\_modified-Spalte mit einer Zeitangabe des vorhergehenden Tags.
- Dieselbe Zeile wird zwar aktualisiert, aber nicht festgeschrieben.
- Eine entfernte Datenbank mit einer last\_download-Uhrzeit der letzten Woche wird synchronisiert.
- Ein download\_cursor-Skript versucht, die Zeile unter Verwendung von READPAST auszuwählen, und überspringt die Zeile.
- Die Transaktion, die die Zeile aktualisiert hat, wird zurückgesetzt. Die nächste Zeit des letzten Downloads für die entfernte Datenbank wird auf heute vorverlegt.

Ab diesem Punkt wird die Zeile erst dann heruntergeladen, wenn sie aktualisiert wurde. Als mögliche Behelfslösung können Sie ein generate\_next\_last\_download\_timestamp- oder modify\_next\_last\_download\_timestamp-Skript implementieren und die Zeit des letzten Downloads auf die Anfangszeit der ältesten offenen Transaktion setzen.

**Siehe auch**

- „Datenskripten“ auf Seite 283
- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „Skripten für den Download von Zeilen“ auf Seite 265
- „download\_cursor-Skripten“ auf Seite 266
- „Partitionierte Zeilen in entfernten Datenbanken“ auf Seite 113
- „download\_delete\_cursor (Tabellenereignis)“ auf Seite 331
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109
- "READPAST mit MobiLink-Synchronisation verwenden" in „FROM-Klausel“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]

**SQL-Beispiel**

Das folgende Beispiel stammt aus einer Oracle-Installation. Die Anweisung kann jedoch mit allen unterstützten Datenbanken eingesetzt werden. Im Beispiel werden alle Zeilen heruntergeladen, die geändert wurden, seit der Benutzer den letzten Datendownload vorgenommen hat, und die mit dem Benutzernamen in der Spalte emp\_name übereinstimmen.

```
CALL ml_add_table_script(
  'Lab',
  'ULOrder',
  'download_cursor',
  'SELECT order_id,
    cust_id,
    prod_id,
    emp_id,
    disc,
    quant,
    notes,
    status
  FROM ULOrder
  WHERE last_modified >= {ml s.last_table_download}
    AND emp_name = {ml s.username}' )
```

**download\_delete\_cursor (Tabellenereignis)**

Ein Datenskript, das einen Cursor für die Auswahl der Zeilen definiert, die in der entfernten Datenbank gelöscht werden sollen.

**Parameter**

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.last_table_download          | TIMESTAMP. Zeitpunkt des letzten Downloads der Tabelle.                                                                                                                                                                                                                      | 1                                 |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

**Standardaktion**

Keine.

**Bemerkungen**

Der MobiLink-Server öffnet einen schreibgeschützten Cursor, mit dem eine Liste von Zeilen abgerufen wird, die heruntergeladen und aus der entfernten Datenbank gelöscht werden sollen. Dieses Skript muss eine SELECT-Anweisung enthalten, die die Primärschlüsselwerte der Zeilen enthält, die aus der Tabelle in der entfernten Datenbank gelöscht werden sollen.

Sie können für jede Tabelle in der entfernten Datenbank ein download\_delete\_cursor-Skript benutzen.

Wenn das download\_delete\_cursor-Ereignis für die Primärschlüsselspalten für eine oder mehrere Zeilen in einer Tabelle NULL enthält, weist der MobiLink-Server die entfernte Datenbank an, alle Zeilen in der Tabelle zu löschen. Siehe [Alle Zeilen in einer Tabelle löschen auf Seite 269](#).

Aus der konsolidierten Datenbank gelöschte Zeilen können nicht in einer Ergebnismenge erscheinen, die von einem download\_delete\_cursor-Ereignis definiert wurde, und werden daher nicht automatisch aus der entfernten Datenbank gelöscht. Eine Technik zum Kennzeichnen von Zeilen, die aus entfernten Datenbanken gelöscht werden sollen, besteht darin, der Tabelle in der konsolidierten Datenbank eine Spalte hinzuzufügen, die eine Zeile als inaktiv kennzeichnet.

Um den Download unnötiger zu löschender Zeilen zu vermeiden, sollten Sie zeitstempelbasierte Downloads in Betracht ziehen.. Fügen Sie eine Zeile ähnlich der folgenden in die WHERE-Klausel Ihres download\_delete\_cursor-Skripts ein:

```
AND last_modified >= {ml s.last_table_download}
```

Dieses Skript muss in SQL implementiert werden. Hinweise zur Java- oder .NET-Verarbeitung von Zeilen finden Sie unter [„Direkte Zeilenbehandlung“ auf Seite 627](#).

Die Verwendung von READPAST-Tabellen-Hints in download\_delete cursor-Skripten kann zu Problemen führen. Ausführliche Informationen hierzu finden Sie unter dem Ereignis download\_cursor.

### Siehe auch

- [„Datenskripten“ auf Seite 283](#)
- [„Skriptparameter“ auf Seite 236](#)
- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)
- [„download\\_cursor \(Tabellenereignis\)“ auf Seite 329](#)
- [„Skripten für den Download von Zeilen“ auf Seite 265](#)
- [„Partitionierte Zeilen in entfernten Datenbanken“ auf Seite 113](#)
- [„download\\_delete\\_cursor-Skripten“ auf Seite 267](#)
- [„Entfernte IDs und MobiLink-Benutzernamen in Skripten“ \[\*MobiLink - Clientadministration\*\]](#)
- [„Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109](#)
- ["READPAST mit MobiLink-Synchronisation verwenden" in „FROM-Klausel“ \[\*SQL Anywhere Server - SQL-Referenzhandbuch\*\]](#)

### SQL-Beispiel

Dieses Beispiel stammt aus dem Beispiel Contact und befindet sich in *Samples\MobiLink\Contact\build\_consol.sql*. Es löscht jeden Kunden aus der entfernten Datenbank, der geändert wurde, nachdem dieser Benutzer den letzten Download durchgeführt hat (`Customer.last_modified >= {ml s.last_table_download}`) und für den eine der folgenden beiden Aussagen zutrifft.

- Er gehört nicht zu dem synchronisierenden Benutzer (`SalesRep.username != {ml s.username}`).
- Er ist in der konsolidierten Datenbank als inaktiv markiert (`Customer.active = 0`).

```
CALL ml_add_table_script(
  'ver1',
  'table1',
  'download_delete_cursor',
  'SELECT cust_id FROM Customer key join SalesRep
   WHERE Customer.last_modified >= {ml s.last_table_download} AND
     ( SalesRep.username != {ml s.username} OR Customer.active = 0 )')
```

## download\_statistics (Verbindungsereignis)

Ermöglicht den Zugriff auf Synchronisationsstatistiken für Download-Vorgänge.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe [„SQL-Java-Datentypen“ auf Seite 466](#) und [„SQL-.NET-Datentypen“ auf Seite 541](#).

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Namen und Fragezeichen dürfen jedoch in einem Skript nicht gemischt verwendet werden. In SQL-

Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername wie er in der Definition SYNCHRONIZATION USER angegeben wurde.                                                                                                                                                                       | 1                              |
| s.warnings                     | INTEGER. Die Anzahl der ausgegebenen Warnungen.                                                                                                                                                                                                                              | 2                              |
| s.errors                       | INTEGER. Die Anzahl der eingetretenen Fehler, einschließlich der bearbeiteten Fehler.                                                                                                                                                                                        | 3                              |
| s.fetched_rows                 | INTEGER. Die Anzahl der Zeilen, die vom Skript download_cursor abgerufen wurden.                                                                                                                                                                                             | 4                              |
| s.deleted_rows                 | INTEGER. Die Anzahl der Zeilen, die vom Skript download_delete_cursor abgerufen wurden.                                                                                                                                                                                      | 5                              |
| s.filtered_rows                | INTEGER. Die Anzahl der Zeilen aus dem Parameter fetched_rows, die tatsächlich an die entfernte Datenbank gesendet wurden. Daraus ist die Download-Filterung der Upload-Werte ersichtlich.                                                                                   | 6                              |
| s.bytes                        | INTEGER. Die Anzahl der per Download an die entfernte Datenbank übertragenen Bytes.                                                                                                                                                                                          | 7                              |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

**Standardaktion**

Keine.

**Bemerkungen**

Mit dem Ereignis `download_statistics` können Sie für jeden beliebigen Benutzer statistische Daten zu Download-Vorgängen sammeln. Das Verbindungsskript `download_statistics` wird unmittelbar vor dem Festschreiben am Ende der Download-Transaktion aufgerufen.

**Hinweis**

Abhängig von der Befehlszeile werden nicht alle Warnungen protokolliert. Die Anzahl der an dieses Skript übergebenen Warnungen ist die Anzahl der Warnungen, die protokolliert würden, wenn keine Warnungen deaktiviert wären. Diese Zahl kann höher sein als die Anzahl der protokollierten Warnungen.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „download\_statistics (Tabellenereignis)“ auf Seite 337
- „upload\_statistics (Verbindungsereignis)“ auf Seite 448
- „upload\_statistics (Tabellenereignis)“ auf Seite 452
- „synchronization\_statistics (Verbindungsereignis)“ auf Seite 421
- „synchronization\_statistics (Tabellenereignis)“ auf Seite 425
- „time\_statistics (Verbindungsereignis)“ auf Seite 428
- „time\_statistics (Tabellenereignis)“ auf Seite 432
- „MobiLink-Profiler“ auf Seite 192
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

**SQL-Beispiel**

Das folgende Beispiel fügt Synchronisationsstatistiken in die Tabelle `download_audit` ein.

```
CALL ml_add_connection_script(
  'ver1',
  'download_statistics',
  'INSERT INTO download_audit(
    user_name,
    warnings,
    errors,
    fetched_rows,
    deleted_rows,
    filtered_rows,
    bytes )
VALUES (
  {ml s.username},
  {ml s.warnings},
  {ml s.errors},
  {ml s.fetched_rows},
  {ml s.deleted_rows},
  {ml s.filtered_rows},
  {ml s.bytes})')

```

Nachdem wesentliche statistische Daten in die Audit-Tabelle eingefügt wurden, können Sie sie zur Überwachung der Synchronisationen einsetzen und ggf. Optimierungen durchführen.

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `downloadStatisticsConnection` als das Skript für das Ereignis `download_statistics`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_connection_script(  
  'ver1',  
  'download_statistics',  
  'ExamplePackage.ExampleClass.downloadStatisticsConnection' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `downloadStatisticsConnection`. Sie gibt die Anzahl der Zeilen aus, die in das MobiLink-Nachrichtenlog geschrieben wurden. (Das Ausgeben der Anzahl der abgerufenen Zeilen in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
package ExamplePackage;  
public class ExampleClass {  
    String _curUser = null;  
    public void downloadStatisticsConnection(  
        String user,  
        int warnings,  
        int errors,  
        int fetchedRows,  
        int deletedRows,  
        int filteredRows,  
        int bytes ) {  
        java.lang.System.out.println(  
            "download connection stats fetchedRows: "  
            + fetchedRows );  
    }  
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `DownloadStats` als das Skript für das Verbindungsereignis `download_statistics`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_dnet_connection_script(  
  'ver1',  
  'download_statistics',  
  'TestScripts.Test.DownloadStats'  
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode `DownloadStats`. Sie gibt die Anzahl der Zeilen aus, die in das MobiLink-Nachrichtenlog geschrieben wurden. (Das Ausgeben der Anzahl der abgerufenen Zeilen in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
namespace TestScripts {  
    public class Test {  
        string _curUser = null;  
        public void DownloadStats(  
            string user,  
            int warnings,  
            int errors,  
            int deletedRows,  
            int fetchedRows,  
            int downloadRows,
```

```

int filteredRows,
int bytes ) {
System.Console.WriteLine(
    "download connection stats fetchedRows: "
    + fetchedRows );
}}

```

## download\_statistics (Tabellenereignis)

Ermöglicht den Zugriff auf Synchronisationsstatistiken für Download-Vorgänge nach Tabelle.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername wie er in der Definition SYNCHRONIZATION USER angegeben wurde.                                  | 1                                 |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                          | 2                                 |
| s.warnings                     | INTEGER. Die Anzahl der ausgegebenen Warnungen.                                                                                         | 3                                 |
| s.errors                       | INTEGER. Die Anzahl der eingetretenen Fehler, einschließlich der bearbeiteten Fehler.                                                   | 4                                 |
| s.fetched_rows                 | INTEGER. Die Anzahl der Zeilen, die vom Skript download_cursor abgerufen wurden.                                                        | 5                                 |
| s.deleted_rows                 | INTEGER. Die Anzahl der Zeilen, die vom Skript download_delete_cursor abgerufen wurden.                                                 | 6                                 |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.filtered_rows                | INTEGER. Die Anzahl der aus fetched_rows gefilterten Zeilen. Daraus ist die Download-Filterung der Upload-Werte ersichtlich.                                                                                                                                                 | 7                                 |
| s.bytes                        | INTEGER. Die Anzahl der per Download an die entfernte Datenbank übertragenen Bytes.                                                                                                                                                                                          | 8                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

### Standardaktion

Keine.

### Bemerkungen

Mit dem Ereignis download\_statistics können Sie für jeden beliebigen Benutzer und jede beliebige Tabelle statistische Daten zu Downloads im Zusammenhang mit der betreffenden Tabelle sammeln. Das Tabellenskript download\_statistics wird unmittelbar vor dem Festschreiben am Ende der Download-Transaktion aufgerufen.

#### Hinweis

Abhängig von der Befehlszeile werden nicht alle Warnungen protokolliert. Die Anzahl der an dieses Skript übergebenen Warnungen ist die Anzahl der Warnungen, die protokolliert würden, wenn keine Warnungen deaktiviert wären. Diese Zahl kann höher sein als die Anzahl der protokollierten Warnungen.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „download\_statistics (Verbindungsereignis)“ auf Seite 333
- „upload\_statistics (Verbindungsereignis)“ auf Seite 448
- „upload\_statistics (Tabellenereignis)“ auf Seite 452
- „synchronization\_statistics (Verbindungsereignis)“ auf Seite 421
- „synchronization\_statistics (Tabellenereignis)“ auf Seite 425
- „time\_statistics (Verbindungsereignis)“ auf Seite 428
- „time\_statistics (Tabellenereignis)“ auf Seite 432
- „MobiLink-Profiler“ auf Seite 192
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

## SQL-Beispiel

Das folgende Beispiel fügt Synchronisationsstatistiken in die Tabelle download\_audit ein. Nachdem wesentliche statistische Daten in die Audit-Tabelle eingefügt wurden, können Sie sie zur Überwachung der Synchronisationen einsetzen und ggf. Optimierungen durchführen.

```
CALL ml_add_table_script(
  'ver1',
  'table1',
  'download_statistics',
  'INSERT INTO download_audit (
    user_name,
    table, warnings,
    errors,
    fetched_rows,
    deleted_rows,
    filtered_rows,
    bytes)
VALUES (
  {ml s.username},
  {ml s.table},
  {ml s.warnings},
  {ml s.errors},
  {ml s.fetched_rows},
  {ml s.deleted_rows},
  {ml s.filtered_rows},
  {ml s.bytes})')'
```

## Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens downloadStatisticsTable als das Skript für das Tabellenereignis download\_statistics, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_table_script(
  'ver1',
  'table1',
  'download_statistics',
  'ExamplePackage.ExampleClass.downloadStatisticsTable' )
```

Im Folgenden sehen Sie die Java-Beispielmethode downloadStatisticsTable. Sie protokolliert einige Statistiken für diese Tabelle im MobiLink-Nachrichtenlog. (Das Ausgeben von Statistiken für eine Tabelle in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
package ExamplePackage;
public class ExampleClass {
  String _curUser = null;
  public void downloadStatisticsTable(
    String user,
    String table,
    int warnings,
    int errors,
    int fetchedRows,
    int deletedRows,
    int filteredRows,
    int bytes ) {
    java.lang.System.out.println( "download table stats "
      + "table: " + table + "bytes: " + bytes );
  }
}
```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens DownloadTableStats als das Skript für das Tabellenereignis download\_statistics, wenn die Skriptversion ver1 und die Tabelle table1 synchronisiert werden.

```
CALL ml_add_dnet_table_script(  
  'ver1',  
  'table1',  
  'download_statistics',  
  'TestScripts.Test.DownloadTableStats'  
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode DownloadTableStats. Sie protokolliert einige Statistiken für diese Tabelle im MobiLink-Nachrichtenlog. (Das Ausgeben von Statistiken für eine Tabelle in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
namespace TestScripts {  
public class Test {  
  string _curUser = null;  
public void DownloadTableStats(  
  string user,  
  string table,  
  int warnings,  
  int errors,  
  int fetchedRows,  
  int deletedRows,  
  int filteredRows,  
  int bytes ) {  
  System.Console.WriteLine( "download table stats "  
    + "table: " + table + "bytes: " + bytes );  
  }  
}}
```

## end\_connection (Verbindungsereignis)

Verarbeitet Anweisungen unmittelbar bevor der MobiLink-Server die Verbindung mit dem Server der konsolidierten Datenbank trennt, entweder um das Herunterfahren vorzubereiten, oder wenn eine Verbindung aus dem Verbindungspool entfernt wird.

### Parameter

Keine.

### Standardaktion

Keine.

### Bemerkungen

Mit dem Skript end\_connection können Sie unmittelbar vor dem Schließen einer Verbindung zwischen dem MobiLink-Server und dem Server der konsolidierten Datenbank eine beliebige Aktion ausführen.

Dieses Skript wird normalerweise dazu verwendet, Aktionen abzuschließen, die vom Skript begin\_connection gestartet wurden, und von ihm beanspruchte Ressourcen freizugeben.

**Siehe auch**

- „begin\_connection (Verbindungsereignis)“ auf Seite 300
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258

**SQL-Beispiel**

Das nachstehende SQL-Skript löscht eine temporäre Tabelle, die mit dem Skript begin\_connection erstellt wurde. Diese Syntax gilt für eine konsolidierte SQL Anywhere-Datenbank. Diese Tabelle muss nicht explizit gelöscht werden, da SQL Anywhere dies bei der Löschung der Verbindung automatisch tun wird. Ob eine temporäre Tabelle explizit gelöscht werden muss, hängt vom Typ Ihrer konsolidierten Datenbank ab.

```
CALL ml_add_connection_script(
  'version 1.0',
  'end_connection',
  'DROP TABLE #sync_info' )
```

**Java-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens endConnection als das Skript für das Ereignis end\_connection, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_connection_script(
  'ver1',
  'end_connection',
  'ExamplePackage.ExampleClass.endConnection' )
```

Im Folgenden sehen Sie die Java-Beispielmethode endConnection. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog geschrieben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
package ExamplePackage;
public class ExampleClass {
    String _curUser = null;
    public void endConnection() {
        java.lang.System.out.println( "Ending connection." );
    }
}
```

**.NET-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens EndConnection als das Skript für das Verbindungsereignis end\_connection, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
  'ver1',
  'end_connection',
  'TestScripts.Test.EndConnection'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode EndConnection. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog geschrieben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
namespace TestScripts {
public class Test {
```

```
string _curUser = null;
public void EndConnection() {
    System.Console.WriteLine( "Ending connection." );
}}}
```

## end\_download (Verbindungsereignis)

Verarbeitet Anweisungen unmittelbar nachdem der MobiLink-Server die Vorbereitung der Download-Daten abgeschlossen hat.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.last_download                | TIMESTAMP. Die Zeit des ältesten Downloads einer synchronisierten Tabelle.                                                                                                                                                                                                   | 1                                 |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

**Standardaktion**

Keine.

**Bemerkungen**

Der MobiLink-Server führt dieses Skript aus, nachdem alle Download-Zeilen aus der konsolidierten Datenbank abgerufen wurden. Die Ausführung dieses Skripts ist die letzte nicht statistisch erfasste Aktion im Download.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „begin\_download (Verbindungsereignis)“ auf Seite 302
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109

**SQL-Beispiel**

Im folgenden Beispiel wird eine mögliche Verwendung des Verbindungsskripts begin\_download veranschaulicht. Dieses Skript löscht Zeilen aus einer temporären Tabelle, mit deren Hilfe der Download generiert wurde.

```
CALL ml_add_connection_script(
  'ver1',
  'end_download',
  'DELETE FROM TempDownloadTable where user = {ml s.username}')
```

**Java-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens endDownloadConnection als das Skript für das Verbindungsereignis end\_download, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_connection_script(
  'ver1',
  'end_download',
  'ExamplePackage.ExampleClass.endDownloadConnection' )
```

Im Folgenden sehen Sie die Java-Beispielmethode endDownloadConnection. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog geschrieben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
package ExamplePackage;
import java.sql.*;
public class ExampleClass {
    String _curUser = null;
    public void endDownloadConnection(
        Timestamp ts,
        String user )
    {
        java.lang.System.out.println( "Ending download for user: " + user );
    }
}
```

**.NET-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens EndDownload als das Skript für das Verbindungsereignis end\_download, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
  'ver1',
  'end_download',
  'TestScripts.Test.EndDownload' )
```

Im Folgenden sehen Sie die .NET-Beispielmethode EndDownload. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog geschrieben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
public void EndDownload(
  DateTime timestamp,
  string user ) {
  System.Console.WriteLine( "Ending download for user: " + user );
}
```

**end\_download (Tabellenereignis)**

Verarbeitet Anweisungen im Zusammenhang mit einer bestimmten Tabelle, unmittelbar nachdem der MobiLink-Server die Vorbereitung der Download-Zeilen abgeschlossen hat.

**Parameter**

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge (veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.last_table_download          | TIMESTAMP. Zeitpunkt des letzten Downloads der Tabelle.                                                                                 | 1                              |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                                 |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                                                                                                                                                               | 3                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

**Standardaktion**

Keine.

**Bemerkungen**

Der MobiLink-Server führt dieses Skript aus, nachdem alle Download-Zeilen aus der konsolidierten Datenbank abgerufen wurden. Die Ausführung dieses Skripts ist die letzte tabellenspezifische, nicht statistisch erfasste Aktion in der Downloadtransaktion.

Sie können für jede Tabelle in der entfernten Datenbank ein end\_download-Skript benutzen.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „begin\_download (Tabellenereignis)“ auf Seite 304
- „end\_download (Verbindungsereignis)“ auf Seite 342
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109

**SQL-Beispiel**

Das Tabellenereignis end\_download wird benutzt, um alle erforderlichen Schritte nach dem Download einer bestimmten Tabelle durchzuführen. Das nachstehende SQL Anywhere SQL-Skript löscht eine temporäre Tabelle, die von einem prepare\_for\_download-Skript erstellt wurde, um Downloadzeilen aus der Tabelle sales\_summary aufzunehmen.

```
CALL ml_add_table_script(
  'MyCorp 1.0',
  'sales_summary',
  'end_download',
  'DROP TABLE #sales_summary_download' )
```

## Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `endDownloadTable` als das Skript für das Tabellenereignis `end_download`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_table_script (
  'ver1',
  'table1',
  'end_download',
  'ExamplePackage.ExampleClass.endDownloadTable' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `endDownloadTable`. Sie setzt die member-Variable der aktuellen Tabelle zurück.

```
public void endDownloadTable(
  Timestamp ts,
  String user,
  String table ) {
  _curTable = null;
}
```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `EndTableDownload` als das Skript für das Tabellenereignis `end_download`, wenn die Skriptversion `ver1` und die Tabelle `table1` synchronisiert werden.

```
CALL ml_add_dnet_table_script(
  'ver1',
  'table1',
  'end_download',
  'TestScripts.Test.EndTableDownload'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode `EndTableDownload`. Sie setzt die member-Variable der aktuellen Tabelle zurück.

```
public void EndTableDownload
  DateTime timestamp,
  string user,
  string table ) {
  _curTable = null;
}}}
```

## end\_download\_deletes (Tabellenereignis)

Verarbeitet Anweisungen im Zusammenhang mit einer angegebenen Tabelle, unmittelbar nachdem in der entfernten Datenbank eine Liste von Zeilen vorbereitet wurde, die in dieser Tabelle gelöscht werden sollen.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.last_table_download          | TIMESTAMP. Zeitpunkt des letzten Downloads der Tabelle.                                                                                                                                                                                                                      | 1                                 |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                                 |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                                                                                                                                                               | 3                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

### Standardaktion

Keine.

### Bemerkungen

Dieses Skript wird unmittelbar nach der Erstellung der Liste von Zeilen ausgeführt, die aus der genannten Tabelle in der entfernten Datenbank gelöscht werden sollen.

**Hinweis**

Für jede Download-Tabelle werden die Ereignisse begin\_download\_deletes, download\_delete\_cursor und end\_download\_deletes nacheinander aufgerufen. Sie sollten in Betracht ziehen, die gesamte Download-Lösch-Logik für eine Tabelle in einem Ereignis download\_delete\_cursor in Form einer einzelnen gespeicherten Prozedur zu implementieren, die eine Ergebnismenge mit allen aus der entfernten Tabelle zu löschenden Zeilen zurückgibt. Die verringerte Anzahl der Skriptaufrufe führt möglicherweise zu einer verbesserten Download-Performance.

Sie können für jede Tabelle in der entfernten Datenbank ein end\_download\_deletes-Skript benutzen.

Siehe auch

- [„Skriptparameter“ auf Seite 236](#)
- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)
- [„begin\\_download\\_deletes \(Tabellenereignis\)“ auf Seite 306](#)
- [„end\\_download \(Verbindungsereignis\)“ auf Seite 342](#)
- [„begin\\_download\\_rows \(Tabellenereignis\)“ auf Seite 308](#)
- [„end\\_download\\_rows \(Tabellenereignis\)“ auf Seite 348](#)
- [„download\\_delete\\_cursor \(Tabellenereignis\)“ auf Seite 331](#)
- [„Entfernte IDs und MobiLink-Benutzernamen in Skripten“ \[\*MobiLink - Clientadministration\*\]](#)
- [„Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109](#)

## end\_download\_rows (Tabellenereignis)

Verarbeitet Anweisungen im Zusammenhang mit einer angegebenen Tabelle, unmittelbar nachdem in der entfernten Datenbank eine Liste mit einzufügenden oder zu aktualisierenden Zeilen aus dieser Tabelle vorbereitet wurde.

Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe [„SQL-Java-Datentypen“ auf Seite 466](#) und [„SQL-.NET-Datentypen“ auf Seite 541](#).

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                            | Reihenfolge (veraltet für SQL) |
|--------------------------------|---------------------------------------------------------|--------------------------------|
| s.last_table_download          | TIMESTAMP. Zeitpunkt des letzten Downloads der Tabelle. | 1                              |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                                 |
| s.table                        | VARCHAR(128). Der Tabellenname.                                                                                                                                                                                                                                              | 3                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

**Standardaktion**

Keine.

**Bemerkungen**

Dieses Skript wird ausgeführt, unmittelbar nachdem die Liste der Zeilen erstellt wurde, die in der genannten Tabelle in der entfernten Datenbank eingefügt oder aktualisiert werden sollen.

**Hinweis**

Für jede Download-Tabelle werden die Ereignisse begin\_download\_deletes, download\_delete\_cursor und end\_download\_deletes nacheinander aufgerufen. Sie sollten in Betracht ziehen, die gesamte Download-Lösch-Logik für eine Tabelle in einem Ereignis download\_delete\_cursor in Form einer einzelnen gespeicherten Prozedur zu implementieren, die eine Ergebnismenge mit allen aus der entfernten Tabelle zu löschenden Zeilen zurückgibt. Die verringerte Anzahl der Skriptaufrufe führt möglicherweise zu einer verbesserten Download-Performance.

Sie können für jede Tabelle in der entfernten Datenbank ein end\_download\_rows-Skript benutzen.

Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „begin\_download\_rows (Tabellenereignis)“ auf Seite 308
- „end\_download (Verbindungsereignis)“ auf Seite 342
- „end\_download\_deletes (Tabellenereignis)“ auf Seite 346
- „begin\_download\_deletes (Tabellenereignis)“ auf Seite 306
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109

end\_publication (Verbindungsereignis)

Bietet nützliche Informationen über die synchronisierten Publikationen.

Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „SQL-Java-Datentypen“ auf Seite 466 und „SQL-.NET-Datentypen“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge (veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.generation_number            | INTEGER. Wenn Ihr Deployment keine dateibasierten Downloads verwendet, kann dieser Parameter ignoriert werden. Der Standardwert ist 1.  | 1                              |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                | 2                              |
| s.publication_name             | VARCHAR(128). Der Name der Publikation                                                                                                  | 3                              |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.last_publication_upload      | TIMESTAMP. Zeit des letzten Uploads dieser Publikation.                                                                                                                                                                                                                      | 4                              |
| s.last_publication_download    | TIMESTAMP. Zeit des letzten Downloads dieser Publikation.                                                                                                                                                                                                                    | 5                              |
| s.subscription_id              | VARCHAR(128). Die entfernte Subskriptions-ID.                                                                                                                                                                                                                                | 6                              |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

**Standardaktion**

Keine.

**Bemerkungen**

Mit diesem Ereignis können Sie Synchronisationslogik basierend auf den gerade synchronisierten Publikationen entwerfen. Dieses Ereignis wird in derselben Transaktion wie das Ereignis end\_synchronization und vor dem Ereignis end\_synchronization aufgerufen. Es wird einmal pro synchronisierter Publikation aufgerufen.

Wenn die laufende Synchronisation einen Upload erfolgreich übernommen hat, enthält der Parameter last\_upload den Zeitpunkt der Übernahme dieses letzten Uploads. last\_publication\_download ist derselbe Wert, der als Uhrzeit des letzten Downloads an die Download-Skripten übergeben wurde.

Wenn eine entfernte UltraLite mit UL\_SYNC\_ALL synchronisiert wird, wird dieses Ereignis einmal mit dem Namen "unknown" aufgerufen.

**Generationsnummer**

Der Parameter generation\_number ist speziell für dateibasierte Downloads vorgesehen. Bei dateibasierten Downloads wird mithilfe von Änderungen an Generierungsnummern ein Upload vor dem Download erzwungen, wenn die Datei in der entfernten Datenbank übernommen wird. Die Nummer wird in der Download-Datei gespeichert.

Der Ausgabewert für die Generierungsnummer wird vom begin\_publication-Skript an das end\_publication-Skript übergeben. Die Bedeutung von generation\_number hängt davon ab, ob die aktuelle Synchronisation benutzt wird, um eine Download-Datei zu erstellen oder ob die aktuelle Synchronisation ein Upload benötigt.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „begin\_publication (Verbindungsereignis)“ auf Seite 310
- „MobiLink - dateibasierte Downloads“ auf Seite 213
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109

### SQL-Beispiel

Sie können die Informationen für jede synchronisierte Publikation aufzeichnen. Das folgende Beispiel ruft `ml_add_connection_script` auf, um das Ereignis der gespeicherten Prozedur `RecordPubEndSync` zuzuweisen.

```
CALL ml_add_connection_script(  
  'version1',  
  'end_publication',  
  'CALL RecordPubEndSync(  
    {ml s.generation_number},  
    {ml s.username},  
    {ml s.publication_name},  
    {ml s.last_publication_upload},  
    {ml s.last_publication_download} )' );
```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `endPublication` als das Skript für das Verbindungsereignis `end_publication`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_connection_script(  
  'ver1',  
  'end_publication',  
  'ExamplePackage.ExampleClass.endPublication' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `endPublication`. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog ausgegeben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
package ExamplePackage;  
import java.sql.*;  
public class ExampleClass {  
  String _curUser = null;  
  public void endPublication(  
    int generation_number,  
    String user,  
    String pub_name,  
    Timestamp last_publication_upload,  
    Timestamp last_publication_download ) {  
    java.lang.System.out.println(  
      "Finished synchronizing publication " + pub_name );  
  }  
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `EndPub` als das Skript für das Verbindungsereignis `end_publication`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
  'ver1',
  'end_publication',
  'TestScripts.Test.EndPub'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode endPub. Damit wird eine Nachricht ins MobiLink-Nachrichtenlog ausgegeben. (Das Ausgeben einer Meldung in das MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
public void EndPub(
    int generation_number,
    string user,
    string pub_name,
    DateTime last_publication_upload,
    DateTime last_publication_download ) {
    System.Console.Write(
        "Finished synchronizing publication " + pub_name );
}
```

## end\_synchronization (Verbindungsereignis)

Verarbeitet Anweisungen am Ende des Synchronisationsprozesses.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge (veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                | 1                              |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.synchronization_ok           | INTEGER. Der Wert ist 1 für eine erfolgreiche Synchronisation und 0 für eine nicht erfolgreiche Synchronisation.                                                                                                                                                             | 2                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

### Standardaktion

Keine.

### Bemerkungen

Der MobiLink-Server führt dieses Skript aus, nachdem die Synchronisation abgeschlossen wurde.

Dieses Skript wird innerhalb einer separaten Transaktion nach der Download-Transaktion ausgeführt. Wenn keine Downloadbestätigung erwartet wird, beendet die entfernte Datenbank möglicherweise die Synchronisation und trennt die Verbindung, bevor das Skript end\_synchronization beginnt oder abgeschlossen ist.

Das Skript end\_synchronization ist bei der Pflege von Statistiken hilfreich. Wenn nämlich das Skript begin\_synchronization aufgerufen wird, wird das Skript end\_synchronization auch dann aufgerufen, wenn ein Fehler in einer vorangegangenen Transaktion vorliegt. Während also die Upload-Transaktion zurückgesetzt wird, werden Statistikdaten beibehalten und gepflegt.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „begin\_synchronization (Verbindungsereignis)“ auf Seite 313
- „begin\_synchronization (Tabellenereignis)“ auf Seite 316
- „end\_synchronization (Tabellenereignis)“ auf Seite 355
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

### SQL-Beispiel

Das nachstehende SQL-Skript ruft eine Systemprozedur auf, die die Beendigungszeit der Synchronisation zusammen mit dem Status des Erfolgs oder Nichterfolgs aufzeichnet. Diese Syntax gilt für konsolidierte SQL Anywhere-Datenbanken.

```
CALL ml_add_connection_script(
    'ver1',
```

```
'end_synchronization',
'CALL RecordEndOfSyncAttempt(
  {ml s.username},
  {ml s.synchronization_ok} )' )
```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens endSynchronizationConnection als das Skript für das Verbindungsereignis end\_synchronization, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_connection_script(
  'ver1',
  'end_synchronization',
  'ExamplePackage.ExampleClass.endSynchronizationConnection'
)
```

Im Folgenden sehen Sie die Java-Beispielmethode endSynchronizationConnection. Sie verwendet eine JDBC-Verbindung, um eine Aktualisierung auszuführen. Diese Syntax gilt für konsolidierte SQL Anywhere-Datenbanken.

```
public void endSynchronizationConnection(
  String user )
  throws java.sql.SQLException {
  execUpdate( _syncConn,
    "UPDATE sync_count set count = count + 1 where user_id = '"
    + user + "' " );
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens EndSync als das Skript für das Verbindungsereignis end\_synchronization, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
  'ver1',
  'end_synchronization',
  'TestScripts.Test.EndSync'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode EndSync. Sie aktualisiert die Tabelle sync\_count. Diese Syntax gilt für konsolidierte SQL Anywhere-Datenbanken.

```
namespace TestScripts {
public class Test {
  string _curUser = null;
  public void EndSync(
    string user ) {
    return(
      "UPDATE sync_count set count = count + 1 where user_id = '"
      + user + "' " );
  }}
}
```

## end\_synchronization (Tabellenereignis)

Verarbeitet Anweisungen am Ende des Synchronisationsprozesses.

## Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 1                                 |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                                                                                                                                                               | 2                                 |
| s.synchronization_ok           | INTEGER. Die Wert ist 1 für eine erfolgreiche Synchronisation und 0 für eine nicht erfolgreiche Synchronisation.                                                                                                                                                             | 3                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

## Standardaktion

Keine.

## Bemerkungen

Der MobiLink-Server führt dieses Skript aus, nachdem eine Anwendung die Synchronisation abgeschlossen hat und dabei ist, die Verbindung zum MobiLink-Server zu trennen. Dies geschieht bevor das gleichnamige Skript der Verbindungsebene ausgeführt wird.

Sie können für jede Tabelle in der entfernten Datenbank ein end\_synchronization-Skript benutzen.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „begin\_synchronization (Tabellenereignis)“ auf Seite 316
- „end\_synchronization (Verbindungsereignis)“ auf Seite 353
- „end\_synchronization (Tabellenereignis)“ auf Seite 355
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

### SQL-Beispiel

Das nachstehende SQL Anywhere SQL-Skript löscht eine temporäre Tabelle, die mit dem Skript begin\_synchronization erstellt wurde.

```
CALL ml_add_table_script(
  'ver1',
  'sales_order',
  'end_synchronization',
  'DROP TABLE #sales_order' )
```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens endSynchronizationTable als das Skript für das Tabellenereignis end\_synchronization, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_table_script(
  'ver1',
  'table1',
  'end_synchronization',
  'ExamplePackage.ExampleClass.endSynchronizationTable' )
```

Im Folgenden sehen Sie die Java-Beispielmethode endSynchronizationTable.

```
package ExamplePackage;
import ianywhere.ml.script.*;
import java.sql.*;
public class ExampleClass {
    private DBConnectionContext _cc = null;
    public ExampleClass( DBConnectionContext cc ) {
        _cc = cc;
    }
    public void endSynchronizationTable() throws SQLException {
        try( Connection conn = _cc.getConnection() ) {
            try( PreparedStatement stmt =
conn.prepareStatement( "DROP TABLE #sales_order" ) ) {
                stmt.executeUpdate();
            }
        }
    }
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens EndTableSync als das Skript für das Tabellenereignis end\_synchronisation, wenn die Skriptversion ver1 und die Tabelle table1 synchronisiert werden.

```
CALL ml_add_dnet_table_script(
    'ver1',
    'table1',
    'end_synchronization',
    'TestScripts.Test.EndTableSync'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode EndSynchronizationTable.

```
using iAnywhere.MobiLink.Script;
namespace TestScripts {
    public class ExampleClass {
        DBConnectionContext _cc = null;
        ExampleClass( DBConnectionContext cc ) {
            _cc = cc;
        }
        public void EndSynchronizationTable() {
            DBConnection conn = _cc.GetConnection();
            try {
                DBCommand cmd = conn.CreateCommand();
                try {
                    cmd.CommandText = "DROP TABLE
#sales_order";

                    cmd.Prepare();
                    cmd.ExecuteNonQuery();
                } finally {
                    cmd.Close();
                }
            } finally {
                conn.Close();
            }
        }
    }
}
```

## end\_upload (Verbindungsereignis)

Verarbeitet Anweisungen unmittelbar nachdem der MobiLink-Server begonnen hat, den Upload-Datenstrom von Einfügungen, Aktualisierungen und Löschungen zu verarbeiten.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 1                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

**Standardaktion**

Keine.

**Bemerkungen**

Der MobiLink-Server führt dieses Skript als letzten Schritt bei der Verarbeitung von Upload-Daten aus. Upload-Informationen werden in einer einzigen Transaktion abgearbeitet. Die Ausführung dieses Skripts ist die letzte Aktion in dieser Transaktion, bevor statistische Skripten ausgeführt werden.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „begin\_upload (Verbindungsereignis)“ auf Seite 319
- „end\_upload (Tabellenereignis)“ auf Seite 361
- „Entfernte IDs und MobiLink-Bennutzernamen in Skripten“ [*MobiLink - Clientadministration*]

**SQL-Beispiel**

Das folgende SQL Anywhere SQL-Skript ruft die gespeicherte Prozedur EndUpload auf.

```
CALL ml_add_connection_script(
  'ver1',
  'end_upload',
  'CALL EndUpload({ml s.username});' )
```

**Java-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens endUploadConnection als das Skript für das Verbindungsereignis end\_upload, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_connection_script(
    'ver1',
    'end_upload',
    'ExamplePackage.ExampleClass.endUploadConnection' )
```

Im Folgenden sehen Sie die Java-Beispielmethode endUploadConnection. Sie ruft eine Methode auf, die die Datenbank bearbeitet.

```
public void endUploadConnection( String user ) {
    // Clean up new and old tables.
    Iterator two_iter = _tables_with_ops.iterator();
    while( two_iter.hasNext() ) {
        TableInfo cur_table = (TableInfo)two_iter.next();
        dumpTableOps( _sync_conn, cur_table );
    }
    _tables_with_ops.clear();
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens EndUpload als das Skript für das Verbindungsereignis end\_upload, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
    'ver1',
    'end_upload',
    'TestScripts.Test.EndUpload'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode EndUpload.

```
using iAnywhere.MobiLink.Script;
namespace TestScripts {
    public class ExampleClass {
        DBConnectionContext _cc = null;
        ExampleClass( DBConnectionContext cc ) {
            _cc = cc;
        }
        public void EndUpload( string userName ) {
            DBConnection conn = _cc.GetConnection();
            try {
                DBCommand cmd = conn.CreateCommand();
                try {
                    cmd.CommandText = "CALL EndUpload( ? )";
                    cmd.Prepare();
                    DBParameter parm = new DBParameter();
                    parm.DbType = SQLType.SQL_CHAR;
                    parm.Value = userName;
                    cmd.Parameters.Add( parm );
                    cmd.ExecuteNonQuery();
                } finally {
                    cmd.Close();
                }
            } finally {
                conn.Close();
            }
        }
    }
}
```

## end\_upload (Tabellenereignis)

Verarbeitet Anweisungen im Zusammenhang mit einer angegebenen Tabelle, unmittelbar nachdem der MobiLink-Server die Verarbeitung des Upload-Datenstroms der Einfügungen, Aktualisierungen und Löschungen abgeschlossen hat.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parameter        | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id      | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username       | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 1                                 |
| s.table          | VARCHAR(128). Der Tabellename.                                                                                                                                                                                                                                               | 2                                 |
| s.script_version | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

### Standardaktion

Keine.

### Bemerkungen

Der MobiLink-Server führt dieses Skript als letzten Schritt bei der Verarbeitung von Upload-Daten aus. Upload-Daten werden in einer separaten Transaktion abgearbeitet. Die Ausführung dieses Skripts ist die letzte tabellenspezifische Aktion in dieser Transaktion.

Sie können für jede Tabelle in der entfernten Datenbank ein end\_upload-Skript verwenden.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „begin\_upload (Tabellenereignis)“ auf Seite 321
- „end\_upload (Verbindungsereignis)“ auf Seite 358
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

### SQL-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur weist das Ereignis end\_upload der gespeicherten Prozedur ULCustomerIDPool\_maintain zu.

```
CALL ml_add_table_script(  
    'custdb',  
    'ULCustomerIDPool',  
    'end_upload',  
    '{ CALL ULCustomerIDPool_maintain( {ml s.username} ) }' )
```

Die folgenden SQL-Anweisungen erstellen die gespeicherte Prozedur ULCustomerIDPool\_maintain. Diese Prozedur fügt neue Primärschlüssel, als Ersatz für die von den hochgeladenen Zeilen verwendeten Schlüssel, in einen Primärschlüsselpool ein, der zu einem späteren Zeitpunkt in derselben Synchronisation in die entfernte Datenbank heruntergeladen wird.

```
CREATE PROCEDURE ULCustomerIDPool_maintain ( IN syncuser_id INTEGER )  
BEGIN  
    DECLARE pool_count INTEGER;  
  
    -- Determine how many ids to add to the pool  
    SELECT COUNT(*) INTO pool_count  
        FROM ULCustomerIDPool WHERE pool_emp_id = syncuser_id;  
  
    -- Top up the pool with new ids  
    WHILE pool_count < 20 LOOP  
        INSERT INTO ULCustomerIDPool ( pool_emp_id ) VALUES ( syncuser_id );  
        SET pool_count = pool_count + 1;  
    END LOOP;  
END
```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens endUploadTable als das Skript für das Tabellenereignis end\_upload, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_table_script(  
    'ver1',  
    'table1',  
    'end_upload',  
    'ExamplePackage.ExampleClass.endUploadTable' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `endUploadTable`. Sie erzeugt eine Löschung für eine Tabelle mit einem Namen, der dem Namen der angegebenen Tabelle entspricht. Diese Syntax gilt für konsolidierte SQL Anywhere-Datenbanken.

```
package ExamplePackage;
public class ExampleClass {
    String _curUser = null;
    public void endUploadTable(
        String user,
        String table ) {
        return( "DELETE from '" + table + "_temp'" );
    }
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `EndUpload` als das Skript für das Tabellenereignis `end_upload`, wenn die Skriptversion `ver1` und die Tabelle `table1` synchronisiert werden.

```
CALL ml_add_dnet_table_script(
    'ver1',
    'table1',
    'end_upload',
    'TestScripts.Test.EndUpload'
)
```

Das folgende .NET-Beispiel verschiebt Zeilen, die in eine temporäre Tabelle eingefügt wurden, und zwar in die Tabelle, die dem Skript übergeben wird.

```
using iAnywhere.MobiLink.Script;
namespace TestScripts
{
    public class Test
    {
        DBConnection _curConn = null;

        public Test( DBConnectionContext cc )
        {
            _curConn = cc.GetConnection();
        }
        public void EndUpload( string user, string table )
        {
            DBCommand stmt = _curConn.CreateCommand();
            // Move the uploaded rows to the destination table.
            stmt.CommandText = "INSERT INTO "
                + table
                + " SELECT * FROM dnet_ul_temp";
            stmt.ExecuteNonQuery();
            stmt.Close();
        }
    }
}
```

## end\_upload\_deletes (Tabellenereignis)

Verarbeitet Anweisungen im Zusammenhang mit einer angegebenen Tabelle, unmittelbar nachdem in der entfernten Datenbank Löschungen in dieser Tabelle ausgeführt wurden.

**Parameter**

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 1                                 |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                                                                                                                                                               | 2                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

**Standardaktion**

Keine.

**Bemerkungen**

Dieses Skript wird unmittelbar nach der Übernahme der Änderungen ausgeführt, die sich aus dem Löschen von Zeilen in der angegebenen entfernten Tabelle ergeben.

Sie können für jede Tabelle in der entfernten Datenbank ein end\_upload\_deletes-Skript benutzen.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „begin\_upload\_deletes (Tabellenereignis)“ auf Seite 324
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

**SQL-Beispiel**

Sie können dieses Ereignis einsetzen, um Zeilen zu verarbeiten, die während des Uploads in einer Zwischentabelle gelöscht wurden. Sie können die Zeilen in der Basistabelle mit Zeilen in der Zwischentabelle vergleichen und entscheiden, was mit der gelöschten Zeile geschehen soll.

Der folgende Aufruf einer MobiLink-Systemprozedur weist die gespeicherte Prozedur EndUploadDeletesLeads dem Ereignis end\_upload\_deletes zu.

```
CALL ml_add_table_script(
  'version1',
  'Leads',
  'end_upload_deletes',
  'call EndUploadDeletesLeads()');
```

Die folgende SQL-Anweisung erstellt die gespeicherte Prozedur EndUploadDeletes.

```
CREATE PROCEDURE EndUploadDeletesLeads ( )
Begin
  FOR names AS curs CURSOR FOR
    SELECT LeadID
      FROM Leads
     WHERE LeadID NOT IN (SELECT LeadID FROM T_Leads)
  DO
    CALL decide_what_to_do( LeadID )
  END FOR;
end
```

**Java-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens endUploadDeletes als das Skript für das Tabellenereignis end\_upload\_deletes, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_table_script(
  'ver1',
  'table1',
  'end_upload_deletes',
  'ExamplePackage.ExampleClass.endUploadDeletes' )
```

Im Folgenden sehen Sie die Java-Beispielmethode endUploadDeletes. Sie ruft eine Java-Methode auf, die die Datenbank verändert.

```
public void endUploadDeletes(
  String user,
  String table )
  throws java.sql.SQLException {
  processUploadedDeletes( _syncConn, table );
}
```

**.NET-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens EndUploadDeletes als das Skript für das Tabellenereignis end\_upload\_deletes, wenn die Skriptversion ver1 und die Tabelle table1 synchronisiert werden.

```
CALL ml_add_dnet_table_script(
    'ver1',
    'table1',
    'end_upload_deletes',
    'TestScripts.Test.EndUploadDeletes'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode EndUploadDeletes. Sie ruft eine .NET-Methode auf, die die Datenbank verändert.

```
namespace TestScripts {
public class Test {
    string _curUser = null;
    public void EndUploadDeletes(
        string user,
        string table ) {
        processUploadedDeletes( _syncConn, table );
    }}}
}
```

**end\_upload\_rows (Tabellenereignis)**

Verarbeitet Anweisungen im Zusammenhang mit einer angegebenen Tabelle, unmittelbar nachdem in der entfernten Datenbank aus dieser Tabelle übertragene Einfügungen und Aktualisierungen übernommen wurden.

**Parameter**

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe [„SQL-Java-Datentypen“ auf Seite 466](#) und [„SQL-.NET-Datentypen“ auf Seite 541](#).

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge (veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 1                                 |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                                                                                                                                                               | 2                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

**Standardaktion**

Keine.

**Bemerkungen**

Dieses Skript wird unmittelbar nach der Übernahme der Änderungen ausgeführt, die sich aus Änderungen an der angegebenen entfernten Tabelle ergeben.

Sie können für jede Tabelle in der entfernten Datenbank ein end\_upload\_rows-Skript benutzen.

**Siehe auch**

- „Hinzufügen und Löschungen von Skripten“ auf Seite 258
- „begin\_upload\_rows (Tabellenereignis)“ auf Seite 326
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

**SQL-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine SQL-Methode namens EndUploadRows als Skript für das Tabellenereignis end\_upload\_rows, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_table_script(
  'version1',
  'table1',
  'end_upload_rows',
  'CALL EndUploadRows(
    { ml s.username },
    { ml s.table } )' )
```

Im Folgenden sehen Sie die SQL-Beispielmethode EndUploadRows. Sie ruft eine SQL-Prozedur auf, die die Datenbank verändert.

```
CREATE PROCEDURE EndUploadRows (
  IN username VARCHAR(128)
  IN tablename VARCHAR{128} )
```

```
BEGIN
  CALL decide_what_to_do(tablename);
END;
```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens EndUploadRows als das Skript für das Tabellenereignis end\_upload\_rows, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_table_script(
  'ver1',
  'table1',
  'end_upload_rows',
  'ExamplePackage.ExampleClass.endUploadRows' )
```

Im Folgenden sehen Sie die Java-Beispielmethode endUploadRows. Sie ruft eine Java-Methode auf, die die Datenbank verändert.

```
public void endUploadRows(
  String user,
  String table )
  throws java.sql.SQLException {
  processUploadedRows( _syncConn, table );
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens EndUploadRows als das Skript für das Tabellenereignis end\_upload\_rows, wenn die Skriptversion ver1 und die Tabelle table1 synchronisiert werden.

```
CALL ml_add_dnet_table_script(
  'ver1',
  'table1',
  'end_upload_rows',
  'TestScripts.Test.EndUploadRows'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode endUploadRows. Sie ruft eine .NET-Methode auf, die die Datenbank verändert.

```
public void EndUploadRows(
  string user,
  string table ) {
  processUploadedRows( _syncConn, table );
}}}
```

## generate\_next\_last\_download\_timestamp (Ereignis)

Das Skript wird zum Aufrufen eines benutzerdefinierten Algorithmus zum Generieren des next\_last\_download\_timestamp-Werts verwendet.

## Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.next_last_download           | TIMESTAMP. Dies ist ein INOUT-Parameter. Der MobiLink-Server initialisiert diesen Parameter mit dem last_download_timestamp-Wert, einem Zeitstempel, mit dessen Hilfe ein Download-Datenstrom in der aktuellen Synchronisation generiert wird.                               | 1                                 |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

## Bemerkungen

Dieses Skript wird in der Transaktion prepare\_for\_download aufgerufen, unmittelbar bevor das Skript prepare\_for\_download aufgerufen wird.

Verwenden Sie dieses Ereignis mit Vorsicht, vor allem bei konsolidierten Datenbanken, die eine Snapshot-Isolationsstufe unterstützen, z.B. SQL Anywhere, Oracle, Microsoft SQL Server und IBM DB2 LUW 9.7. Der MobiLink-Server verwendet für den Download mit Oracle immer die Isolationsstufe snapshot. Standardmäßig wird die Isolationsstufe snapshot auch für den Download mit SQL Anywhere und Microsoft SQL Server verwendet, wenn sie in der Datenbank aktiviert ist.

Für eine zuverlässige zeitstempelbasierte Synchronisation muss die Ausgabe von `next_last_download` die frühere der folgenden Zeitangaben sein:

1. der aktuelle Zeitstempel
2. der Start-Zeitstempel der frühesten offenen Transaktion zum Aktualisieren (z.B. Einfügen, Aktualisieren oder Löschen) einer Tabelle oder Ansicht, die beim Erstellen des Downloads verwendet wird.

Dieses Skript kann auch mit der Klausel `--{ml_ignore}` als ignoriertes Skript angegeben werden. Wenn dieses Skript als ignoriertes Skript definiert ist, ruft der MobiLink-Server es nicht auf und generiert nicht mithilfe MobiLink-interner Logik den nächsten Zeitstempel für den letzten Download. Stattdessen sendet der MobiLink-Server den letzten Download-Zeitstempel an den Client zurück, der vom Client in der aktuellen Synchronisation gesendet wurde. Sie können diese Methode für Synchronisationen verwenden, die immer für alle Synchronisationstabellen alle Zeilen aus der konsolidierten Datenbank herunterladen. Für die zeitbasierte Synchronisation auf Zeitstempelbasis, sollten Sie dieses Skript als echtes Skript definieren und mit der entsprechenden Geschäftslogik den nächsten Zeitstempel für den letzten Download generieren. Eine andere Möglichkeit besteht darin, kein Skript für dieses Ereignis zu definieren. In diesem Fall generiert der MobiLink-Server mithilfe seiner internen Logik den nächsten Zeitstempel für den letzten Download.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109
- „So werden Download-Zeitstempel generiert und verwendet“ auf Seite 110
- „`modify_next_last_download_timestamp` (Verbindungsereignis)“ auf Seite 395
- „`modify_last_download_timestamp` (Verbindungsereignis)“ auf Seite 392

### SQL-Beispiel

Das Skript `generate_next_last_download_timestamp` kann im MobiLink-Server verwendet werden, um auf der UTC-Zeit basierende Downloads zu erstellen. Im Folgenden finden Sie die Schritte zur Einrichtung eines auf der UTC-Zeit basierenden Downloads für Oracle:

1. Angenommen, Sie haben eine Synchronisationstabelle mit dem Namen **my\_table**, die wie folgt definiert ist:

```
CREATE TABLE my_table (  pk          INT PRIMARY KEY NOT NULL,
                          c1          VARCHAR(100) ,
                          last_modified  TIMESTAMP DEFAULT
SYS_EXTRACT_UTC( SYSTIMESTAMP )
)
```

- Erstellen Sie eine gespeicherte Prozedur mit dem Namen **GenerateNextDownloadTimestamp**, um die Startzeit der frühesten offenen Transaktion in UTC-Zeit aus der Oracle Datenbank abzurufen:

```
CREATE PROCEDURE GenerateNextDownloadTimestamp ( p_ts IN OUT TIMESTAMP )
AS
BEGIN
    SELECT SYS_EXTRACT_UTC( NVL( MIN( TO_TIMESTAMP( START_TIME,
    'mm/dd/rr hh24:mi:ss' ) ),
                                SYSTIMESTAMP ) )
    INTO p_ts FROM GV$TRANSACTION;
END;
```

- Rufen Sie das Skript ml\_add\_connect\_script auf, um das Skript zu installieren:

```
call ml_add_connection_script(
    'my_script_version',
    'generate_next_last_download_timestamp',
    '{ call GenerateNextDownloadTimestamp( {ml s.next_last_download} ) }' )
```

#### Hinweis

Die Login-ID für den MobiLink-Server muss das SELECT-Privileg für GV\_\$TRANSACTION haben.

## handle\_DownloadData (Verbindungsereignis)

Ein Nicht-SQL-Datenskript, das von der direkten Zeilenbehandlung zum Erstellen einer Gruppe von herunterzuladenden Zeilen verwendet wird.

#### Parameter

Keine.

#### Standardaktion

Keine.

#### Bemerkungen

Mit dem Ereignis handle\_DownloadData legen Sie fest, welche Vorgänge unter Verwendung der direkten Zeilenbehandlung in MobiLink-Clients heruntergeladen werden.

Mit der direkten Zeilenbehandlung werden Datenquellen synchronisiert, bei denen es sich nicht um konsolidierte MobiLink-Datenbanken handelt. Siehe „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

Sie erstellen den direkten Download mithilfe der Klassen DownloadData und DownloadTableData in der API für Java oder .NET des MobiLink-Servers.

Für Java gibt die Methode DBConnectionContext getDownloadData eine DownloadData-Instanz für die aktuelle Synchronisation zurück. DownloadData umfasst alle Download-Vorgänge, die an einen entfernten Client gesendet werden. Mit den Methoden DownloadData getDownloadTables und getDownloadTableByName können Sie eine DownloadTableData-Instanz abrufen. DownloadTableData umfasst Download-Vorgänge für eine bestimmte Tabelle. Sie können mit der

getUpsertPreparedStatement-Methode vorbereitete Anweisungen für Einfügungs- und Aktualisierungsvorgänge abrufen. Mit der Methode DownloadTableData getDeletePreparedStatement rufen Sie vorbereitete Anweisungen für Löschvorgänge ab.

Für .NET gibt die Methode DBConnectionContext getDownloadData eine DownloadData-Instanz für die aktuelle Synchronisation zurück. DownloadData umfasst alle Download-Vorgänge, die an einen entfernten Client gesendet werden. Mit den Methoden DownloadData getDownloadTables und GetDownloadTableByName können Sie eine DownloadTableData-Instanz abrufen. DownloadTableData umfasst Download-Vorgänge für eine bestimmte Tabelle. Sie können mit der Methode GetUpsertPreparedStatement Befehle für Einfügungs- und Aktualisierungsvorgänge abrufen. Mit der Methode DownloadTableData getDeleteCommand rufen Sie Befehle für Löschvorgänge ab.

Weitere Hinweise zu Java finden Sie unter [DBConnectionContext-Schnittstelle \[MobiLink-Server Java\]](#) auf Seite 477.

Weitere Hinweise zu .NET finden Sie unter [DBConnectionContext-Schnittstelle \[MobiLink-Server .NET\]](#) auf Seite 565.

Sie können den Download in handle\_DownloadData oder einem anderen Synchronisationsereignis erstellen. MobiLink ist so flexibel, dass Sie den Download einrichten können, wenn Daten hochgeladen werden oder wenn bestimmte Ereignisse auftreten. Wenn Sie den direkten Download in einem anderen Ereignis als handle\_DownloadData erstellen möchten, müssen Sie ein handle\_DownloadData-Skript erstellen, dessen Methode keinen Vorgang ausführt. Außer bei reinen Upload-Synchronisationen verlangt der MobiLink-Server, dass mindestens ein Skript handle\_DownloadData definiert ist, um eine direkte Zeilenbehandlung von Downloads zu ermöglichen.

Falls Sie den direkten Download in einem anderen Ereignis als handle\_DownloadData erstellen, darf das Ereignis nicht vor dem Ereignis begin\_synchronization und nicht nach dem Ereignis end\_download stattfinden.

**Hinweis**

Dieses Ereignis kann nicht als SQL implementiert werden.

**Siehe auch**

- „Datenskripten“ auf Seite 283
- „Direkte Zeilenbehandlung“ auf Seite 627
- „Direkte Downloads“ auf Seite 637
- DownloadData-Schnittstelle [MobiLink-Server Java] auf Seite 483
- DownloadTableData-Schnittstelle [MobiLink-Server Java] auf Seite 485
- DownloadData-Schnittstelle [MobiLink-Server .NET] auf Seite 586
- DownloadTableData-Schnittstelle [MobiLink-Server .NET] auf Seite 588
- „handle\_UploadData (Verbindungsereignis)“ auf Seite 383
- „Erforderliche Skripten“ auf Seite 258
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258

## Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `handleDownload` für das Verbindungsereignis `handle_DownloadData`, wenn die Skriptversion `ver1` synchronisiert wird. Sie führen diese Systemprozedur für Ihre konsolidierte MobiLink-Datenbank aus.

```
CALL ml_add_java_connection_script(
  'ver1',
  'handle_DownloadData',
  'MyPackage.MobiLinkOrders.handleDownload' )
```

Siehe „[ml\\_add\\_java\\_connection\\_script-Systemprozedur](#)“ auf Seite 655.

Im folgenden Beispiel wird gezeigt, wie Sie mit der `handleDownload`-Methode einen Download erstellen.

Der folgende Code richtet eine Instanz der Klassenebene `DBConnectionContext` im Konstruktor für eine Klasse namens `MobiLinkOrders` ein.

```
import ianywhere.ml.script.*;
import java.io.*;
import java.sql.*;
import java.lang.System;

public class MobiLinkOrders{

    DBConnectionContext _cc;

    public MobiLinkOrders( DBConnectionContext cc ) {
        _cc = cc;
    }
}
```

In der Methode `HandleDownload` geben Sie mit der Methode `DBConnectionContext getDownloadData` eine `DownloadData`-Instanz für die aktuelle Synchronisation zurück. Die Methode `DownloadData getDownloadTableByName` gibt eine `DownloadTableData`-Instanz für die `remoteOrders`-Tabelle zurück. Die Methode `DownloadTableData getUpsertPreparedStatement` gibt ein `java.sql.PreparedStatement` zurück. Um dem Download einen Vorgang hinzuzufügen, richten Sie alle Spaltenwerte ein und rufen die Methode `executeUpdate` auf.

Beim Folgenden handelt es sich um die Methode `handleDownload` der Klasse `MobiLinkOrders`. Sie fügt dem Download zwei Zeilen für eine Tabelle namens `remoteOrders` hinzu.

```
// Method used for the handle_DownloadData event.
public void handleDownload() throws SQLException {
    // Get DownloadData instance for current synchronization.
    DownloadData downloadData = _cc.getDownloadData();

    // Get a DownloadTableData instance for the remoteOrders table.
    DownloadTableData td = downloadData.getDownloadTableByName("remoteOrders");

    // Get a java.sql.PreparedStatement for upsert (update/insert) operations.
    PreparedStatement upsertPS = td.getUpsertPreparedStatement();

    // Set values for one row.
    upsertPS.setInt( 1, 2300 );
    upsertPS.setInt( 2, 100 );

    // Add the values to the download.
    int updateResult = upsertPS.executeUpdate();
}
```

```
// Set values for another row.
upsertPS.setInt( 1, 2301 );
upsertPS.setInt( 2, 50 );
updateResult = upsertPS.executeUpdate();
// ...
upsertPS.close();
}
```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens HandleDownload als das Skript für das Verbindungsereignis handle\_DownloadData, wenn die Skriptversion ver1 synchronisiert wird. Diese Syntax gilt für konsolidierte SQL Anywhere-Datenbanken.

```
CALL ml_add_dnet_connection_script(
    'ver1', 'handle_DownloadData',
    'TestScripts.MobiLinkOrders.HandleDownload'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode HandleDownload.

```
using System;
using System.Data;
using System.IO;
using iAnywhere.MobiLink.Script;
using iAnywhere.MobiLink;

namespace MyScripts
{
    /// <summary>
    /// Tests that scripts are called correctly for most sync events.
    /// </summary>
    public class MobiLinkOrders
    {
        private DBConnectionContext _cc;

        public MobiLinkOrders( DBConnectionContext cc )
        {
            _cc = cc;
        }

        ~MobiLinkOrders()
        {
        }

        public void handleDownload()
        {
            // Get DownloadData instance for current synchronization.
            DownloadData my_dd = _cc.GetDownloadData();

            // Get a DownloadTableData instance for the remoteOrders table.
            DownloadTableData td = my_dd.GetDownloadTableByName("remoteOrders");

            // Get an IDbCommand for upsert (update/insert) operations.
            IDbCommand upsert_stmt = td.GetUpsertCommand();

            IDataParameterCollection parameters = upsert_stmt.Parameters;

            // Set values for one row.
            parameters[ 0 ] = 2300;
            parameters[ 1 ] = 100;
        }
    }
}
```

```

// Add the values to the download.
int update_result = upsert_stmt.ExecuteNonQuery();

// Set values for another row.
parameters[ 0 ] = 2301;
parameters[ 1 ] = 50;
update_result = upsert_stmt.ExecuteNonQuery();

// ...
}
}
}

```

## handle\_error (Verbindungsereignis)

Wird ausgeführt, wenn der MobiLink-Server beim Aufrufen eines Datenskripts auf einen SQL-Fehler stößt.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.action_code                  | INTEGER. Dies ist ein INOUT-Parameter. Setzen Sie diesen Wert, um den MobiLink-Server anzuweisen, wie er auf den Fehler reagieren soll. | 1                                 |
| s.error_code                   | INTEGER. Der native RDBMS-Fehlercode                                                                                                    | 2                                 |
| s.error_message                | TEXT. Die native RDBMS-Fehlermeldung.                                                                                                   | 3                                 |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                   |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 4                                 |
| s.table                        | VARCHAR(128). Die Tabelle, deren Skript den Fehler verursacht hat. Wenn das Skript kein Tabellenskript ist, ist der Tabellename NULL.                                                                                                                                        | 5                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

### Standardaktion

Der MobiLink-Server wählt eine Standardaktion. Sie können die Aktion im Skript ändern und einen Wert zurückgeben, der MobiLink anweist, wie fortgefahren werden soll. Der action\_code-Parameter hat einen der folgenden Werte:

- **1000** Aktuelle Zeile überspringen und Verarbeitung fortsetzen.
- **3000** Aktuelle Transaktion zurücksetzen und aktuelle Synchronisation abbrechen. Dies ist der Standard-Aktionscode. Er wird verwendet, wenn kein handle\_error-Skript definiert ist oder dieses Skript einen Fehler verursacht.
- **4000** Aktuelle Transaktion zurücksetzen, die Synchronisation abbrechen und den MobiLink-Server herunterfahren.

### Bemerkungen

Der MobiLink-Server sendet im aktuellen Aktionscode. Anfangs ist dieser für jede Gruppe von Fehlern, die durch einen SQL-Vorgang verursacht wurden, auf 3000 festgelegt. Normalerweise tritt pro SQL-Vorgang nur ein Fehler auf, es können jedoch auch mehr sein. Wenn Zeilen in Batches hochgeladen werden, wird dieses Skript handle\_error einmal pro Fehler im Batch aufgerufen. Während derselben Synchronisation wird der Aktionscode 3000 in den ersten Fehler übergeben. Nachfolgende Aufrufe werden in den Aktionscode übergeben, der vom vorigen Aufruf zurückgegeben wurde. Bei mehreren Aufrufen verwendet MobiLink den numerisch höchsten Wert.

Weitere Hinweise zu Batch-Uploads von Zeilen finden Sie unter „[mlsrv16-Option -s](#)“ auf Seite 77.

Sie können den Aktionscode im Skript ändern und einen Wert zurückgeben, der MobiLink anweist, wie weitergearbeitet werden soll. Der Aktionscode vermittelt dem MobiLink-Server, was als Nächstes zu tun ist. Bevor der MobiLink-Server dieses Skript aufruft, setzt er den Aktionscode auf einen Standardwert,

der davon abhängt, wie schwerwiegend der Fehler ist. Ihr Skript kann diesen Wert ändern. Ihr Skript muss einen Aktionscode zurückgeben.

Anhand des Fehlercodes und der Fehlermeldung können Sie die Art des Fehlers bestimmen.

Der MobiLink-Server führt dieses Skript aus, wenn ein ODBC-Fehler auftritt, während MobiLink ein Einfügungs-, Aktualisierungs- oder Lösch-Skript während des Uploads verarbeitet oder Downloadzeilen abrufen. Wenn ein ODBC-Fehler zu einem anderen Zeitpunkt auftritt, ruft der MobiLink-Server das Skript `report_error` oder `report_ODBC_error` auf und bricht die Synchronisation ab.

Ist der Fehler aufgetreten, während eine bestimmte Tabelle bearbeitet wurde, wird der Tabellename geliefert. Andernfalls ist dieser Wert NULL. Der Tabellename ist der Name einer Tabelle in der Clientanwendung. Dieser Name verfügt möglicherweise nicht über eine direkte Entsprechung in der konsolidierten Datenbank, je nachdem, wie Ihre entfernten Tabellennamen den konsolidierten Tabellen zugeordnet werden.

SQL-Skripten für das Ereignis `handle_error` müssen als gespeicherte Prozedur implementiert werden.

Sie können einen Wert aus dem Skript `handle_error` folgende Arten zurückgeben.

- Übergeben Sie den `action_code`-Parameter an den `OUTPUT`-Parameter einer Prozedur:

```
CALL my_handle_error( {ml s.action_code}, {ml s.error_code}, {ml
s.error_message}, {ml s.username}, {ml s.table} )
```

- Stellen Sie den `action_code`-Parameter über den Rückgabewert einer Prozedur oder einer Funktion ein:

```
{ml s.action_code} = CALL my_handle_error( {ml s.error_code}, {ml
s.error_message}, {ml s.username}, {ml s.table} )
```

Die meisten RDBMS verwenden die `RETURN`-Anweisung, um den Rückgabewert einer Prozedur oder einer Funktion festzulegen.

Die Beispielanwendung `CustDB` enthält Fehlerbehandlungsroutinen für verschiedene Datenbank-Managementsysteme.

## Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „`report_error` (Verbindungsereignis)“ auf Seite 410
- „`report_odbc_error` (Verbindungsereignis)“ auf Seite 414
- „`handle_odbc_error` (Verbindungsereignis)“ auf Seite 379
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Datenskripten“ auf Seite 283

## SQL-Beispiel

Das folgende Beispiel eignet sich für eine konsolidierte SQL Anywhere-Datenbank. Es gestattet Ihrer Anwendung, redundante Einfügungen zu ignorieren.

Der nachfolgende Aufruf einer MobiLink-Systemprozedur weist die gespeicherte Prozedur `ULHandleError` dem Ereignis `handle_error` zu.

```
CALL ml_add_connection_script(
  'ver1',
  'handle_error',
  'CALL ULHandleError(
    {ml s.action_code},
    {ml s.error_code},
    {ml s.error_message},
    {ml s.username},
    {ml s.table} )' )
```

Die folgende SQL-Anweisung erstellt die gespeicherte Prozedur ULHandleError.

```
CREATE PROCEDURE ULHandleError(
  INOUT action integer,
  IN error_code integer,
  IN error_message varchar(1000),
  IN user_name varchar(128),
  IN table_name varchar(128) )
BEGIN
  -- -196 is SQLE_INDEX_NOT_UNIQUE
  -- -194 is SQLE_INVALID_FOREIGN_KEY
  IF error_code = -196 or error_code = -194 then
    -- ignore the error and keep going
    SET action = 1000;
  ELSE
    -- abort the synchronization
    SET action = 3000;
  END IF;
END
```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens handleError als das Skript für das Verbindungsereignis handle\_error, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_connection_script(
  'ver1',
  'handle_error',
  'ExamplePackage.ExampleClass.handleError' )
```

Im Folgenden sehen Sie die Java-Beispielmethode handleError. Sie verarbeitet einen Fehler basierend auf den Daten, die übergeben werden. Außerdem legt sie den sich daraus ergebenden Fehlercode fest.

```
package ExamplePackage;
public class ExampleClass
{
    public void handleError( ianywhere.ml.script.InOutInteger    actionCode,
                           int                                   errorCode,
                           String                                errorMessage,
                           String                                user,
                           String                                table )
    {
        // -196 is SQLE_INDEX_NOT_UNIQUE
        // -194 is SQLE_INVALID_FOREIGN_KEY
        if( errorCode == -196 || errorCode == -194 ) {
            // ignore the error and keep going
            actionCode.setValue( 1000 );
        } else {
            // abort the synchronization
            actionCode.setValue( 3000 );
        }
    }
}
```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens HandleError als das Skript für das Verbindungsereignis handle\_error, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
  'ver1',
  'handle_error',
  'TestScripts.Test.HandleError' )
```

Im Folgenden sehen Sie die .NET-Beispielmethode HandleError.

```
namespace TestScripts
{
    public class Test
    {
        public void HandleError( ref int      actionCode,
                                int          errorCode,
                                string        errorMessage,
                                string        user,
                                string        table )
        {
            // -196 is SQLE_INDEX_NOT_UNIQUE
            // -194 is SQLE_INVALID_FOREIGN_KEY
            if( errorCode == -196 || errorCode == -194 ) {
                // ignore the error and keep going
                actionCode = 1000;
            } else {
                // abort the synchronization
                actionCode = 3000;
            }
        }
    }
}
```

## handle\_odbc\_error (Verbindungsereignis)

Wird ausgeführt, wenn der MobiLink-Server beim Aufrufen eines Datenskripts auf einen ODBC-Fehler stößt.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.action_code                  | INTEGER. Dies ist ein INOUT-Parameter. Setzen Sie diesen Wert, um den MobiLink-Server anzuweisen, wie er auf den Fehler reagieren soll.                                                                                                                                      | 1                              |
| s.odbc_state                   | VARCHAR(5). Der ODBC SQLSTATE-Wert.                                                                                                                                                                                                                                          | 2                              |
| s.error_message                | TEXT. Die ODBC-Fehlermeldung                                                                                                                                                                                                                                                 | 3                              |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 4                              |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                                                                                                                                                               | 5                              |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

### Standardaktion

Der MobiLink-Server wählt eine Standardaktion. Sie können die Aktion im Skript ändern und einen Wert zurückgeben, der MobiLink anweist, wie fortgefahren werden soll. Der action\_code-Parameter hat einen der folgenden Werte:

- **1000** Aktuelle Zeile überspringen und Verarbeitung fortsetzen.
- **3000** Aktuelle Transaktion zurücksetzen und aktuelle Synchronisation abbrechen. Dies ist der Standard-Aktionscode. Er wird verwendet, wenn kein handle\_error-Skript definiert ist oder dieses Skript einen Fehler verursacht.
- **4000** Aktuelle Transaktion zurücksetzen, die Synchronisation abbrechen und den MobiLink-Server herunterfahren.

### Bemerkungen

Der MobiLink-Server führt dieses Skript aus, wenn er auf einen Fehler stößt, der vom ODBC-Treibermanager markiert wurde, während MobiLink ein Einfüge-, Aktualisierungs- oder Löschskript bei der Uploadtransaktion verarbeitet oder Downloadzeilen abrufen. Wenn ein ODBC-Fehler zu einem anderen Zeitpunkt auftritt, ruft der MobiLink-Server das Skript report\_error oder report\_ODBC\_error auf und bricht die Synchronisation ab.

An Hand des Fehlercodes können Sie die Art des Fehlers feststellen.

Der Aktionscode vermittelt dem MobiLink-Server, was als Nächstes zu tun ist. Bevor der MobiLink-Server dieses Skript aufruft, setzt er den Aktionscode auf einen Standardwert, der davon abhängt, wie schwerwiegend der Fehler ist. Ihr Skript kann diesen Wert ändern. Ihr Skript muss einen Aktionscode zurückgeben.

Das Skript `handle_odbc_error` wird nach den Skripten `handle_error` und `report_error` und vor dem Skript `report_odbc_error` aufgerufen.

Wenn nur eines von beiden Fehlerbehandlungsskripten definiert ist, entscheidet der Rückgabewert von diesem Skript über das Fehlerbehandlungsverhalten. Wenn beide Fehlerbehandlungsskripte definiert sind, verwendet der MobiLink-Server den numerisch höchsten Aktionscode. Wenn sowohl `handle_error` als auch `handle_ODBC_error` definiert wurden, verwendet MobiLink den numerisch höchsten Aktionscode, der von den Aufrufen zurückgegeben wurde.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „`handle_error` (Verbindungsereignis)“ auf Seite 375
- „`report_error` (Verbindungsereignis)“ auf Seite 410
- „`report_odbc_error` (Verbindungsereignis)“ auf Seite 414
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

### SQL-Beispiel

Das folgende Beispiel eignet sich für eine konsolidierte SQL Anywhere-Datenbank. Damit wird Ihrer Anwendung gestattet, Verletzungen der ODBC-Integrität zu ignorieren.

Der nachfolgende Aufruf einer MobiLink-Systemprozedur weist die gespeicherte Prozedur `HandleODBCErr` dem Ereignis `handle_odbc_error` zu.

```
CALL ml_add_connection_script(
  'ver1',
  'handle_odbc_error',
  'CALL HandleODBCErr(
    {ml s.action_code},
    {ml s.ODBC_state},
    {ml s.error_message},
    {ml s.username},
    {ml s.table} )' )
```

Die folgende SQL-Anweisung erstellt die gespeicherte Prozedur `HandleODBCErr`.

```
CREATE PROCEDURE HandleODBCErr(
  INOUT action integer,
  IN odbc_state varchar(5),
  IN error_message varchar(1000),
  IN user_name varchar(128),
  IN table_name varchar(128) )
BEGIN
  IF odbc_state = '23000' then
    -- Ignore the error and keep going.
    SET action = 1000;
  ELSE
    -- Abort the synchronization.
    SET action = 3000;
```

```

        END IF;
    END

```

## Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `handleODBCError` als das Skript für das Ereignis `handle_odbc_error`, wenn die Skriptversion `ver1` synchronisiert wird.

```

CALL ml_add_java_connection_script(
    'ver1',
    'handle_odbc_error',
    'ExamplePackage.ExampleClass.handleODBCError'
)

```

Im Folgenden sehen Sie die Java-Beispielmethode `handleODBCError`. Sie verarbeitet einen Fehler basierend auf den Daten, die übergeben werden. Außerdem legt sie den sich daraus ergebenden Fehlercode fest.

```

package ExamplePackage;
public class ExampleClass
{
    public void handleODBCError( ianywhere.ml.script.InOutInteger
        actionCode,
                                String          odbcState,
                                String          errorMessage,
                                String          user,
                                String          table )
    {
        if( odbcState == "23000" ) {
            // Ignore the error and keep going.
            actionCode.setValue( 1000 );
        } else {
            // Abort the synchronization.
            actionCode.setValue( 3000 );
        }
    }
}

```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `HandleODBCError` als das Skript für das Ereignis `handle_odbc_error`, wenn die Skriptversion `ver1` synchronisiert wird.

```

CALL ml_add_dnet_connection_script(
    'ver1',
    'handle_odbc_error',
    'TestScripts.Test.HandleODBCError' )

```

Im Folgenden sehen Sie die .NET-Beispielmethode `HandleODBCError`.

```

namespace TestScripts
{
    public class Test
    {
        public void HandleODBCError( ref int    actionCode,
                                    string      odbcState,
                                    string      errorMessage,
                                    string      user,

```

```

        string table )
    {
        if( odbcState == "23000" ) {
            // Ignore the error and keep going.
            actionCode = 1000;
        } else {
            // Abort the synchronization.
            actionCode = 3000;
        }
    }
}

```

## handle\_UploadData (Verbindungsereignis)

Ein Nicht-SQL-Datenskript, das von der direkten Zeilenbehandlung zum Verarbeiten von hochgeladenen Zeilen verwendet wird.

### Parameter

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| UploadData                     | Eine .NET- oder Java-Klasse mit Tabellenvorgängen, die von einem MobiLink-Client hochgeladen wurden. Diese Klasse ist in der API für Java und .NET des MobiLink-Servers definiert.                                                                                           | 1                              |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

### Standardaktion

Keine.

### Bemerkungen

Das Ereignis handle\_UploadData ermöglicht Ihnen, den Upload für die direkte Zeilenbehandlung in MobiLink zu verarbeiten. Dieses Ereignis wird für jede Uploadtransaktion in einer Synchronisation einmal ausgelöst. Wenn Sie jedoch Uploads auf Transaktionsebene verwenden, wird das Ereignis für jede Transaktion ausgelöst.

Siehe „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

Dieses Ereignis verwendet nur einen UploadData-Parameter. Die Java- oder .NET-Methode kann mit den Methoden UploadData getUploadedTables oder getUploadedTableByName Instanzen von

UploadedTableData abrufen. Mit UploadedTableData greifen Sie auf Einfüge-, Aktualisierungs- und Löschvorgänge zu, die ein MobiLink-Client in die aktuelle Synchronisation hochgeladen hat.

Weitere Hinweise zu den Klassen UploadData und UploadedTableData finden Sie unter [„Direkte Uploads“ auf Seite 631](#).

Standardmäßig werden Spaltennamen bei der ersten Synchronisation mit einem MobiLink-Serverinstanz immer gesendet und anschließend vom MobiLink-Server im Cache abgelegt, damit sie nicht erneut gesendet werden müssen. Optional können Sie Spaltennamen mit der ml\_add\_column-Systemprozedur festlegen (nicht mehr empfohlen). Andernfalls können Sie Spalten nach Index referenzieren, wie in der entfernten Datenbank definiert.

Um die hochgeladenen Pre-Image-Spalten für eine Aktualisierung abzurufen, verwenden Sie die Methoden SetOldRowValues und SetNewRowValues. Siehe [„Konflikte bei direkten Uploads“ auf Seite 632](#).

### Hinweis

Dieses Ereignis kann nicht als SQL implementiert werden.

### Siehe auch

- [„Datenskripten“ auf Seite 283](#)
- [„Direkte Zeilenbehandlung“ auf Seite 627](#)
- [„Direkte Uploads“ auf Seite 631](#)
- [UploadData-Schnittstelle \[MobiLink-Server Java\] auf Seite 528](#)
- [UploadedTableData-Schnittstelle \[MobiLink-Server Java\] auf Seite 530](#)
- [UploadData-Schnittstelle \[MobiLink-Server .NET\] auf Seite 616](#)
- [UploadedTableData-Schnittstelle \[MobiLink-Server .NET\] auf Seite 618](#)
- [„handle\\_DownloadData \(Verbindungsereignis\)“ auf Seite 371](#)
- [„Erforderliche Skripten“ auf Seite 258](#)
- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)

### Java-Beispiele

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens handleUpload für das Verbindungsereignis handle\_UploadData, wenn die Skriptversion ver1 synchronisiert wird. Sie führen diese Systemprozedur für Ihre konsolidierte MobiLink-Datenbank aus.

```
CALL ml_add_java_connection_script(  
    'ver1',  
    'handle_UploadData',  
    'MyPackage.MyClass.handleUpload' )
```

Weitere Hinweise zu ml\_add\_java\_connection\_script finden Sie unter [„ml\\_add\\_java\\_connection\\_script-Systemprozedur“ auf Seite 655](#).

Die folgende Java-Methode verarbeitet den Upload für die remoteOrders-Tabelle. Die Methode UploadData.getUploadedTableByName gibt eine UploadedTableData-Instanz für die remoteOrders-Tabelle zurück. Die Methode UploadedTableData.getInserts gibt eine java.sql.ResultSet-Instanz mit neuen Zeilen zurück.

```

package MyPackage;
import ianywhere.ml.script.*;
import java.sql.*;
import java.io.*;
// ...
public class MyClass {
    String _curUser = null;
    public void handleUpload( UploadData ut )
        throws SQLException, IOException {
        // Get an UploadedTableData instance representing the
        // remoteOrders table.
        UploadedTableData remoteOrdersTable =
        ut.getUploadedTableByName("remoteOrders");
        // Get inserts uploaded by the MobiLink client.
        java.sql.ResultSet results = remoteOrdersTable.getInserts();
        while( results.next() ) {
            // Get the primary key.
            int pk = results.getInt("pk");

            // Get the uploaded num_ordered value.
            int numOrdered = results.getInt("num_ordered");

            // The current insert row is now ready to be uploaded to wherever
            // you want it to go (a file, a web service, and so on).

        }

        results.close();
    }
}

```

Das folgende Beispiel gibt Einfüge-, Aktualisierungs- und Löschvorgänge aus, die von einer entfernten MobiLink-Datenbank hochgeladen wurden. Die Methode UploadData getUploadedTables gibt UploadedTableData-Instanzen zurück, die alle Tabellen repräsentieren, die von einem entfernten Client hochgeladen wurden. Die Reihenfolge der Tabellen in diesem Array ist die Reihenfolge, in der sie vom entfernten Client hochgeladen wurden. Die Methoden UploadedTableData getInserts, getUpdates und getDeletes geben Standard-JDBC-Ergebnismengen zurück. Sie können die Methode println benutzen oder die Daten in eine Textdatei oder an einer anderen Stelle ausgeben.

```

import ianywhere.ml.script.*;
import java.sql.*;
import java.io.*;
// ...

public void handleUpload( UploadData ud )
    throws SQLException, IOException {
    UploadedTableData tables[] = ud.getUploadedTables();
    for( int i = 0; i < tables.length; i++ ) {
        UploadedTableData currentTable = tables[i];
        println( "table " + java.lang.Integer.toString( i ) +
        " name: " + currentTable.getName() );
        // Print out insert result set.
        println( "Inserts" );
        printRSInfo( currentTable.getInserts() );
        // print out update result set
        println( "Updates" );
        printUpdateRSInfo( currentTable.getUpdates() );
        // Print out delete result set.
        println( "Deletes" );
        printRSInfo( currentTable.getDeletes() );
    }
}

```

```
}
}
```

Die Methode `printRSInfo` druckt die Ergebnismenge der Einfüge-, Aktualisierungs- oder Löschvorgänge aus und akzeptiert ein einzelnes `java.sql.ResultSet`-Objekt. Ausführliche Spalteninformationen einschließlich Spaltenlabel liefert das Objekt `ResultSetMetaData`, das von der Methode `ResultSet.getMetaData` zurückgegeben wird. Die Methode `printRow` druckt jede Zeile in einer Ergebnismenge aus.

```
public void printRSInfo( ResultSet results )
    throws SQLException, IOException {

    // Obtain the result set metadata.
    ResultSetMetaData metaData = results.getMetaData();
    String columnHeading = "";

    // Print out column headings.
    for( int c = 1; c <= metaData.getColumnCount(); c++ ) {
        columnHeading += metaData.getColumnLabel(c);
        if( c < metaData.getColumnCount() ) {
            columnHeading += ", ";
        }
    }

    println( columnHeading );
    while( results.next() ) {

        // Print out each row.
        printRow( results, metaData.getColumnCount() );

        // Close the java.sql.ResultSet.
        results.close();
    }
}
```

Die nachfolgend aufgeführte Methode `printRow` ruft jeden Spaltenwert mit der Methode `ResultSet.getString` ab.

```
public void printRow( ResultSet results, int colCount )
    throws SQLException, IOException {
    String row = "( ";

    for( int c = 1; c <= colCount; c++ ) {
        // Get a column value.
        String currentColumn = results.getString( c );

        // Check for null values.
        if( currentColumn == null ) {
            currentColumn = "<NULL>";
        }

        // Add the column value to the row string.
        row += cur_col;
        if( c < colCount ) {
            row += ", ";
        }
    }

    row += " )";

    // Print out the row.
    println( row );
}
```

**.NET-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens HandleUpload für das Verbindungsereignis handle\_UploadData, wenn die Skriptversion ver1 synchronisiert wird. Sie führen diese Systemprozedur für Ihre konsolidierte MobiLink-Datenbank aus.

```
CALL ml_add_dnet_connection_script(
    'ver1',
    'handle_UploadData',
    'TestScripts.Test.HandleUpload' )
```

Die folgende .NET-Methode verarbeitet den Upload für die remoteOrders-Tabelle. Im folgenden Beispiel werden die Methoden SetOldRowValues und SetNewRowValues verwendet, um sowohl auf das Pre-Image als auch auf das Post-Image jeder Aktualisierung zuzugreifen.

```
using System;
using System.Data;
using System.IO;
using iAnywhere.MobiLink.Script;
using iAnywhere.MobiLink;

namespace MyScripts
{
    public class MyUpload
    {
        public MyUpload( DBConnectionContext cc )
        {
        }

        ~MyUpload()
        {
        }

        public void handleUpload( UploadData ut )
        {
            int i;
            UploadedTableData[] tables = ut.GetUploadedTables();

            for( i=0; i<tables.Length; i+=1 ) {
                UploadedTableData cur_table = tables[i];
                Console.Write( "table " + i + " name: " + cur_table.GetName() );

                // Print out insert result set.
                Console.Write( "Inserts" );
                printRSInfo( cur_table.GetInserts() );

                // print out update result set
                Console.Write( "Updates" );
                printUpdateRSInfo( cur_table.GetUpdates() );

                // Print out delete result set.
                Console.Write( "Deletes" );
                printRSInfo( cur_table.GetDeletes() );
            }
        }

        public void printRSInfo( IDataReader dr )
        {
            // Obtain the result set metadata.
            DataTable dt = dr.GetSchemaTable();
            DataColumnCollection cc = dt.Columns;
            DataColumn dc;
```

```

        String columnHeading = "";

        // Print out column headings.
        for( int c=0; c < cc.Count; c = c + 1 ) {
            dc = cc[ c ];
            columnHeading += dc.ColumnName;
            if( c < cc.Count - 1 ) {
                columnHeading += ", ";
            }
        }
        Console.Write( columnHeading );

        while( dr.Read() ) {
            // Print out each row.
            printRow( dr, cc.Count );
        }

        // Close the java.sql.ResultSet.
        dr.Close();
    }

    public void printUpdaterRSInfo( UpdateDataReader utr )
    {
        // Obtain the result set metadata.
        DataTable dt = utr.GetSchemaTable();
        DataColumnCollection cc = dt.Columns;
        DataColumn dc;
        String columnHeading = "TYPE, ";

        // Print out column headings.
        for( int c = 0; c < cc.Count; c = c + 1 ) {
            dc = cc[ c ];
            columnHeading += dc.ColumnName;
            if( c < cc.Count - 1 ) {
                columnHeading += ", ";
            }
        }
        Console.Write( columnHeading );

        while( utr.Read() ) {
            // Print out the new values for the row.
            utr.SetNewRowValues();
            Console.Write( "NEW:" );
            printRow( utr, cc.Count );

            // Print out the old values for the row.
            utr.SetOldRowValues();
            Console.Write( "OLD:" );
            printRow( utr, cc.Count );
        }

        // Close the java.sql.ResultSet.
        utr.Close();
    }

    public void printRow( IDataReader dr, int col_count )
    {
        String row = "( ";
        int c;

        for( c = 0; c < col_count; c = c + 1 ) {
            // Get a column value.
            String cur_col = dr.GetString( c );

```

```

// Check for null values.
if( cur_col == null ) {
    cur_col = "<NULL>";
}

// Add the column value to the row string.
row += cur_col;
if( c < col_count ) {
    row += ", ";
}
}

row += " )";

// Print out the row.
Console.Write( row );
}
}
}

```

## modify\_error\_message (Verbindungsereignis)

Das Skript kann verwendet werden, um den Meldungstext (Fehler, Warnung und Information), der an die entfernten Datenbanken gesendet wird, anzupassen.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge (veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.error_message                | VARBINARY(1024). Dies ist ein INOUT-Parameter, der die Fehlermeldung repräsentiert.                                                     | 1                              |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für<br>SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                                    |
| s.error_code                   | INTEGER. Der mit der Fehlermeldung verknüpfte MobiLink-Fehlercode.                                                                                                                                                                                                           | 3                                    |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                      |

### Standardaktion

Keine.

### Bemerkungen

Dieses Skript bietet Ihnen die Möglichkeit, die Fehlermeldung so zu ändern, dass der entfernte Benutzer und/oder die Anwendung sie besser verstehen kann als die ursprüngliche Meldung.

SQL-Skripten für das Ereignis `modify_error_message` müssen als gespeicherte Prozeduren implementiert werden.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [[MobiLink - Clientadministration](#)]

### SQL-Beispiel

Das folgende Beispiel liest alle Daten vom Vortag ein, unabhängig davon, ob die Datenbanken seitdem synchronisiert wurden.

Die folgende SQL-Anweisung erstellt die gespeicherte Prozedur `ModifyLastErrorMessage`.

```
CREATE PROCEDURE ModifyLastErrorMessage(
    inout error_message VARBINARY(1024),
    in username VARCHAR(128),
    in error_code INT )
BEGIN
    SELECT dateadd(day, -1, last_download_time )
    INTO last_download_time
END
```

Der folgende Aufruf einer MobiLink-Systemprozedur weist die Prozedur `ModifyLastErrorMessage` dem Verbindungsereignis `modify_error_message` für die Skriptversion `modify_ts_test` zu:

```
CALL ml_add_connection_script(
  'modify_ts_test',
  'modify_error_message',
  'CALL ModifyLastErrorMessage (
    {ml s.error_message},
    {ml s.username},
    {ml s.error_code} )' );
```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `modifyLastErrorMessage` als das Skript für das Verbindungsereignis `modify_error_message`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_connection_script(
  'ver1',
  'modify_error_message',
  'ExamplePackage.ExampleClass.modifyLastErrorMessage' );
```

Im Folgenden sehen Sie die Java-Beispielmethode `modifyLastErrorMessage`. Sie druckt die aktuelle Fehlermeldung und den Fehlercode aus.

```
package ExamplePackage;
public class ExampleClass {
    String _curUser = null;
    public void modifyLastErrorMessage(
        anywhere.ml.script.InOutString lastErrorMessage,
        String userName,
        int errorCode ) {
        java.lang.System.out.println( "error message: " +
            lastErrorMessage );
        java.lang.System.out.println( "error code: " +
            String.valueOf(errorCode) );
    }
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `ModifyLastErrorMessage` als das Skript für das Verbindungsereignis `modify_error_message`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
  'ver1',
  'modify_error_message',
  'TestScripts.Test.ModifyLastErrorMessage' );
```

Im Folgenden sehen Sie die .NET-Beispielmethode `ModifyLastErrorMessage`. Sie druckt den aktuellen Fehlercode und die Fehlermeldung aus.

```
namespace TestScripts {
    public class Test {
        string _curUser = null;
        public void ModifyLastErrorMessage (
            ref string errorMessage,
            string userName,
            string errorCode ) {
            System.Console.WriteLine( "error message: " + errorMessage );
        }
    }
}
```

```
System.Console.WriteLine( "error code: " + errorCode );
}}}
```

## modify\_last\_download\_timestamp (Verbindungsereignis)

Benutzen Sie dieses Skript, wenn Sie die Uhrzeit des letzten Downloads für die aktuelle Synchronisation ändern wollen.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.last_download                | TIMESTAMP. Die Zeit des ältesten Downloads einer synchronisierten Tabelle. Dies ist ein INOUT-Parameter.                                                                                                                                                                     | 1                              |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                              |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

**Standardaktion**

Keine.

**Bemerkungen**

Verwenden Sie dieses Skript, um den Zeitstempel `last_download` für die aktuelle Synchronisation zu ändern. Wenn dieses Skript definiert ist, verwendet der MobiLink-Server den geänderten `last_download`-Zeitstempel als den an die Download-Skripten übergebenen `last_download`-Zeitstempel. Dieses Skript wird gewöhnlich dazu verwendet, verlorene Daten in der entfernten Datenbank wiederherzustellen. Sie können den Zeitstempel `last_download` auf eine frühere Uhrzeit wie z.B. 1900-01-01 00:00 zurücksetzen, sodass bei der nächsten Synchronisation alle Daten heruntergeladen werden. Außerdem gilt: Wenn Aktualisierungen an konsolidierten Tabellen Zeitstempel zulassen, die vor der Uhrzeit der tatsächlichen Aktualisierung liegen, z.B. über DBMS-Replikation, können Sie mit diesem Skript den Zeitpunkt des letzten Downloads anpassen, um zu vermeiden, dass diese Aktualisierungen beim Download fehlen.

SQL-Skripten für das Ereignis `modify_last_download_timestamp` müssen als gespeicherte Prozeduren implementiert werden.

Dieses Skript wird direkt vor dem `prepare_for_download`-Skript in derselben Transaktion ausgeführt.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109
- „So werden Download-Zeitstempel generiert und verwendet“ auf Seite 110
- „`modify_next_last_download_timestamp` (Verbindungsereignis)“ auf Seite 395
- „`generate_next_last_download_timestamp` (Ereignis)“ auf Seite 368

**SQL-Beispiel**

Die folgende SQL-Anweisung erstellt eine gespeicherte Prozedur. Die folgende Syntax gilt für konsolidierte Oracle-Datenbanken. Wenn Sie eine gespeicherte Prozedur in Oracle erstellen, die einen Parameter entgegennimmt und übergibt, müssen Sie sicherstellen, dass der Parameter als **IN OUT** markiert ist, wie unten gezeigt wird:

```
CREATE OR REPLACE PROCEDURE ModifyDownloadTimestamp (
    download_timestamp IN OUT TIMESTAMP,
    user_name IN VARCHAR)
AS
BEGIN
    -- N is the maximum replication latency in consolidated cluster
    download_timestamp := download_timestamp - 1;
END;
```

Die folgende Syntax gilt für konsolidierte SQL Anywhere-, Adaptive Server Enterprise- und SQL Server-Datenbanken.

```
CREATE PROCEDURE ModifyDownloadTimestamp
    @download_timestamp DATETIME OUTPUT,
    @user_name VARCHAR( 128 )
AS
BEGIN
```

```
-- N is the maximum replication latency in consolidated cluster
SELECT @download_timestamp = @download_timestamp - N
END
```

Der folgende Aufruf einer MobiLink-Systemprozedur weist die gespeicherte Prozedur `ModifyDownloadTimestamp` dem Ereignis `modify_last_download_timestamp` zu. Die folgende Syntax gilt für eine konsolidierte SQL Anywhere-Datenbank.

```
CALL ml_add_connection_script(
  'my_version',
  'modify_last_download_timestamp',
  '{CALL ModifyDownloadTimestamp(
    {ml s.last_download},
    {ml s.username} ) }' )
```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `modifyLastDownloadTimestamp` als das Skript für das Verbindungsereignis `modify_last_download_timestamp`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_connection_script(
  'ver1',
  'modify_last_download_timestamp',
  'ExamplePackage.ExampleClass.modifyLastDownloadTimestamp' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `modifyLastDownloadTimestamp`. Sie gibt den aktuellen und den neuen Zeitstempel aus und ändert den Zeitstempel, der übergeben wird.

```
public void modifyLastDownloadTimestamp(
    Timestamp lastDownloadTime,
    String userName ) {
    java.lang.System.out.println( "old date: " +
        lastDownloadTime.toString() );
    lastDownloadTime.setDate(
        lastDownloadTime.getDate() -1 );
    java.lang.System.out.println( "new date: " +
        lastDownloadTime.toString() );
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `ModifyLastDownloadTimestamp` als das Skript für das Verbindungsereignis `modify_last_download_timestamp`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
  'ver1',
  'modify_last_download_timestamp',
  'TestScripts.Test.ModifyLastDownloadTimestamp' )
```

Im Folgenden sehen Sie die .NET-Beispielmethode `ModifyLastDownloadTimestamp`.

```
public void ModifyLastDownloadTimestamp(
    ref DateTime lastDownloadTime,
    string userName ) {
    System.Console.WriteLine( "old date: " +
        last_download_time.ToString() );
    last_download_time = DateTime.Now;
    System.Console.WriteLine( "new date: " +
```

```
last_download_time.ToString() );
}
```

## modify\_next\_last\_download\_timestamp (Verbindungsereignis)

Benutzen Sie dieses Skript, wenn Sie die Uhrzeit des letzten Downloads für die nächste Synchronisation ändern möchten.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.next_last_download           | TIMESTAMP. Dies ist ein INOUT-Parameter. Der MobiLink-Server generiert diesen Wert direkt nach Festschreibung des Uploads.              | 1                                 |
| s.last_download                | TIMESTAMP. Dies ist die Uhrzeit des letzten Downloads für die aktuelle Synchronisation.                                                 | 2                                 |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                | 3                                 |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

**Standardaktion**

Keine.

**Bemerkungen**

Das Skript kann zur Änderung des Zeitstempels `next_last_download` verwendet werden, der den Zeitstempel `last_download` für die nächste Synchronisation ändert. Dies ermöglicht Ihnen, die nächste Synchronisation ohne Auswirkung auf die aktuelle Synchronisation zurückzusetzen. Während der normalen Synchronisation ist die `next_last_download`-Zeit später als die `last_download`-Zeit. Manchmal sind die beiden Werte aber auch gleich.

SQL-Skripten für das Ereignis `modify_next_last_download_timestamp` müssen als gespeicherte Prozeduren implementiert werden. Der MobiLink-Server übergibt den `next_last_download`-Zeitstempel als ersten Parameter an die gespeicherte Prozedur und ersetzt den Zeitstempel durch den ersten Wert, der von der gespeicherten Prozedur übergeben wird.

Dieses Skript wird in der Download-Transaktion ausgeführt, nach dem Herunterladen von Benutzertabellen, aber der Ausgabewert Ihrer gespeicherten Prozedur sollte dem Beginn der Download-Transaktion entsprechen, damit die während der Download-Transaktion geänderte Zeilen bei der nächsten Synchronisation heruntergeladen werden.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109
- „So werden Download-Zeitstempel generiert und verwendet“ auf Seite 110
- „`modify_last_download_timestamp` (Verbindungsereignis)“ auf Seite 392
- „`generate_next_last_download_timestamp` (Ereignis)“ auf Seite 368

**SQL-Beispiel**

Folgendes Beispiel zeigt eine Anwendung für dieses Skript. Erstellen Sie eine gespeicherte Prozedur. Die folgende Syntax gilt für eine konsolidierte SQL Anywhere-Datenbank.

```
CREATE PROCEDURE ModifyNextDownloadTimestamp(  
    inout next_last_download TIMESTAMP ,
```

```

in last_download TIMESTAMP ,
in user_name VARCHAR(128) )
BEGIN
    SELECT dateadd(hour, -1, next_last_download )
    INTO next_last_download
END

```

Installieren Sie das Skript in der konsolidierten SQL Anywhere-Datenbank:

```

CALL ml_add_connection_script(
    'modify_ts_test',
    'modify_next_last_download_timestamp',
    'CALL ModifyNextDownloadTimestamp (
        {ml s.next_last_download},
        {ml s.last_download},
        {ml s.username} )' )

```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `modifyNextDownloadTimestamp` als das Skript für das Verbindungsereignis `modify_next_last_download_timestamp`, wenn die Skriptversion `ver1` synchronisiert wird.

```

CALL ml_add_java_connection_script(
    'ver1',
    'modify_next_last_download_timestamp',
    'ExamplePackage.ExampleClass.modifyNextDownloadTimestamp' )

```

Im Folgenden sehen Sie die Java-Beispielmethode `modifyNextDownloadTimestamp`. Sie setzt den Download-Zeitstempel um eine Stunde zurück.

```

public void modifyNextDownloadTimestamp(
    Timestamp NextLastDownload,
    Timestamp lastDownload,
    String userName ) {
    NextLastDownload.setHours(
        NextLastDownload.getHours() -1 );
}

```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `ModifyNextDownloadTimestamp` als das Skript für das Verbindungsereignis `modify_next_last_download_timestamp`, wenn die Skriptversion `ver1` synchronisiert wird.

```

CALL ml_add_dnet_connection_script(
    'ver1',
    'modify_next_last_download_timestamp',
    'TestScripts.Test.ModifyNextDownloadTimestamp' )

```

Im Folgenden sehen Sie die .NET-Beispielmethode `ModifyNextDownloadTimestamp`. Sie setzt den Download-Zeitstempel um eine Stunde zurück.

```

using System;
using System.Data;
namespace TestScripts {
    public class Test {
        String _curUser = null;
        public void ModifyNextDownloadTimestamp (
            ref DateTime next_last_download,

```

```
DateTime last_download,
string user_name ) {
next_last_download = next_last_download.AddHours( -1 );
}}}
```

# modify\_user (Verbindungsereignis)

Ändert den MobiLink-Benutzernamen.

## Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „SQL-Java-Datentypen“ auf Seite 466 und „SQL-.NET-Datentypen“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername. Dies ist ein INOUT-Parameter.                                                                                                                                                                                                       | 1                              |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

## Standardaktion

Keine.

## Bemerkungen

Dieses Skript wird am Ende der Authentifizierungstransaktion aufgerufen.

Der MobiLink-Server gibt den Benutzernamen als Parameter an, wenn er Skripten aufruft. Der Benutzername wird vom MobiLink-Client gesendet. Manchmal kann es sinnvoll sein, einen anderen Benutzernamen zu verwenden. Mit diesem Skript können Sie den Benutzernamen ändern, der beim Aufruf von MobiLink-Skripten verwendet wird.

Der username-Parameter muss lang genug sein, um den Benutzernamen zu speichern.

SQL-Skripten für das Ereignis `modify_user` müssen als gespeicherte Prozeduren implementiert werden.

### Hinweis

Eine flexiblere Methode zur Zuordnung des MobiLink-Benutzernamens ist die Verwendung von benutzerdefinierten benannten Parametern. Siehe „[Benutzerdefinierte benannte Parameter](#)“ auf Seite 252.

## Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „`authenticate_user` (Verbindungsereignis)“ auf Seite 291
- „`authenticate_user_hashed` (Verbindungsereignis)“ auf Seite 296
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

## SQL-Beispiel

Im folgenden Beispiel wird der Name eines Benutzers einer entfernten Datenbank der ID des Benutzers zugeordnet, der dieses Gerät verwendet, indem es eine Zuordnungstabelle namens `user_device` verwendet. Diese Technik kann benutzt werden, wenn dieselbe Person mehrere entfernte Datenbanken betreibt (z.B. einen PDA und einen Laptop), für die dieselbe Synchronisationslogik erforderlich ist (basierend auf dem Namen oder der ID des Benutzers).

Der folgende Aufruf einer MobiLink-Systemprozedur weist die gespeicherte Prozedur `ModifyUser` dem Ereignis `modify_user` zu. Diese Syntax gilt für eine konsolidierte SQL Anywhere-Datenbank.

```
CALL ml_add_connection_script(
  'ver1',
  'modify_user',
  'call ModifyUser( {ml s.username} )' )
```

Die folgende SQL-Anweisung erstellt die gespeicherte Prozedur `ModifyUser`.

```
CREATE PROCEDURE ModifyUser( INOUT u_name varchar(128) )
BEGIN
  SELECT user_name
  INTO u_name
  FROM user_device
  WHERE device_name = u_name;
END
```

## Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `modifyUser` als das Skript für das Verbindungsereignis `modify_user`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_connection_script(  
    'ver1',  
    'modify_user',  
    'ExamplePackage.ExampleClass.modifyUser' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `modifyUser`. Sie erhält die Benutzer-ID von der Datenbank und verwendet sie dann, um den Benutzernamen festzulegen.

```
package ExamplePackage;  
import java.lang.Integer;  
import java.sql.*;  
import ianywhere.ml.script.*;  
public class ExampleClass  
{  
    DBConnectionContext curConn;  
  
    public ExampleClass( DBConnectionContext cc )  
    {  
        curConn = cc;  
    }  
  
    public void modifyUser( InOutString ioUserName )  
        throws SQLException  
    {  
        Connection conn = curConn.getConnection();  
        PreparedStatement uidSelect =  
            conn.prepareStatement( "SELECT rep_id FROM SalesRep WHERE name  
= ?" );  
        try {  
            uidSelect.setString( 1, ioUserName.getValue() );  
            ResultSet uidResult = uidSelect.executeQuery();  
            try {  
                if( uidResult.next() ) {  
                    ioUserName.setValue( Integer.toString(uidResult.getInt( 1 )) );  
                }  
            } finally {  
                uidResult.close();  
            }  
        } finally {  
            uidSelect.close();  
        }  
    }  
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `ModUser` als das Skript für das Verbindungsereignis `modify_user`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_dnet_connection_script(  
    'ver1',  
    'modify_user',  
    'TestScripts.Test.ModUser'  
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode `ModUser`.

```
using iAnywhere.MobiLink.Script;  
namespace TestScripts  
{  
    public class Test  
    {
```

```

DBConnectionContext curConn;

public Test( DBConnectionContext cc )
{
    curConn = cc;
}

public void ModifyUser( ref string ioUserName )
{
    DBCommand cmd = curConn.GetConnection().CreateCommand();
    cmd.CommandText = "SELECT rep_id FROM SalesRep WHERE name = ?";
    cmd.Parameters[0] = ioUserName;
    DBRowReader r = cmd.ExecuteReader();
    object[] row;
    if( (row = r.NextRow()) != null ) {
        ioUserName = (string) row[0];
    }
}
}

```

## nonblocking\_download\_ack (Verbindungsereignis)

Wenn Sie die Downloadbestätigung verwenden, können Sie in diesem Skript aufzeichnen, dass ein Download erfolgreich angewendet wurde, oder eine Geschäftslogik auslösen, die davon abhängt, dass der Download als angewendet bestätigt wurde.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge (veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                | 1                              |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für<br>SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| s.last_download                | TIMESTAMP. Dies ist die Uhrzeit des letzten Downloads für die aktuelle Synchronisation.                                                                                                                                                                                      | 2                                    |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                      |

### Bemerkungen

Mithilfe dieses Ereignisses können Sie die Uhrzeit aufzeichnen, zu der der Download erfolgreich in der entfernten Datenbank angewendet wurde.

Dieses Ereignis wird nur aufgerufen, wenn die nicht blockierende Downloadbestätigung verwendet wird. Im nicht blockierenden Modus wird die Download-Transaktion festgeschrieben und die Synchronisation beendet, wenn der Download gesendet wurde. Dieses Ereignis wird aufgerufen, wenn der Synchronisationsclient in einer neuen Verbindung den erfolgreichen Download bestätigt und nachdem das Skript end\_synchronization der ursprünglichen Synchronisation verarbeitet wurde. Die Aktionen dieses Ereignisses werden zusammen mit einer Aktualisierung der Downloadzeit in den MobiLink-Systemtabellen festgeschrieben.

Aufgrund der spezifischen Natur dieses Skripts sind während der Synchronisation keine Variablen auf Verbindungsebene verfügbar, wenn dieses Ereignis ausgeführt wird.

#### Hinweis

Wenn der Download fehlgeschlagen ist oder die Netzwerkverbindung getrennt wird, gibt es keine Bestätigung und dieses Skript wird nicht aufgerufen. Wenn die Downloadbestätigung für Ihr Geschäft entscheidend ist, sollten Sie den last\_download-Parameter des Skripts prepare\_for\_download oder den last\_publication\_download-Parameter des Skripts begin\_publication als Sicherung für die Verarbeitung Ihrer Downloadbestätigungen verwenden.

### Siehe auch

- „publication\_nonblocking\_download\_ack (Verbindungsereignis)“ auf Seite 407
- dbmlsync: „Erweiterte Option SendDownloadAck (sa)“ [*MobiLink - Clientadministration*]
- UltraLite: „Synchronisationsparameter Send Download Acknowledgement“ [*UltraLite - Datenbankverwaltung*]

## SQL-Beispiel

Das folgende Skript fügt der Tabelle `download_pubs_acked` einen Datensatz hinzu. Der Datensatz enthält die entfernte ID, den ersten Authentifizierungsparameter und den Download-Zeitstempel.

```
INSERT INTO download_pubs_acked( rem_id, auth_parm, last_download )
VALUES( {ml s.remote_id}, {ml a.1}, {ml s.last_download} )
```

## Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `confirmDownload` als Skript für das Tabellenereignis `nonblocking_download_ack`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_connection_script(
  'ver1',
  'nonblocking_download_ack',
  'ExamplePackage.ExampleClass.confirmDownload' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `confirmDownload`. Sie ruft eine Java-Methode auf, die auf der Basis des zu bestätigenden Downloads, bis zum angegebenen Zeitstempel und für den angegebenen Benutzer, die Geschäftslogik ausführt.

```
package ExamplePackage;
import ianywhere.ml.script.*;
import java.sql.*;
public class ExampleClass
{
    DBConnectionContext _cc;

    public ExampleClass( DBConnectionContext cc )
    {
        _cc = cc;
    }

    public void confirmDownload( String user,
                                Timestamp ts )
    throws SQLException
    {
        Connection conn = _cc.getConnection();
        PreparedStatement stmt = conn.prepareStatement(
            "INSERT INTO download_pubs_acked( rem_id, last_download ) " +
            "VALUES( ?, ? )" );
        stmt.setString( 1, _cc.getRemoteID() );
        stmt.setTimestamp( 2, ts );
        stmt.executeUpdate();
        stmt.close();
    }
}
```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `ConfirmDownload` als Skript für das Verbindungsereignis `nonblocking_download_ack`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
  'ver1',
  'nonblocking_download_ack',
  'TestScripts.Test.ConfirmDownload'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode ConfirmDownload. Sie ruft eine .NET-Methode auf, die auf der Basis des zu bestätigenden Downloads, bis zum angegebenen Zeitstempel und für den angegebenen Benutzer, die Geschäftslogik ausführt.

```
using System;
using iAnywhere.MobiLink.Script;
namespace TestScripts
{
    public class Test
    {
        DBConnectionContext _cc;

        public Test( DBConnectionContext cc )
        {
            _cc = cc;
        }

        public void ConfirmDownload( string user,
                                    DateTime dt )
        {
            DBConnection conn = _cc.GetConnection();

            DBCommand cmd = conn.CreateCommand();
            cmd.CommandText =
                "INSERT INTO download_pubs_acked( rem_id, last_download ) " +
                "VALUES( ?, ? )";
            cmd.Parameters[0] = _cc.GetRemoteID();
            cmd.Parameters[1] = dt;
            cmd.ExecuteNonQuery();
        }
    }
}
```

## prepare\_for\_download (Verbindungseignis)

Führt alle erforderlichen Vorgänge zwischen Upload- und Downloadtransaktionen aus.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.last_download                | TIMESTAMP. Die Zeit des ältesten Downloads einer synchronisierten Tabelle.                                                                                                                                                                                                   | 1                                 |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 2                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

**Standardaktion**

Keine.

**Bemerkungen**

Der MobiLink-Server führt dieses Skript zwischen der Upload-Transaktion und dem Anfang der Download-Transaktion als separate Transaktion aus .

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „end\_upload (Verbindungsereignis)“ auf Seite 358
- „begin\_download (Verbindungsereignis)“ auf Seite 302
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]
- „Zeitpunkt des letzten Downloads in Skripten“ auf Seite 109

**SQL-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine gespeicherte SQL-Prozedur namens prepareForDownload als Skript für das Ereignis prepare\_for\_download, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_connection_script(
    'ver1',
    'prepare_for_download',
```

```
'CALL prepareForDownload(  
  { ml s.username } )' )
```

Im Folgenden finden Sie die SQL-Beispielprozedur prepareForDownload. Diese gespeicherte Prozedur bereitet Downloads für zwei Tabellen vor. Zum Beispiel könnte sie Informationen aus vielen Tabellen in temporären Tabellen speichern, die von den download\_cursor-Skripten für die Tabellen T1 und T2 referenziert werden.

```
CREATE PROCEDURE prepareForDownload (  
  IN ts TIMESTAMP,  
  IN "user" VARCHAR(128))  
BEGIN  
  CALL prepareT1Download( user, ts );  
  CALL prepareT2Download( user, ts );  
END;
```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens prepareForDownload als das Skript für das Ereignis prepare\_for\_download, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_connection_script(  
  'ver1',  
  'prepare_for_download',  
  'ExamplePackage.ExampleClass.prepareForDownload' )
```

Im Folgenden sehen Sie die Java-Beispielmethode prepareForDownload. Diese Methode bereitet Downloads für zwei Tabellen vor. Zum Beispiel könnte sie Informationen aus vielen Tabellen sowie weitere über Java zugängliche Informationen in temporären Tabellen speichern, die von den download\_cursor-Skripten für die Tabellen T1 und T2 referenziert werden.

```
public void prepareForDownload(  
  Timestamp ts,  
  String user ) {  
  prepareT1ForDownload( _syncconn, user, ts );  
  prepareT2ForDownload( _syncconn, user, ts );  
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens PrepareForDownload als das Skript für das Verbindungsereignis prepare\_for\_download, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_dnet_connection_script(  
  'ver1',  
  'prepare_for_download',  
  'TestScripts.Test.PrepareForDownload'  
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode PrepareForDownload. Diese Methode bereitet Downloads für zwei Tabellen vor. Zum Beispiel könnte sie Informationen aus vielen Tabellen sowie weitere über .NET zugängliche Informationen in temporären Tabellen speichern, die von den download\_cursor-Skripten für die Tabellen T1 und T2 referenziert werden.

```
public void PrepareForDownload(  
  DateTime ts,
```

```
string user ) {
PrepareT1ForDownload ( _syncConn, user, ts );
PrepareT2ForDownload ( _syncConn, user, ts );
}
```

## publication\_nonblocking\_download\_ack (Verbindungsereignis)

Wenn Sie die Downloadbestätigung verwenden, können Sie in diesem Skript aufzeichnen, dass eine Publikation erfolgreich heruntergeladen wurde.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                | 1                                 |
| s.last_publication_download    | TIMESTAMP. Frühester Zeitstempel für den letzten Download einer synchronisierten Tabelle.                                               | 2                                 |
| s.publication name             | VARCHAR(128). Der Name der Publikation                                                                                                  | 3                                 |
| s.subscription_id              | VARCHAR(128). Die entfernte Subskriptions-ID.                                                                                           | 4                                 |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

### Bemerkungen

Mithilfe dieses Ereignisses können Sie die Zeit aufzeichnen, wann diese Publikation erfolgreich in der entfernten Datenbank angewendet wurde.

Dieses Ereignis wird nur aufgerufen, wenn die nicht blockierende Downloadbestätigung verwendet wird. Im nicht blockierenden Modus wird die Downloadtransaktion festgeschrieben, und die Synchronisation beendet, wenn der Download gesendet wurde, wenn der Synchronisationsclient in einer neuen Verbindung den erfolgreichen Download bestätigt und nachdem das Skript end\_synchronization der ursprünglichen Synchronisation verarbeitet wurde. Die Aktionen dieses Ereignisses werden zusammen mit einer Aktualisierung der Downloadzeit in den MobiLink-Systemtabellen festgeschrieben.

#### Hinweis

Wenn der Download fehlgeschlagen ist oder die Netzwerkverbindung getrennt wird, gibt es keine Bestätigung und dieses Skript wird nicht aufgerufen. Wenn die Downloadbestätigung für Ihr Geschäft entscheidend ist, sollten Sie den last\_download-Parameter des Skripts prepare\_for\_download oder den last\_publication\_download-Parameter des Skripts begin\_publication als Sicherung für die Verarbeitung Ihrer Downloadbestätigungen verwenden.

Aufgrund der spezifischen Natur dieses Skripts sind während der Synchronisation keine Variablen auf Verbindungsebene verfügbar, wenn dieses Ereignis ausgeführt wird.

### Siehe auch

- „nonblocking\_download\_ack (Verbindungsereignis)“ auf Seite 401
- dbmlsync: „Erweiterte Option SendDownloadAck (sa)“ [*MobiLink - Clientadministration*]
- UltraLite: „Synchronisationsparameter Send Download Acknowledgement“ [*UltraLite - Datenbankverwaltung*]

### SQL-Beispiel

Das folgende Skript fügt der Tabelle download\_pubs\_acked einen Datensatz hinzu. Der Datensatz enthält den Publikationsnamen, den ersten Authentifizierungsparameter und einen Download-Zeitstempel.

```
INSERT INTO download_pubs_acked( pub_name, auth_parm, last_download )
VALUES( {ml s.publication_name}, {ml a.1}, {ml
s.last_publication_download} )
```

## Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `publicationDownloadACK` als Skript für das Tabellenereignis `publication_nonblocking_download_ack`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_connection_script (
  'ver1',
  'publication_nonblocking_download_ack',
  'ExamplePackage.ExampleClass.publicationDownloadACK' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `publicationDownloadACK`. Sie führt eine Geschäftslogik aus und reagiert damit auf die Bestätigung, wenn eine besonders wichtig Publikation heruntergeladen wurde.

```
package ExamplePackage;
import ianywhere.ml.script.*;
import java.sql.*;
public class ExampleClass
{
    DBConnectionContext _cc;

    public ExampleClass( DBConnectionContext cc )
    {
        _cc = cc;
    }

    public void confirmDownload( String user,
                                Timestamp ts,
                                String pubName )
        throws SQLException
    {
        Connection conn = _cc.getConnection();
        PreparedStatement stmt = conn.prepareStatement(
            "INSERT INTO download_pubs_acked( rem_id, last_download, pub_name )
" +
            "VALUES( ?, ?, ? )" );
        stmt.setString( 1, _cc.getRemoteID() );
        stmt.setTimestamp( 2, ts );
        stmt.setString( 3, pubName );
        stmt.executeUpdate();
        stmt.close();
    }
}
```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `EndTableDownload` als Skript für das Tabellenereignis `end_download`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
  'ver1',
  'publication_nonblocking_download_ack',
  'TestScripts.Test.EndTableDownload'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode `EndTableDownload`. Sie führt eine Geschäftslogik aus und reagiert damit auf die Bestätigung, wenn eine besonders wichtig Publikation heruntergeladen wurde.

```
using System;
using iAnywhere.MobiLink.Script;
namespace TestScripts
{
    public class Test
    {
        DBConnectionContext _cc;

        public Test( DBConnectionContext cc )
        {
            _cc = cc;
        }

        public void ConfirmDownload( string user,
                                    DateTime dt,
                                    string pubName )
        {
            DBConnection conn = _cc.GetConnection();

            DBCommand cmd = conn.CreateCommand();
            cmd.CommandText =
                "INSERT INTO download_pubs_acked( rem_id, last_download,
pub_name ) " +
                "VALUES( ?, ?, ? )";
            cmd.Parameters[0] = _cc.GetRemoteID();
            cmd.Parameters[1] = dt;
            cmd.Parameters[2] = pubName;
            cmd.ExecuteNonQuery();
        }
    }
}
```

## report\_error (Verbindungsergebnis)

Mit diesem Skript können Sie Fehler protokollieren und die vom handle\_error-Skript ausgewählten Aktionen aufzeichnen.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe [„SQL-Java-Datentypen“ auf Seite 466](#) und [„SQL-.NET-Datentypen“ auf Seite 541](#).

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.action_code                  | INTEGER. Dies ist ein INOUT-Parameter. Dieser Parameter ist obligatorisch.                                                                                                                                                                                                   | 1                              |
| s.error_code                   | INTEGER. Der native DBMS-Fehlercode                                                                                                                                                                                                                                          | 2                              |
| s.error_message                | TEXT. Die native DBMS-Fehlermeldung.                                                                                                                                                                                                                                         | 3                              |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 4                              |
| s.table                        | VARCHAR(128). Die Tabelle, deren Skript den Fehler verursacht hat.                                                                                                                                                                                                           | 5                              |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

### Standardaktion

Keine.

### Bemerkungen

Mit diesem Skript können Sie Fehler protokollieren und die vom handle\_error-Skript ausgewählten Aktionen aufzeichnen. Dieses Skript wird nach dem Ereignis handle\_error ausgeführt, unabhängig davon, ob ein handle\_error-Skript definiert wurde. Es wird immer in einer eigenen Transaktion und in einer anderen Datenbankverbindung ausgeführt als die Synchronisationsverbindung (Verwaltungs- bzw. Informationsverbindung).

Der MobiLink-Server gibt immer eine Fehlermeldung aus, wenn der Fehler wiederhergestellt werden kann und der MobiLink-Server den Aufruf des Skripts handle\_error oder handle\_odbc\_error plant. Wenn zum Beispiel ein Fehler auftritt, während der MobiLink-Server versucht, eine Einfügung hochzuladen, meldet der MobiLink-Server diesen Fehler und ruft das Skript handle\_error auf. Wenn das Skript handle\_error den Aktionscode 1000 zurückgibt, ignoriert der Server den Fehler und fährt mit der Synchronisation fort. Wenn der MobiLink-Server jedoch vor dem Senden an die konsolidierte Datenbank einen Fehler erkennt, meldet der Server den Fehler möglicherweise nicht, weil der Fehler nicht wiederherstellbar ist. Genauer gesagt, meldet der MobiLink-Server die generierten Fehler, die vom ODBC-Treiber und von der konsolidierten Datenbank generiert wurden.

Anhand des Fehlercodes und der Fehlermeldung können Sie die Art des Fehlers bestimmen. Dies ist der vom letzten Aufruf eines Fehlerbehandlungsskripts zurückgegebene Aktionscodewert für den SQL-Vorgang, der den aktuellen Fehler verursacht hat.

Wenn der Fehler als Teil der Synchronisation aufgetreten ist, wird der Benutzername geliefert. Andernfalls ist dieser Wert NULL.

Ist der Fehler aufgetreten, während eine bestimmte Tabelle bearbeitet wurde, wird der Tabellename geliefert. Andernfalls ist dieser Wert NULL. Der Tabellename ist der Name einer Tabelle in der entfernten Datenbank. Dieser Name verfügt möglicherweise nicht über eine direkte Entsprechung in der konsolidierten Datenbank, je nachdem, wie Ihre entfernten Tabellennamen den Tabellennamen der konsolidierten Datenbank zugeordnet werden.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „handle\_error (Verbindungsereignis)“ auf Seite 375
- „handle\_odbc\_error (Verbindungsereignis)“ auf Seite 379
- „report\_odbc\_error (Verbindungsereignis)“ auf Seite 414
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [[MobiLink - Clientadministration](#)]

### SQL-Beispiel

Das folgende Beispiel eignet sich für eine konsolidierte SQL Anywhere-Datenbank. Es fügt eine Zeile in eine Tabelle ein, um Synchronisationsfehler aufzuzeichnen.

```
CALL ml_add_connection_script(  
  'ver1',  
  'report_error',  
  'INSERT INTO sync_error(  
    action_code,  
    error_code,  
    error_message,  
    user_name,  
    table_name )  
VALUES (  
  {ml s.action_code},  
  {ml s.error_code},  
  {ml s.error_message},  
  {ml s.username},  
  {ml s.table} )' )
```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens reportError als das Skript für das Verbindungsereignis report\_error, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_connection_script(  
  'ver1',  
  'report_error',  
  'ExamplePackage.ExampleClass.reportError' )
```

Im Folgenden sehen Sie die Java-Beispielmethode reportError. Sie protokolliert den Fehler in einer Tabelle über die JDBC-Verbindung, die von MobiLink bereitgestellt wird. Sie setzt auch den Aktionscode.

```

package ExamplePackage;
import java.sql.*;
import ianywhere.ml.script.*;
public class ExampleClass
{
    DBConnectionContext _cc;

    public ExampleClass( DBConnectionContext cc )
    {
        _cc = cc;
    }

    public void reportError( ianywhere.ml.script.InOutInteger    actionCode,
                           int                                   errorCode,
                           String                                errorMessage,
                           String                                user,
                           String                                table )
    throws SQLException
    {
        actionCode.setValue( errorCode );
        // Insert error information in a table,
        Connection conn = _cc.getConnection();
        PreparedStatement stmt = conn.prepareStatement(
            "INSERT INTO sync_error( action_code, error_code, error_message, " +
            "user_name, table_name ) VALUES ( ?, ?, ?, ?, ? )" );
        stmt.setInt( 1, actionCode.getValue() );
        stmt.setInt( 2, errorCode );
        stmt.setString( 3, errorMessage );
        stmt.setString( 4, user );
        stmt.setString( 5, table );
        stmt.executeUpdate();
        stmt.close();
    }
}

```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens ReportError als das Skript für das Verbindungsereignis report\_error, wenn die Skriptversion ver1 synchronisiert wird.

```

CALL ml_add_dnet_connection_script(
    'ver1',
    'report_error',
    'TestScripts.Test.ReportError' )

```

Im Folgenden sehen Sie die .NET-Beispielmethode ReportError. Sie protokolliert unter Verwendung einer .NET-Methode den Fehler in einer Tabelle.

```

using System;
using iAnywhere.MobiLink.Script;
namespace TestScripts
{
    public class Test
    {
        DBConnectionContext _cc;

        public Test( DBConnectionContext cc )
        {
            _cc = cc;
        }

        public void ReportError( ref int actionCode,
                                int errorCode,

```

```
        string errorMessage,
        string user,
        string table )
    {
        actionCode = errorCode;
        DBConnection connn = _cc.GetConnection();

        DBCommand cmd = connn.CreateCommand();
        cmd.CommandText =
            "INSERT INTO sync_error( action_code, error_code, error_message, " +
            "user_name, table_name ) VALUES ( ?, ?, ?, ?, ? )";
        cmd.Parameters[0] = actionCode;
        cmd.Parameters[1] = errorCode;
        cmd.Parameters[2] = errorMessage;
        cmd.Parameters[3] = user;
        cmd.Parameters[4] = table;
        cmd.ExecuteNonQuery();
    }
}
```

## report\_odbc\_error (Verbindungsereignis)

Mit diesem Skript können Sie Fehler protokollieren und die vom handle\_odbc\_error-Skript ausgewählten Aktionen aufzeichnen.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe [„SQL-Java-Datentypen“ auf Seite 466](#) und [„SQL-.NET-Datentypen“ auf Seite 541](#).

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge (veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.action_code                  | INTEGER. Dies ist ein INOUT-Parameter. Dieser Parameter ist obligatorisch.                                                              | 1                              |
| s.odbc_state                   | VARCHAR(5). Der ODBC SQLSTATE-Wert.                                                                                                     | 2                              |
| s.error_message                | TEXT. Die ODBC-Fehlermeldung                                                                                                            | 3                              |
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 4                              |
| s.table                        | VARCHAR(128). Die Tabelle, deren Skript den Fehler verursacht hat.                                                                                                                                                                                                           | 5                              |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

### Standardaktion

Keine.

### Bemerkungen

Mit diesem Skript können Sie Fehler protokollieren und die vom `handle_odbc_error`-Skript ausgewählten Aktionen auswählen. Dieses Skript wird nach dem Ereignis `handle_odbc_error` ausgeführt, unabhängig davon, ob ein `handle_odbc_error`-Skript definiert wurde. Es wird immer in einer eigenen Transaktion und in einer anderen Datenbankverbindung ausgeführt als die Synchronisationsverbindung (Verwaltungs- bzw. Informationsverbindung).

Anhand des ODBC-Status und der Fehlermeldung können Sie die Art des Fehlers bestimmen. Dies ist der vom letzten Aufruf eines Fehlerbehandlungsskripts zurückgegebene Aktionscodewert für den SQL-Vorgang, der den aktuellen Fehler verursacht hat.

Wenn der Fehler als Teil der Synchronisation aufgetreten ist, wird der Benutzername geliefert. Andernfalls ist dieser Wert NULL.

Ist der Fehler aufgetreten, während eine bestimmte Tabelle bearbeitet wurde, wird der Tabellename geliefert. Andernfalls ist dieser Wert NULL. Der Tabellename ist der Name einer Tabelle in der entfernten Datenbank. Dieser Name verfügt möglicherweise nicht über eine direkte Entsprechung in der konsolidierten Datenbank, je nachdem, wie Ihre entfernten Tabellennamen den Tabellennamen der konsolidierten Datenbank zugeordnet werden.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „`handle_error` (Verbindungsereignis)“ auf Seite 375
- „`handle_odbc_error` (Verbindungsereignis)“ auf Seite 379
- „`report_error` (Verbindungsereignis)“ auf Seite 410
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

## SQL-Beispiel

Das folgende Beispiel eignet sich für eine konsolidierte SQL Anywhere-Datenbank. Es fügt eine Zeile in eine Tabelle ein, um Synchronisationsfehler aufzuzeichnen.

```
CALL ml_add_connection_script(
  'ver1',
  'report_odbc_error',
  'INSERT INTO sync_error(
    action_code,
    odbc_state,
    error_message,
    user_name,
    table_name )
VALUES(
  {ml s.action_code},
  {ml s.odbc_state},
  {ml s.error_message},
  {ml s.username},
  {ml s.table} )' )
```

## Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens reportODBCError als das Skript für das Ereignis report\_odbc\_error, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_connection_script(
  'ver1',
  'report_odbc_error',
  'ExamplePackage.ExampleClass.reportODBCError' )
```

Im Folgenden sehen Sie die Java-Beispielmethode reportODBCError. Sie protokolliert den Fehler in einer Tabelle über die JDBC-Verbindung, die von MobiLink bereitgestellt wird. Sie setzt auch den Aktionscode.

```
package ExamplePackage;
import java.sql.*;
import ianywhere.ml.script.*;
public class ExampleClass
{
    DBConnectionContext _cc;

    public ExampleClass( DBConnectionContext cc )
    {
        _cc = cc;
    }

    public void reportODBCError( InOutInteger    actionCode,
                                String          odbcState,
                                String          odbcMessage,
                                String          user,
                                String          table )
    throws SQLException
    {
        // Insert error information in a table,
        Connection conn = _cc.getConnection();
        PreparedStatement stmt = conn.prepareStatement(
            "INSERT INTO sync_error( action_code, odbc_state, error_message, " +
            "user_name, table_name ) VALUES ( ?, ?, ?, ?, ? )" );
        stmt.setInt( 1, actionCode.getValue() );
        stmt.setString( 2, odbcState );
```

```

        stmt.setString( 3, odbcMessage );
        stmt.setString( 4, user );
        stmt.setString( 5, table );
        stmt.executeUpdate();
        stmt.close();
    }
}

```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens ReportODBCError als das Skript für das Ereignis report\_odbc\_error, wenn die Skriptversion ver1 synchronisiert wird.

```

CALL ml_add_dnet_connection_script(
    'ver1',
    'report_odbc_error',
    'TestScripts.Test.ReportODBCError' )

```

Im Folgenden sehen Sie die .NET-Beispielmethode ReportODBCError. Sie protokolliert unter Verwendung einer .NET-Methode den Fehler in einer Tabelle.

```

using System;
using iAnywhere.MobiLink.Script;
namespace TestScripts
{
    public class Test
    {
        DBConnectionContext _cc;

        public Test( DBConnectionContext cc )
        {
            _cc = cc;
        }

        public void ReportODBCError( ref int actionCode,
                                     string odbcState,
                                     string errorMessage,
                                     string user,
                                     string table )
        {
            DBConnection conn = _cc.GetConnection();

            DBCommand cmd = conn.CreateCommand();
            cmd.CommandText =
                "INSERT INTO sync_error( action_code, odbc_state, error_message, " +
                "user_name, table_name ) VALUES ( ?, ?, ?, ?, ? )";
            cmd.Parameters[0] = actionCode;
            cmd.Parameters[1] = odbcState;
            cmd.Parameters[2] = errorMessage;
            cmd.Parameters[3] = user;
            cmd.Parameters[4] = table;
            cmd.ExecuteNonQuery();
        }
    }
}

```

## resolve\_conflict (Tabellenereignis)

Definiert einen Prozess zum Lösen eines Konflikts in einer bestimmten Tabelle.

## Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 1                                 |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                                                                                                                                                               | 2                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

## Standardaktion

Keine.

## Bemerkungen

Wenn eine Zeile in der entfernten Datenbank aktualisiert wird, speichert der MobiLink-Client eine Kopie der ursprünglichen Werte. Der Client sendet sowohl die alten als auch die neuen Werte an den MobiLink-Server.

Wenn der MobiLink-Server eine aktualisierte Zeile empfängt, vergleicht er die ursprünglichen Werte mit den vorhandenen Werten in der konsolidierten Datenbank. Der Vergleich wird mithilfe des Skripts upload\_fetch durchgeführt.

Wenn die alten gesendeten Werte mit den aktuellen Werten in der konsolidierten Datenbank nicht übereinstimmen, verursacht die Zeile einen Konflikt. Anstatt die Zeile zu aktualisieren, fügt der MobiLink-Server den alten und den neuen Wert in die konsolidierte Datenbank ein. Die alte und die neue Zeile werden mit den Skripten `upload_old_row_insert` bzw. `upload_new_row_insert` verarbeitet.

Nachdem die Werte eingefügt wurden, führt der MobiLink-Server das Skript `resolve_conflict` aus. Mit diesem Skript kann der Konflikt gelöst werden. Sie können ein Schema Ihrer Wahl implementieren.

Dieses Skript wird einmal pro Konflikt ausgeführt.

Anstatt das Skript `resolve_conflict` zu definieren können Sie Konflikte in einer gruppenorientierten Weise lösen, indem Sie die Konfliktlösungslogik in Ihr `end_upload_rows`- oder Ihr `end_upload`-Tabellenskript aufnehmen.

Sie können für jede Tabelle in der entfernten Datenbank ein `resolve_conflict`-Skript benutzen.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „`upload_old_row_insert` (Tabellenereignis)“ auf Seite 445
- „`upload_new_row_insert` (Tabellenereignis)“ auf Seite 443
- „`upload_update` (Tabellenereignis)“ auf Seite 458
- „`end_upload_rows` (Tabellenereignis)“ auf Seite 366
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

### SQL-Beispiel

Die folgende Anweisung definiert ein `resolve_conflict`-Skript, das für die CustDB-Beispielanwendung für eine Oracle-Installation geeignet ist. Es ruft die gespeicherte Prozedur `ULResolveOrderConflict` auf.

```
exec ml_add_table_script(
    'custdb', 'ULOrder', 'resolve_conflict',
    'begin ULResolveOrderConflict();
end; ')
CREATE OR REPLACE PROCEDURE ULResolveOrderConflict()
AS
    new_order_id integer;
    new_status   varchar(20);
    new_notes    varchar(50);
BEGIN
    -- approval overrides denial
    SELECT order_id, status, notes
    INTO new_order_id, new_status, new_notes
    FROM ULNewOrder
    WHERE syncuser_id = SyncUserID;
    IF new_status = 'Approved' THEN
        UPDATE ULOrder o
        SET o.status = new_status, o.notes =
            new_notes
        WHERE o.order_id = new_order_id;
    END IF;
    DELETE FROM ULOldOrder;
    DELETE FROM ULNewOrder;
END;
```

## Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `resolveConflict` als das Skript für das Tabellenereignis `resolve_conflict`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_table_script(  
  'ver1',  
  'table1',  
  'resolve_conflict',  
  'ExamplePackage.ExampleClass.resolveConflict' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `resolveConflict`. Sie ruft eine Java-Methode auf, die den Konflikt mit der JDBC-Verbindung löst, die von MobiLink bereitgestellt wird.

```
package ExamplePackage;  
import java.sql.*;  
import ianywhere.ml.script.*;  
public class ExampleClass  
{  
    DBConnectionContext _cc;  
  
    public ExampleClass( DBConnectionContext cc )  
    {  
        _cc = cc;  
    }  
  
    public void resolveConflict( String user,  
                               String table )  
    throws java.sql.SQLException  
    {  
        if( table == "Order" ) {  
            // Insert error information in a table,  
            Connection conn = _cc.getConnection();  
            String conflictTable = "New" + table;  
            PreparedStatement stmt = conn.prepareStatement(  
                "SELECT order_id, new_status, new_notes " +  
                "FROM " + conflictTable +  
                "WHERE rid = " + _cc.getRemoteID() );  
            ResultSet rs = stmt.executeQuery();  
            PreparedStatement updt = conn.prepareStatement(  
                "UPDATE ULOrder SET status = ?, notes = ? WHERE order_id = ?" );  
  
            while( rs.next() ) {  
                if( rs.getString( 2 ) == "Approved" ) {  
                    updt.setString( 1, rs.getString( 2 ) );  
                    updt.setString( 2, rs.getString( 3 ) );  
                    updt.setInt( 3, rs.getInt( 1 ) );  
                }  
            }  
            updt.close();  
            stmt.close();  
        }  
    }  
}
```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `ResolveConflict` als das Skript für das Tabellenereignis `resolve_conflict`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_dnet_table_script(
    'ver1',
    'table1',
    'resolve_conflict',
    'TestScripts.Test.ResolveConflict' )
```

Im Folgenden sehen Sie die .NET-Beispielmethode ResolveConflict. Sie ruft eine .NET-Methode auf, die den Konflikt löst.

```
using System;
using iAnywhere.MobiLink.Script;
namespace TestScripts
{
    public class Test
    {
        DBConnectionContext _cc;

        public Test( DBConnectionContext cc )
        {
            _cc = cc;
        }

        public void ResolveConflict( string user,
                                    string table )
        {
            if( table == "Order" ) {
                // Insert error information in a table,
                DBConnection conn = _cc.GetConnection();
                String conflictTable = "New" + table;
                DBCommand cmd = conn.CreateCommand();
                cmd.CommandText =
                    "SELECT order_id, new_status, new_notes " +
                    "FROM " + conflictTable +
                    "WHERE rid = " + _cc.GetRemoteID();
                DBRowReader dr = cmd.ExecuteReader();
                DBCommand updt = conn.CreateCommand();
                updt.CommandText =
                    "UPDATE ULOrder SET status = ?, notes = ? WHERE order_id = ?";

                object[] row;
                while( (row = dr.NextRow() ) != null ) {
                    if( row[1].Equals( "Approved" ) ) {
                        updt.Parameters[0] = row[1];
                        updt.Parameters[1] = row[2];
                        updt.Parameters[2] = row[0];
                    }
                }
                updt.Close();
                cmd.Close();
                conn.Close();
            }
        }
    }
}
```

## synchronization\_statistics (Verbindungsereignis)

Verfolgt die Synchronisationsstatistik.

## Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                | 1                                 |
| s.warnings                     | INTEGER. Die Anzahl der während der Synchronisation ausgegebenen Warnungen.                                                             | 2                                 |
| s.errors                       | INTEGER. Anzahl der Fehler, die während der Synchronisation vorgekommen sind.                                                           | 3                                 |
| s.deadlocks                    | INTEGER. Gesamtzahl der Deadlocks in der konsolidierten Datenbank, die für die Synchronisation festgestellt wurden.                     | 4                                 |
| s.synchronized_tables          | INTEGER. Anzahl der Client-Tabellen, die an der Synchronisation beteiligt waren.                                                        | 5                                 |
| s.connection_retries           | INTEGER. Angabe, wie oft der MobiLink-Server versucht hat, eine Verbindung zur konsolidierten Datenbank herzustellen.                   | 6                                 |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

**Standardaktion**

Keine.

**Bemerkungen**

Mit dem Ereignis synchronization\_statistics können Sie für jeden beliebigen Benutzer und für jede beliebige Verbindung diverse statistische Daten über die aktuelle Synchronisation sammeln. Das Verbindungsskript synchronization\_statistics wird unmittelbar vor dem Festschreiben am Ende der letzten Synchronisationstransaktion aufgerufen.

**Hinweis**

Abhängig von der Befehlszeile werden nicht alle Warnungen protokolliert. Die Anzahl der an dieses Skript übergebenen Warnungen ist die Anzahl der Warnungen, die protokolliert würden, wenn keine Warnungen deaktiviert wären. Diese Zahl kann höher sein als die Anzahl der protokollierten Warnungen.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „download\_statistics (Verbindungsereignis)“ auf Seite 333
- „download\_statistics (Tabellenereignis)“ auf Seite 337
- „upload\_statistics (Verbindungsereignis)“ auf Seite 448
- „upload\_statistics (Tabellenereignis)“ auf Seite 452
- „synchronization\_statistics (Tabellenereignis)“ auf Seite 425
- „time\_statistics (Verbindungsereignis)“ auf Seite 428
- „time\_statistics (Tabellenereignis)“ auf Seite 432
- „MobiLink-Profiler“ auf Seite 192
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

**SQL-Beispiel**

Das folgende Beispiel fügt Synchronisationsstatistiken in die Tabelle sync\_con\_audit ein.

```
CALL ml_add_connection_script(
  'ver1',
  'synchronization_statistics',
```

```
'INSERT INTO sync_con_audit(  
    ml_user,  
    warnings,  
    errors,  
    deadlocks,  
    synchronized_tables,  
    connection_retries)  
VALUES (  
    {ml s.username},  
    {ml s.warnings},  
    {ml s.errors},  
    {ml s.deadlocks},  
    {ml s.synchronized_tables},  
    {ml s.connection_retries})' )
```

Nachdem statistische Daten in die Audit-Tabelle eingefügt wurden, können Sie sie zur Überwachung der Synchronisationen einsetzen und ggf. Optimierungen durchführen.

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `synchronizationStatisticsConnection` als das Skript für das Verbindungsereignis `synchronization_statistics`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_connection_script(  
    'ver1',  
    'synchronization_statistics',  
    'ExamplePackage.ExampleClass.synchronizationStatisticsConnection'  
)
```

Im Folgenden sehen Sie die Java-Beispielmethode `synchronizationStatisticsConnection`. Sie protokolliert einige der Statistiken im MobiLink-Nachrichtenlog. (Das Protokollieren von Statistiken im MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
package ExamplePackage;  
public class ExampleClass {  
    String _curUser = null;  
    public void synchronizationStatisticsConnection(  
        String user,  
        int warnings,  
        int errors,  
        int deadlocks,  
        int synchronizedTables,  
        int connectionRetries ) {  
        java.lang.System.out.println(  
            "synch statistics number of deadlocks: "  
            + deadlocks );  
    }  
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `SyncStats` als das Skript für das Verbindungsereignis `synchronization_statistics`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_dnet_connection_script(  
    'ver1',  
    'synchronization_statistics',
```

```

    'TestScripts.Test.SyncStats'
)

```

Im Folgenden sehen Sie die .NET-Beispielmethode SyncStats. Sie protokolliert einige der Statistiken im MobiLink-Nachrichtenlog. (Das Protokollieren von Statistiken im MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```

namespace TestScripts
{
    public class Test
    {
        public void SyncStats( string    user,
                               int       warnings,
                               int       errors,
                               int       deadLocks,
                               int       syncedTables,
                               int       connRetries )
        {
            System.Console.WriteLine( "synch statistics number of deadlocks: " +
deadLocks );
        }
    }
}

```

## synchronization\_statistics (Tabellenereignis)

Ermöglicht den Zugriff auf statistische Daten der Synchronisation.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                   |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für<br>SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 1                                    |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                                                                                                                                                               | 2                                    |
| s.warnings                     | INTEGER. Gesamtzahl der Warnungen, die während der Synchronisation für die Tabelle ausgegeben wurden.                                                                                                                                                                        | 3                                    |
| s.errors                       | INTEGER. Gesamtzahl der Fehler, die während der Synchronisation für die Tabelle auftraten.                                                                                                                                                                                   | 4                                    |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                      |

### Standardaktion

Keine.

### Bemerkungen

Mit dem Ereignis `synchronization_statistics` können Sie für jeden beliebigen Benutzer und für jede beliebige Tabelle die Anzahl der Warnungen und Fehlermeldungen erfassen, die während der aktuellen Synchronisation aufgetreten sind. Das Tabellenskript `synchronization_statistics` wird unmittelbar vor dem Festschreiben am Ende der letzten Synchronisationstransaktion aufgerufen.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „download\_statistics (Verbindungsereignis)“ auf Seite 333
- „download\_statistics (Tabellenereignis)“ auf Seite 337
- „upload\_statistics (Verbindungsereignis)“ auf Seite 448
- „upload\_statistics (Tabellenereignis)“ auf Seite 452
- „synchronization\_statistics (Verbindungsereignis)“ auf Seite 421
- „time\_statistics (Verbindungsereignis)“ auf Seite 428
- „time\_statistics (Tabellenereignis)“ auf Seite 432
- „MobiLink-Profiler“ auf Seite 192
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

**SQL-Beispiel**

Das folgende Beispiel fügt Synchronisationsstatistiken in die Tabelle sync\_tab\_audit ein.

```
CALL ml_add_table_script(
  'ver1',
  'table1',
  'upload_insert',
  'INSERT INTO sync_tab_audit (
    ml_user,
    table,
    warnings,
    errors)
VALUES (
  {ml s.username},
  {ml s.table},
  {ml s.warnings},
  {ml s.errors} ) ' )
```

Nachdem statistische Daten der Synchronisation in die Audit-Tabelle eingefügt wurden, können Sie sie zur Überwachung der Synchronisationen einsetzen und ggf. Optimierungen durchführen.

**Java-Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens synchronizationStatisticsTable als das Skript für das Tabellenereignis synchronization\_statistics, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_table_script(
  'ver1',
  'table1',
  'synchronization_statistics',
  'ExamplePackage.ExampleClass.synchronizationStatisticsTable'
)
```

Im Folgenden sehen Sie die Java-Beispielmethode synchronizationStatisticsTable. Sie protokolliert einige der Statistiken im MobiLink-Nachrichtenlog. (Das Protokollieren von Statistiken im MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
package ExamplePackage;
public class ExampleClass {
    String _curUser = null;
```

```
public void synchronizationStatisticsTable(
    String user,
    String table,
    int warnings,
    int errors ) {
    java.lang.System.out.println( "synch statistics for table: "
    + table + " errors: " + errors );
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens SyncTableStats als das Skript für das Tabellenereignis synchronization\_statistics, wenn die Skriptversion ver1 und die Tabelle table1 synchronisiert werden.

```
CALL ml_add_dnet_table_script(
    'ver1',
    'table1',
    'synchronization_statistics',
    'TestScripts.Test.SyncTableStats'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode SyncTableStats. Sie protokolliert einige der Statistiken im MobiLink-Nachrichtenlog. (Das Protokollieren von Statistiken im MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
namespace TestScripts {
public class Test {
    string _curUser = null;
public void SyncTableStats(
    string user,
    string table,
    int warnings,
    int errors ) {
    System.Console.WriteLine( "synch statistics for table: "
    + table + " errors: " + errors );
}}}
```

## time\_statistics (Verbindungsereignis)

Verfolgt die statistischen Zeitdaten nach Benutzern und Ereignissen.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 1                                 |
| s.event_name                   | VARCHAR(128). Das Ereignis, dessen Statistik protokolliert wird.                                                                                                                                                                                                             | 2                                 |
| s.number_of_calls              | INTEGER. Gibt an, wie häufig das Skript aufgerufen wurde.                                                                                                                                                                                                                    | 3                                 |
| s.minimum_time                 | INTEGER. ms. Die kürzeste Zeit, die zum Ausführen eines Skripts während dieser Synchronisation benötigt wird.                                                                                                                                                                | 4                                 |
| s.maximum_time                 | INTEGER. ms. Die längste Zeit, die zum Ausführen eines Skripts während dieser Synchronisation benötigt wird.                                                                                                                                                                 | 5                                 |
| s.total_time                   | INTEGER. ms. Die Gesamtzeit, die zum Ausführen aller Skripten während der Synchronisation benötigt wird. (Dies ist nicht identisch mit der Länge/Dauer der Synchronisation.)                                                                                                 | 6                                 |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |

**Standardaktion**

Keine.

### Bemerkungen

Mit dem Ereignis `time_statistics` können Sie statistische Zeitdaten für eine Synchronisation sammeln. Die statistischen Daten werden nur für die Ereignisse gesammelt, für die es entsprechende Skripten gibt. Mit dem Skript werden in Fällen, in denen ein Einzelereignis mehrmals eintritt, Aggregatdaten gesammelt.

### Siehe auch

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „`time_statistics` (Tabellenereignis)“ auf Seite 432
- „`download_statistics` (Verbindungsereignis)“ auf Seite 333
- „`download_statistics` (Tabellenereignis)“ auf Seite 337
- „`upload_statistics` (Verbindungsereignis)“ auf Seite 448
- „`upload_statistics` (Tabellenereignis)“ auf Seite 452
- „`synchronization_statistics` (Verbindungsereignis)“ auf Seite 421
- „`synchronization_statistics` (Tabellenereignis)“ auf Seite 425
- „MobiLink-Profiler“ auf Seite 192
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

### SQL-Beispiel

Das folgende Beispiel fügt statistische Informationen in die Tabelle `time_statistics` ein.

```
CALL ml_add_connection_script(  
  'ver1',  
  'time_statistics',  
  'INSERT INTO time_statistics (  
    id,  
    ml_user,  
    event_name,  
    number_of_calls,  
    minimum_time,  
    maximum_time,  
    total_time)  
VALUES (  
  ts_id.nextval,  
  {ml s.username},  
  {ml s.event_name},  
  {ml s.number_of_calls},  
  {ml s.minimum_time},  
  {ml s.maximum_time},  
  {ml s.total_time} ) ' )
```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `timeStatisticsConnection` als das Skript für das Verbindungsereignis `time_statistics`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_connection_script(  
  'ver1',  
  'time_statistics',  
  'ExamplePackage.ExampleClass.timeStatisticsConnection' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `timeStatisticsConnection`. Sie gibt Statistiken für das Ereignis `prepare_for_download` aus. (Das Ausgeben von Statistiken im MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
package ExamplePackage;
public class ExampleClass
{
    public void timeStatisticsConnection(
        String      username,
        String      eventName,
        int          numberOfCalls,
        int          minimumTime,
        int          maximumTime,
        int          totalTime )
    {
        if( eventName.equals( "prepare_for_download" ) ) {
            System.out.println( "prepare_for_download num_calls: " +
                numberOfCalls +
                " total_time: " + totalTime );
        }
    }
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `TimeStats` als das Skript für das Verbindungsereignis `time_statistics`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
    'ver1',
    'time_statistics',
    'TestScripts.Test.TimeStats'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode `TimeStats`. Sie gibt Statistiken für das Ereignis `prepare_for_download` aus. (Das Ausgeben von Statistiken im MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
namespace TestScripts
{
    public class test
    {
        public void TimeStats( string      user,
                               string      eventName,
                               int          numberOfCalls,
                               int          minimumTime,
                               int          maximumTime,
                               int          totTime )
        {
            if( eventName == "prepare_for_download" ) {
                System.Console.WriteLine( "prepare_for_download num_calls: " +
                    numberOfCalls +
                    "total_time: " + totTime );
            }
        }
    }
}
```

## time\_statistics (Tabellenereignis)

Verfolgt die Zeitstatistik.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                | 1                                 |
| s.table                        | VARCHAR(128). Der Tabellenname.                                                                                                         | 2                                 |
| s.event_name                   | VARCHAR(128). Das Ereignis, dessen Statistik protokolliert wird.                                                                        | 3                                 |
| s.number_of_calls              | INTEGER. Gibt an, wie häufig das Skript aufgerufen wurde.                                                                               | 4                                 |
| s.minimum_time                 | INTEGER. ms. Die kürzeste Zeit, die zum Ausführen eines Skripts während der Synchronisation dieser Tabelle benötigt wird.               | 5                                 |
| s.maximum_time                 | INTEGER. ms. Die längste Zeit, die zum Ausführen eines Skripts während der Synchronisation dieser Tabelle benötigt wird.                | 6                                 |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für<br>SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| s.total_time                   | INTEGER. ms. Die Gesamtzeit, die zum Ausführen aller Skripten während der Synchronisation der Tabelle benötigt wird. (Dies ist nicht identisch mit der Länge/ Dauer der Synchronisation.)                                                                                    | 7                                    |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                      |

**Standardaktion**

Keine.

**Bemerkungen**

Mit dem Ereignis time\_statistics können Sie während der Synchronisation statistische Zeitdaten für eine Tabelle sammeln. Die statistischen Daten werden nur für die Ereignisse gesammelt, für die es entsprechende Skripten gibt. Mit dem Skript werden in Fällen, in denen ein Einzelereignis mehrmals eintritt, Aggregatdaten gesammelt.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „time\_statistics (Verbindungsereignis)“ auf Seite 428
- „download\_statistics (Verbindungsereignis)“ auf Seite 333
- „download\_statistics (Tabellenereignis)“ auf Seite 337
- „upload\_statistics (Verbindungsereignis)“ auf Seite 448
- „upload\_statistics (Tabellenereignis)“ auf Seite 452
- „synchronization\_statistics (Verbindungsereignis)“ auf Seite 421
- „synchronization\_statistics (Tabellenereignis)“ auf Seite 425
- „MobiLink-Profiler“ auf Seite 192
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

**SQL-Beispiel**

Das folgende Beispiel fügt statistische Informationen in die Tabelle time\_statistics ein.

```
CALL ml_add_table_script (
  'ver1',
  'table1',
  'time_statistics',
  'INSERT INTO time_statistics(
    ml_user,
    table,
    event_name,
    number_of_calls,
    minimum_time,
    maximum_time,
    total_time)
VALUES (
  {ml s.username},
  {ml s.table},
  {ml s.event_name},
  {ml s.number_of_calls},
  {ml s.minimum_time},
  {ml s.maximum_time},
  {ml s.total_time} )' );
```

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `timeStatisticsTable` als das Skript für das Tabellenereignis `time_statistics`, wenn die Skriptversion `ver1` synchronisiert wird.

```
CALL ml_add_java_table_script(
  'ver1',
  'table1',
  'time_statistics',
  'ExamplePackage.ExampleClass.timeStatisticsTable' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `timeStatisticsTable`. Sie gibt Statistiken für das Ereignis `upload_old_row_insert` aus.

```
public void timeStatisticsTable(
  String username,
  String tableName,
  String eventName,
  int numberOfCalls,
  int minimumTime,
  int maximumTime,
  int totalTime ) {
  if( eventName.equals( "upload_old_row_insert" ) ) {
    java.lang.System.out.println(
      "upload_old_row_insert num_calls: " + numCalls +
      "total_time: " + totalTime );
  }
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `TimeTableStats` als das Skript für das Tabellenereignis `time_statistics`, wenn die Skriptversion `ver1` und die Tabelle `table1` synchronisiert werden.

```
CALL ml_add_dnet_table_script(
  'ver1',
  'table1',
  'time_statistics',
```

```

    'TestScripts.Test.TimeTableStats'
)

```

Im Folgenden sehen Sie die .NET-Beispielmethode TimeTableStats. Sie gibt Statistiken für das Ereignis upload\_old\_row\_insert aus.

```

public void TimeTableStats(
    string user,
    string table,
    string eventName,
    int numberOfCalls,
    int minimumTime,
    int maximumTime,
    int totTime ) {
    if( event_name == "upload_old_row_insert" ) {
        System.Console.WriteLine(
            "upload_old_row_insert num_calls: " + num_calls +
            "total_time: " + total_time );
    }
}

```

## upload\_delete (Tabellenereignis)

Ein Datenskript, das ein Ereignis liefert, das der MobiLink-Server während der Verarbeitung des Uploads zur Verarbeitung der in der entfernten Datenbank gelöschten Zeilen verwendet.

### Parameter

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                            | Reihenfolge (veraltet für SQL) |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| <i>s.remote_id</i>             | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden. | k.A.                           |
| <i>s.username</i>              | VARCHAR(128). Der MobiLink-Benutzername. Dieser Parameter ist optional.                                                                 | Nicht anwendbar                |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| <i>s.Skriptversion</i>         | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |
| <i>r.pk-column-1</i>           | Erforderlich. Der erste gelöschte Primärschlüssel-Spaltenwert, referenziert nach Spaltenname oder Spaltennummer.                                                                                                                                                             | 1                                 |
| ...                            | ...                                                                                                                                                                                                                                                                          | ...                               |
| <i>r.pk-column-N</i>           | Erforderlich. Der letzte gelöschte Primärschlüssel-Spaltenwert, referenziert nach Spaltenname oder Spaltennummer.                                                                                                                                                            | N                                 |

### Standardaktion

Keine.

### Bemerkungen

Die in der konsolidierten Datenbank ausgeführte Aktion kann eine DELETE-Anweisung sein, muss aber nicht.

Für jede Tabelle in der entfernten Datenbank kann ein upload\_delete-Skript vorhanden sein.

Dieses Skript muss in SQL implementiert werden. Hinweise zur Java- oder .NET-Verarbeitung von Zeilen finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

#### Hinweis

Die Konflikterkennung erfolgt in der Regel viel schneller, wenn sie im Skript upload\_update auf einmal ausgeführt wird. Siehe „[upload\\_update \(Tabellenereignis\)](#)“ auf Seite 458.

### Siehe auch

- „[Datenskripten](#)“ auf Seite 283
- „[Skriptparameter](#)“ auf Seite 236
- „[Hinzufügungen und Löschungen von Skripten](#)“ auf Seite 258
- „[upload\\_insert \(Tabellenereignis\)](#)“ auf Seite 441
- „[upload\\_update \(Tabellenereignis\)](#)“ auf Seite 458

## SQL-Beispiel

Dieses Beispiel stammt aus dem Beispiel Contact und befindet sich in *Samples\MobiLink\Contact\build\_consol.sql*. Damit werden Kunden, die aus der entfernten Datenbank gelöscht wurden, als inaktiv markiert:

```
CALL ml_add_table_script(
  'ver1',
  'Customer',
  'upload_delete',
  'UPDATE Customer
   SET active = 0
   WHERE cust_id={ml r.cust_id}' )
```

## upload\_fetch (Tabellenereignis)

Ein Datenskript, das Zeilen aus einer synchronisierten Tabelle in der konsolidierten Datenbank für die Konflikterkennung auf Zeilenebene abruft.

### Parameter

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| <i>s.remote_id</i>             | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                |
| <i>s.username</i>              | VARCHAR(128). Der MobiLink-Benutzername. Dieser Parameter ist optional.                                                                                                                                                                                                      | Optional                       |
| <i>s.Skriptversion</i>         | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                |

| Parametername für SQL-Skripten | Beschreibung                                                                                            | Reihenfolge<br>(veraltet für SQL)                 |
|--------------------------------|---------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| <b>r.primary-key-1</b>         | Erforderlich. Der erste Primärschlüssel-Spaltenwert, referenziert nach Spaltenname oder Spaltennummer.  | 1 (2, falls der Benutzername referenziert wird)   |
| <b>r.primary-key-2</b>         | Erforderlich. Der zweite Primärschlüssel-Spaltenwert, referenziert nach Spaltenname oder Spaltennummer. | 2                                                 |
| ...                            | ...                                                                                                     | ...                                               |
| <b>r.primary-key-N</b>         | Erforderlich. Der letzte Primärschlüssel-Spaltenwert, referenziert nach Spaltenname oder Spaltennummer. | N (N+1, falls der Benutzername referenziert wird) |

### Standardaktion

Keine.

### Bemerkungen

Das Skript upload\_fetch ist ein Begleitskript des Ereignisses upload\_update.

Die Spalten der Ergebnismenge müssen mit Anzahl und Reihenfolge der für diese Tabelle aus der entfernten Datenbank hochgeladenen Spalten übereinstimmen. Wenn die zurückgegebenen Werte nicht mit dem Pre-Image in der hochgeladenen Zeile übereinstimmen, gilt dies als Konflikt.

Verwenden Sie keine READPAST-Tabellen-Hints in upload\_fetch-Skripten. Wenn das Skript eine gesperrte Zeile mit READPAST überspringt, nimmt die Synchronisationslogik an, dass die Zeile gelöscht wurde. Abhängig von den definierten Skripten wird entweder die hochgeladene Aktualisierung ignoriert oder die Konfliktlösung ausgelöst. Ein Ignorieren der Aktualisierung ist ein nicht akzeptables, gefährliches Verhalten. Das Auslösen einer Konfliktlösung stellt, je nach implementierter Lösungslogik, kein Problem dar.

Für jede Tabelle in der entfernten Datenbank kann ein upload-fetch- oder ein upload\_fetch\_column\_conflict-Skript vorhanden sein.

Dieses Skript muss in SQL implementiert werden. Hinweise zur Java- oder .NET-Verarbeitung von Zeilen finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

Dieses Skript wird eventuell ignoriert, wenn keines der folgenden Skripte definiert wurde: upload\_new\_row\_insert, upload\_old\_row\_insert und resolve\_conflict.

**Siehe auch**

- „Datenskripten“ auf Seite 283
- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „Konflikterkennung“ auf Seite 125
- „resolve\_conflict (Tabellenereignis)“ auf Seite 417
- „upload\_delete (Tabellenereignis)“ auf Seite 435
- „upload\_insert (Tabellenereignis)“ auf Seite 441
- „upload\_update (Tabellenereignis)“ auf Seite 458
- "READPAST mit MobiLink-Synchronisation verwenden" in „FROM-Klausel“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]

**SQL-Beispiel**

Das folgende SQL-Skript stammt aus dem Contact-Beispiel und befindet sich in %SQLANYSDAMP16%\MobiLink\Contact\build\_consol.sql. Hiermit werden Konflikte erkannt, die auftreten, wenn Zeilen übertragen werden, die in der Tabelle Product der entfernten Datenbank aktualisiert wurden. Mit diesem Skript werden Zeilen aus der Tabelle Product ausgewählt, jedoch abhängig vom Schema Ihrer konsolidierten und entfernten Datenbank können die beiden Tabellennamen unterschiedlich sein.

```
CALL ml_add_table_script(
  'ver1',
  'Product',
  'upload_fetch',
  'SELECT id, name, size, quantity, unit_price
   FROM Product
   WHERE id={ml r.id}' )
```

**upload\_fetch\_column\_conflict (Tabellenereignis)**

Ein Datenskript, das Zeilen aus einer synchronisierten Tabelle in der konsolidierten Datenbank für die Konflikterkennung auf Spaltenebene abrufen.

**Parameter**

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL)                 |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| <i>s.remote_id</i>             | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                                   |
| <i>s.username</i>              | VARCHAR(128). Der MobiLink-Benutzername. Dieser Parameter ist optional.                                                                                                                                                                                                      | Optional                                          |
| <i>s.Skriptversion</i>         | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                                   |
| <i>r.pk-column-1</i>           | Erforderlich. Der erste Primärschlüssel-Spaltenwert, referenziert nach Spaltenname oder Spaltennummer.                                                                                                                                                                       | 1 (2, falls der Benutzername referenziert wird)   |
| ...                            | ...                                                                                                                                                                                                                                                                          | ...                                               |
| <i>r.pk-column-N</i>           | Erforderlich. Der letzte Primärschlüssel-Spaltenwert, referenziert nach Spaltenname oder Spaltennummer.                                                                                                                                                                      | N (N+1, falls der Benutzername referenziert wird) |

**Standardaktion**

Keine.

**Bemerkungen**

Das Skript `upload_fetch_column_conflict` ist ein Begleitskript des Ereignisses `upload_update`.

Dieses Skript kann nur für entfernte Tabellen ohne BLOBs definiert werden.

Mit diesem Skript erkennt der MobiLink-Server nur dann einen Konflikt für eine Zeile, wenn dieselbe Spalte seit der letzten Synchronisation in der entfernten Datenbank und in der konsolidierten Datenbank aktualisiert wurde. Verschiedene Benutzer können dieselbe Zeile aktualisieren, ohne einen Konflikt zu generieren, sofern sie nicht dieselbe Spalte aktualisieren.

Mit dem Skript `upload_fetch_column_conflict` könnten Sie beispielsweise das Erkennen eines Konflikts vermeiden, wenn einer der entfernten Benutzer die Spalte `quant` der Tabelle `ULOrder` aktualisiert hat und

ein anderer entfernter Benutzer die Spalte notes derselben Zeile. Ein Konflikt wird nur erkannt, wenn beide Benutzer die Spalte quant aktualisiert hätten.

#### Hinweis

Die Konflikterkennung erfolgt in der Regel viel schneller, wenn sie im Skript upload\_update auf einmal ausgeführt wird. Siehe „[upload\\_update \(Tabellenereignis\)](#)“ auf Seite 458.

Wenn Sie ein Skript upload\_fetch\_column\_conflict verwenden und kein Konflikt erkannt wird, stammen die in Ihr Skript upload\_update übergebenen Zeilenwerte entweder aus dem Upload der entfernten Datenbank oder aus den aktuellen Werten der konsolidierten Datenbank gemäß Ihrem Skript upload\_fetch\_column\_conflict. Der Wert der entfernten Datenbank wird für Spalten verwendet, die in der entfernten Datenbank aktualisiert wurden. Andernfalls wird der aktuelle Wert der konsolidierten Datenbank verwendet. Mit anderen Worten: Nur die Spalten, die in der entfernten Datenbank aktualisiert wurden, werden in der konsolidierten Datenbank aktualisiert.

Für jede Tabelle in der entfernten Datenbank kann ein upload-fetch- oder ein upload\_fetch\_column\_conflict-Skript vorhanden sein.

Dieses Skript wird eventuell ignoriert, wenn keines der folgenden Skripte definiert wurde: upload\_new\_row\_insert, upload\_old\_row\_insert und resolve\_conflict.

Dieses Skript muss in SQL implementiert werden. Hinweise zur Java- oder .NET-Verarbeitung von Zeilen finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

#### Siehe auch

- „Datenskripten“ auf Seite 283
- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „Konflikterkennung“ auf Seite 125
- „upload\_fetch (Tabellenereignis)“ auf Seite 437
- „resolve\_conflict (Tabellenereignis)“ auf Seite 417
- „upload\_delete (Tabellenereignis)“ auf Seite 435
- „upload\_insert (Tabellenereignis)“ auf Seite 441
- „upload\_update (Tabellenereignis)“ auf Seite 458

## upload\_insert (Tabellenereignis)

Ein Datenskript, das ein Ereignis liefert, das der MobiLink-Server während der Verarbeitung des Uploads zur Verarbeitung der in die entfernte Datenbank eingefügten Zeilen verwendet.

#### Parameter

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn

Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| <i>s.remote_id</i>             | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| <i>s.username</i>              | VARCHAR(128). Der MobiLink-Benutzername. Dieser Parameter ist optional.                                                                                                                                                                                                      | Nicht anwendbar                   |
| <i>s.Skriptversion</i>         | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |
| <i>r.column-1</i>              | Erforderlich. Der erste eingefügte Spaltenwert, referenziert nach Spaltenname oder Spaltenwert.                                                                                                                                                                              | 1                                 |
| ...                            | ...                                                                                                                                                                                                                                                                          | ...                               |
| <i>r.Spalten-N</i>             | Erforderlich. Der letzte eingefügte Spaltenwert, referenziert nach Spaltenname oder Spaltenwert.                                                                                                                                                                             | N                                 |

### Standardaktion

Keine.

### Bemerkungen

Für jede Tabelle in der entfernten Datenbank kann ein upload\_insert-Skript vorhanden sein.

Dieses Skript muss in SQL implementiert werden. Hinweise zur Java- oder .NET-Verarbeitung von Zeilen finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

**Siehe auch**

- „Datenskripten“ auf Seite 283
- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „upload\_delete (Tabellenereignis)“ auf Seite 435
- „upload\_update (Tabellenereignis)“ auf Seite 458
- „upload\_fetch (Tabellenereignis)“ auf Seite 437

**SQL-Beispiel**

Dieses Beispiel verarbeitet Einfügungen in die Tabelle Customer in der entfernten Datenbank. Mit dem Skript werden die Werte in die Tabelle Customer in der konsolidierten Datenbank eingefügt. In der letzten Spalte wird der Kunde als aktiv gekennzeichnet. Die letzte Spalte erscheint nicht in der entfernten Datenbank.

```
CALL ml_add_table_script(
  'ver1',
  'Customer',
  'upload_insert',
  'INSERT INTO Customer(
    cust_id,
    name,
    rep_id,
    active )
VALUES (
  {ml r.cust_id},
  {ml r.name},
  {ml r.rep_id},
  1 )' );
```

**upload\_new\_row\_insert (Tabellenereignis)**

Konfliktlösungsskripten für anweisungsbasierte Uploads benötigen normalerweise den Zugriff auf alte und neue Werte für Zeilen, die aus der entfernten Datenbank hochgeladen wurden. Mit diesem Datenskriptereignis können Sie die neuen, aktualisierten Werte von aus der entfernten Datenbank hochgeladenen Zeilen verarbeiten.

**Parameter**

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL)                    |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername. Dieser Parameter ist optional.                                                                                                                                                                                                      | Optional (1, falls referenziert)                  |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                                   |
| <b>r.column-1</b>              | Erforderlich. Der erste Spaltenwert aus der neuen Zeile (im Post-Image), referenziert nach Spaltenname oder Spaltennummer.                                                                                                                                                   | 1 (2, falls der Benutzername referenziert wird)   |
| ...                            | ...                                                                                                                                                                                                                                                                          | ...                                               |
| <b>r.Spalten-N</b>             | Erforderlich. Der letzte Spaltenwert aus der neuen Zeile (im Post-Image), referenziert nach Spaltenname oder Spaltennummer.                                                                                                                                                  | N (N+1, falls der Benutzername referenziert wird) |

### Standardaktion

Keine.

### Bemerkungen

Sendet ein MobiLink-Client eine aktualisierte Zeile an den MobiLink-Server, so enthält diese Zeile nicht nur die neuen Werte (das Post-Image), sondern auch eine Kopie der Werte der alten Zeilen (das Pre-Image). Wenn das Pre-Image nicht mit den aktuellen Werten in der konsolidierten Datenbank übereinstimmt, wird ein Konflikt festgestellt.

Nachdem MobiLink einen Konflikt erkannt hat, können Sie mit diesem Ereignis Post-Image-Werte in einer Tabelle speichern. Dieses Ereignis kann Sie beim Entwickeln von Konfliktlösungsprozeduren für Aktualisierungen unterstützen. Die Parameter für dieses Ereignis enthalten neue Werte für die Zeile in der entfernten Datenbank, bevor die Aktualisierung in der entsprechenden Tabelle der konsolidierten Datenbank durchgeführt wird. Dieses Ereignis wird auch im Modus "Erzwungener Konflikt" zum Einfügen von Zeilen verwendet. (Der Modus "Erzwungener Konflikt" ist veraltet.)

#### Hinweis

Die Konflikterkennung erfolgt in der Regel viel schneller, wenn sie im Skript `upload_update` auf einmal ausgeführt wird. Siehe „[upload\\_update \(Tabellenereignis\)](#)“ auf Seite 458.

Das Skript für dieses Ereignis ist gewöhnlich eine Einfügensanweisung, die die neue Zeile in eine temporäre Tabelle einfügt, damit ein resolve\_conflict-Skript darauf zugreifen kann.

Für jede Tabelle in der entfernten Datenbank kann ein upload\_new\_row\_insert-Skript vorhanden sein.

Dieses Skript muss in SQL implementiert werden. Hinweise zur Java- oder .NET-Verarbeitung von Zeilen finden Sie unter [„Direkte Zeilenbehandlung“ auf Seite 627](#).

### Siehe auch

- [„Datenskripten“ auf Seite 283](#)
- [„Skriptparameter“ auf Seite 236](#)
- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)
- [„Konfliktverarbeitung“ auf Seite 124](#)
- [„resolve\\_conflict \(Tabellenereignis\)“ auf Seite 417](#)
- [„upload\\_old\\_row\\_insert \(Tabellenereignis\)“ auf Seite 445](#)
- [„upload\\_update \(Tabellenereignis\)“ auf Seite 458](#)
- [„Entfernte IDs und MobiLink-Benutzernamen in Skripten“ \[\*MobiLink - Clientadministration\*\]](#)

### SQL-Beispiel

Dieses Beispiel verarbeitet Aktualisierungen in der Tabelle product in der entfernten Datenbank. Mit dem Skript wird der neue Wert der Zeile in eine globale temporäre Tabelle mit dem Namen product\_conflict eingefügt. In der letzten Spalte der Tabelle wird die Zeile als neue Zeile gekennzeichnet.

```
CALL ml_add_table_script(  
  'ver1',  
  'table1',  
  'upload_new_row_insert',  
  'INSERT INTO DBA.product_conflict(  
    id,  
    name,  
    size,  
    quantity,  
    unit_price,  
    row_type )  
VALUES(  
  {ml r.id},  
  {ml r.name},  
  {ml r.size},  
  {ml r.quantity},  
  {ml r.unit_price},  
  'New' )' )
```

## upload\_old\_row\_insert (Tabellenereignis)

Konfliktlösungsskripten für anweisungsbasierte Uploads benötigen normalerweise den Zugriff auf alte und neue Werte für Zeilen, die aus der entfernten Datenbank hochgeladen wurden. Mit diesem Datenskriptereignis können Sie die alten Werte von aus der entfernten Datenbank hochgeladenen Zeilen verarbeiten.

## Parameter

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge (veraltet für SQL)                    |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | k.A.                                              |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername. Dieser Parameter ist optional.                                                                                                                                                                                                      | Optional (1, falls referenziert)                  |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                                   |
| <b>r.column-1</b>              | Erforderlich. Der erste Spaltenwert aus der alten Zeile (im Pre-Image), referenziert nach Spaltenname oder Spaltennummer.                                                                                                                                                    | 1 (2, falls der Benutzername referenziert wird)   |
| ...                            | ...                                                                                                                                                                                                                                                                          | ...                                               |
| <b>r.Spalten-N</b>             | Erforderlich. Der letzte Spaltenwert aus der alten Zeile (im Pre-Image), referenziert nach Spaltenname oder Spaltennummer.                                                                                                                                                   | N (N+1, falls der Benutzername referenziert wird) |

## Standardaktion

Keine.

## Bemerkungen

Sendet ein MobiLink-Client eine aktualisierte Zeile an den MobiLink-Server, so enthält diese Zeile nicht nur die neuen Werte (das Post-Image), sondern auch eine Kopie der Werte der alten Zeilen (das Pre-Image). Wenn das Pre-Image nicht mit den aktuellen Werten in der konsolidierten Datenbank übereinstimmt, wird ein Konflikt festgestellt.

Nachdem MobiLink einen Konflikt erkannt hat, können Sie mit diesem Ereignis Pre-Image-Werte in einer Tabelle speichern. Dieses Ereignis kann Sie beim Entwickeln von Konfliktlösungsprozeduren

unterstützen. Die Parameter für dieses Ereignis enthalten alte Zeilenwerte aus der entfernten Datenbank, bevor die Aktualisierung in der entsprechenden Tabelle der konsolidierten Datenbank durchgeführt wird. Dieses Ereignis wird auch im Modus "Erzwungener Konflikt" zum Einfügen von Zeilen verwendet. (Der Modus "Erzwungener Konflikt" ist veraltet.)

#### Hinweis

Die Konflikterkennung erfolgt in der Regel viel schneller, wenn sie im Skript `upload_update` auf einmal ausgeführt wird. Siehe [„upload\\_update \(Tabellenereignis\)“ auf Seite 458](#).

Das Skript für dieses Ereignis ist gewöhnlich eine Einfügensanweisung, die die alte Zeile in eine temporäre Tabelle einfügt, damit ein `resolve_conflict`-Skript darauf zugreifen kann.

Für jede Tabelle in der entfernten Datenbank kann ein `upload_old_row_insert`-Skript vorhanden sein.

Dieses Skript muss in SQL implementiert werden. Hinweise zur Java- oder .NET-Verarbeitung von Zeilen finden Sie unter [„Direkte Zeilenbehandlung“ auf Seite 627](#).

#### Siehe auch

- [„Datenskripten“ auf Seite 283](#)
- [„Skriptparameter“ auf Seite 236](#)
- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)
- [„Konfliktverarbeitung“ auf Seite 124](#)
- [„resolve\\_conflict \(Tabellenereignis\)“ auf Seite 417](#)
- [„upload\\_new\\_row\\_insert \(Tabellenereignis\)“ auf Seite 443](#)
- [„upload\\_update \(Tabellenereignis\)“ auf Seite 458](#)
- [„Entfernte IDs und MobiLink-Benutzernamen in Skripten“ \[MobiLink - Clientadministration\]](#)

#### SQL-Beispiel

Dieses Beispiel verarbeitet Aktualisierungen in der Tabelle `product` in der entfernten Datenbank. Mit dem Skript wird der alte Wert der Zeile in eine globale temporäre Tabelle mit dem Namen `product_conflict` eingefügt. In der letzten Spalte der Tabelle wird die Zeile als alte Zeile gekennzeichnet.

```
CALL ml_add_table_script(
  'ver1',
  'table1',
  'upload_old_row_insert',
  'INSERT INTO DBA.product_conflict (
    id,
    name,
    size,
    quantity,
    unit_price,
    row_type )
VALUES (
  {ml r.id},
  {ml r.name},
  {ml r.size},
  {ml r.quantity},
  {ml r.unit_price},
  ''Old'' )' )
```

## upload\_statistics (Verbindungsereignis)

Ermöglicht den Zugriff auf Synchronisationsstatistiken für Upload-Vorgänge.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 1                                 |
| s.warnings                     | INTEGER. Anzahl der aufgetretenen Warnungen.                                                                                                                                                                                                                                 | 2                                 |
| s.errors                       | INTEGER. Anzahl der aufgetretenen Fehler.                                                                                                                                                                                                                                    | 3                                 |
| s.inserted_rows                | INTEGER. Anzahl der Zeilen, die erfolgreich in die konsolidierte Datenbank eingefügt wurden.                                                                                                                                                                                 | 4                                 |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                                                                                             | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.deleted_rows                 | INTEGER. Anzahl der Zeilen, die erfolgreich aus der konsolidierten Datenbank gelöscht wurden.                                                                                                                                                                                                                                                            | 5                                 |
| s.updated_rows                 | INTEGER. Anzahl der Zeilen, die erfolgreich in der konsolidierten Datenbank aktualisiert wurden.                                                                                                                                                                                                                                                         | 6                                 |
| s.conflicted_updates           | INTEGER. Die Anzahl der aktualisierten Zeilen, die einen Konflikt verursacht haben. Eine Zeile wird nur dann einbezogen, wenn für sie ein Konfliktlösungsskript erfolgreich aufgerufen wurde.                                                                                                                                                            | 9                                 |
| s.ignored_inserts              | INTEGER. Gesamtanzahl der Upload-Einfügungen, die ignoriert wurden. Die Zeilen wurden ignoriert, weil 1) kein upload_insert-Skript im Normalmodus vorhanden ist oder 2) beim Aufrufen des entsprechenden Skripts durch den MobiLink-Server Fehler aufgetreten sind und das Ereignis handle_error oder handle_odbc_error der Wert 1000 zurückgegeben hat. | 10                                |
| s.ignored_deletes              | INTEGER. Die Anzahl der Upload-Löschungszeilen, die zu Fehlern führten, während das Skript upload_delete aufgerufen wurde, wenn handle_error oder handle_odbc_error definiert ist und den Wert 1000 zurückgegeben hat oder wenn für die angegebene Tabelle kein upload_delete-Skript definiert ist.                                                      | 11                                |
| s.ignored_updates              | INTEGER. Anzahl der Upload-Aktualisierungszeilen, die einen Konflikt verursacht haben, aber kein Konfliktlösungsskript erfolgreich aufgerufen oder kein upload_update-Skript definiert ist.                                                                                                                                                              | 12                                |

| Parametername für SQL-Skripten | Beschreibung                                                                                                        | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.bytes                        | INTEGER. Verwendete Speichermenge im MobiLink-Server zum Speichern des Uploads.                                     | 13                                |
| s.deadlocks                    | INTEGER. Gesamtzahl der Deadlocks in der konsolidierten Datenbank, die für die Synchronisation festgestellt wurden. | 14                                |

**Standardaktion**

Keine.

**Bemerkungen**

Mit dem Ereignis upload\_statistics können Sie für jeden beliebigen Benutzer statistische Daten zu Uploadvorgängen sammeln. Das Verbindungsskript upload\_statistics wird unmittelbar vor dem Festschreiben am Ende der Upload-Transaktion aufgerufen.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „download\_statistics (Verbindungsereignis)“ auf Seite 333
- „download\_statistics (Tabellenereignis)“ auf Seite 337
- „upload\_statistics (Tabellenereignis)“ auf Seite 452
- „synchronization\_statistics (Verbindungsereignis)“ auf Seite 421
- „synchronization\_statistics (Tabellenereignis)“ auf Seite 425
- „time\_statistics (Verbindungsereignis)“ auf Seite 428
- „time\_statistics (Tabellenereignis)“ auf Seite 432
- „MobiLink-Profiler“ auf Seite 192
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

**SQL-Beispiel**

Das folgende Beispiel fügt Synchronisationsstatistiken für Uploadvorgänge in die Tabelle upload\_summary\_audit ein.

```
CALL ml_add_connection_script (  
  'ver1',  
  'upload_statistics',  
  'INSERT INTO upload_summary_audit (  
    ml_user,  
    warnings,  
    errors,  
    inserted_rows,  
    deleted_rows,  
    updated_rows,  
    conflicted_updates,  
    ignored_inserts,  
    ignored deletes,
```

```

        ignored_updates,
        bytes, deadlocks )
VALUES (
    {ml s.username},
    {ml s.warnings},
    {ml s.errors},
    {ml s.inserted_rows},
    {ml s.deleted_rows},
    {ml s.updated_rows},
    {ml s.conflicted_updates},
    {ml s.ignored_inserts},
    {ml s.ignored_deletes},
    {ml s.ignored_updates},
    {ml s.bytes},
    {ml s.deadlocks} ) ' )

```

Nachdem statistische Daten in die Audit-Tabelle eingefügt wurden, können Sie sie zur Überwachung der Synchronisationen einsetzen und ggf. Optimierungen durchführen.

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens `uploadStatisticsConnection` als das Skript für das Verbindungsereignis `upload_statistics`, wenn die Skriptversion `ver1` synchronisiert wird.

```

CALL ml_add_java_connection_script(
    'ver1',
    'upload_statistics',
    'ExamplePackage.ExampleClass.uploadStatisticsConnection' )

```

Im Folgenden sehen Sie die Java-Beispielmethode `uploadStatisticsConnection`. Sie protokolliert einige Statistiken im MobiLink-Nachrichtenlog. (Das Protokollieren von Statistiken im MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```

package ExamplePackage;
public class ExampleClass {
    String _curUser = null;
    public void uploadStatisticsConnection(
        String user,
        int warnings,
        int errors,
        int insertedRows,
        int deletedRows,
        int updatedRows,
        int conflictedInserts,
        int conflictedDeletes,
        int conflictedUpdates,
        int ignoredInserts,
        int ignoredDeletes,
        int ignoredUpdates,
        int bytes,
        int deadlocks ) {
        java.lang.System.out.println( "updated rows: " +
            updatedRows );
    }
}

```

## .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens UploadStats als das Skript für das Verbindungsereignis upload\_statistics, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_dnet_connection_script(
    'ver1',
    'upload_statistics',
    'TestScripts.Test.UploadStats'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode UploadStats. Sie protokolliert einige Statistiken im MobiLink-Nachrichtenlog. (Das Protokollieren von Statistiken im MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
namespace TestScripts {
public class Test {
    string _curUser = null;
public void UploadStats (
    string user,
    int warnings,
    int errors,
    int insertedRows,
    int deletedRows,
    int updatedRows,
    int conflictInserts,
    int conflictDeletes,
    int conflictUpdates,
    int ignoredInserts,
    int ignoredDeletes,
    int ignoredUpdates,
    int bytes,
    int deadlocks ) {
    System.Console.WriteLine( "updated rows: " +
        updatedRows );
    }
}
```

## upload\_statistics (Tabellenereignis)

Ermöglicht den Zugriff auf Synchronisationsstatistiken für Upload-Vorgänge einer bestimmten Tabelle.

### Parameter

In der folgenden Tabelle stellt die Beschreibung den SQL-Datentyp bereit. Wenn Sie Ihr Skript in Java oder .NET schreiben, müssen Sie den entsprechenden Datentyp verwenden. Siehe „[SQL-Java-Datentypen](#)“ auf Seite 466 und „[SQL-.NET-Datentypen](#)“ auf Seite 541.

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.remote_id                    | VARCHAR(128). Die entfernte ID von MobiLink. Sie können die entfernte ID nur dann referenzieren, wenn Sie benannte Parameter verwenden.                                                                                                                                      | Nicht anwendbar                   |
| s.script_version               | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                   |
| s.username                     | VARCHAR(128). Der MobiLink-Benutzername.                                                                                                                                                                                                                                     | 1                                 |
| s.table                        | VARCHAR(128). Der Tabellename.                                                                                                                                                                                                                                               | 2                                 |
| s.warnings                     | INTEGER. Die Anzahl der beim Upload der Tabelle ausgegebenen Warnungen.                                                                                                                                                                                                      | 3                                 |
| s.errors                       | INTEGER. Die Anzahl der Fehler, einschließlich der bearbeiteten Fehler, die beim Upload der Tabelle aufgetreten sind.                                                                                                                                                        | 4                                 |
| s.inserted_rows                | INTEGER. Anzahl der Zeilen, die erfolgreich in die konsolidierte Datenbank eingefügt wurden.                                                                                                                                                                                 | 5                                 |
| s.deleted_rows                 | INTEGER. Anzahl der Zeilen, die erfolgreich aus der konsolidierten Datenbank gelöscht wurden.                                                                                                                                                                                | 6                                 |
| s.updated_rows                 | INTEGER. Anzahl der Zeilen, die erfolgreich in der konsolidierten Datenbank aktualisiert wurden.                                                                                                                                                                             | 7                                 |

| Parametername für SQL-Skripten | Beschreibung                                                                                                                                                                                                                                                                                                                                             | Reihenfolge<br>(veraltet für SQL) |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| s.conflicted_updates           | INTEGER. Die Anzahl der aktualisierten Zeilen, die einen Konflikt verursacht haben. Eine Zeile wird nur dann einbezogen, wenn für sie ein Konfliktlösungsskript erfolgreich aufgerufen wurde.                                                                                                                                                            | 10                                |
| s.ignored_inserts              | INTEGER. Gesamtanzahl der Upload-Einfügungen, die ignoriert wurden. Die Zeilen wurden ignoriert, weil 1) kein upload_insert-Skript im Normalmodus vorhanden ist oder 2) beim Aufrufen des entsprechenden Skripts durch den MobiLink-Server Fehler aufgetreten sind und das Ereignis handle_error oder handle_odbc_error der Wert 1000 zurückgegeben hat. | 11                                |
| s.ignored_deletes              | INTEGER. Die Anzahl der Upload-Löschungszeilen, die zu Fehlern führten, während das Skript upload_delete aufgerufen wurde, wenn handle_error oder handle_odbc_error definiert ist und den Wert 1000 zurückgegeben hat oder wenn für die angegebene Tabelle kein upload_delete-Skript definiert ist.                                                      | 12                                |
| s.ignored_updates              | INTEGER. Anzahl der Upload-Aktualisierungszeilen, die einen Konflikt verursacht haben, aber kein Konfliktlösungsskript erfolgreich aufgerufen oder kein upload_update-Skript definiert ist.                                                                                                                                                              | 13                                |
| s.bytes                        | INTEGER. Verwendete Speichermenge im MobiLink-Server zum Speichern des Uploads.                                                                                                                                                                                                                                                                          | 14                                |
| s.deadlocks                    | INTEGER. Gesamtzahl der Deadlocks in der konsolidierten Datenbank, die für die Synchronisation festgestellt wurden.                                                                                                                                                                                                                                      | 15                                |

**Standardaktion**

Keine.

**Bemerkungen**

Mit dem Ereignis upload\_statistics können Sie für jeden beliebigen Benutzer wesentliche statistische Daten zu Synchronisationen im Zusammenhang mit der betreffenden Tabelle sammeln. Das Tabellenskript upload\_statistics wird unmittelbar vor dem Festschreiben am Ende der Upload-Transaktion aufgerufen.

**Hinweis**

Abhängig von der Befehlszeile werden nicht alle Warnungen protokolliert. Die Anzahl der an dieses Skript übergebenen Warnungen ist die Anzahl der Warnungen, die protokolliert würden, wenn keine Warnungen deaktiviert wären. Diese Zahl kann höher sein als die Anzahl der protokollierten Warnungen.

**Siehe auch**

- „Skriptparameter“ auf Seite 236
- „Hinzufügungen und Löschungen von Skripten“ auf Seite 258
- „download\_statistics (Verbindungsereignis)“ auf Seite 333
- „upload\_statistics (Verbindungsereignis)“ auf Seite 448
- „upload\_statistics (Tabellenereignis)“ auf Seite 452
- „synchronization\_statistics (Verbindungsereignis)“ auf Seite 421
- „synchronization\_statistics (Tabellenereignis)“ auf Seite 425
- „time\_statistics (Verbindungsereignis)“ auf Seite 428
- „time\_statistics (Tabellenereignis)“ auf Seite 432
- „MobiLink-Profiler“ auf Seite 192
- „Entfernte IDs und MobiLink-Benutzernamen in Skripten“ [*MobiLink - Clientadministration*]

**SQL-Beispiel**

Das folgende Beispiel fügt eine Zeile in eine Tabelle ein, um Upload-Statistiken aufzuzeichnen.

```
CALL ml_add_connection_script(
  'ver1',
  'upload_statistics',
  'INSERT INTO my_upload_statistics (
    user_name,
    table_name,
    num_warnings,
    num_errors,
    inserted_rows,
    deleted_rows,
    updated_rows,
    conflicted_updates,
    ignored_inserts,
    ignored_deletes,
    ignored_updates, bytes,
    deadlocks )
VALUES(
  {ml s.username},
  {ml s.table},
  {ml s.warnings},
  {ml s.errors},
  {ml s.inserted_rows},
```

```
{ml s.deleted_rows},  
{ml s.updated_rows},  
{ml s.conflicted_updates},  
{ml s.ignored_inserts},  
{ml s.ignored_deletes},  
{ml s.ignored_updates},  
{ml s.bytes},  
{ml s.deadlocks} )' )
```

Das folgende Beispiel ist für eine konsolidierte Oracle-Datenbank geeignet.

```
CALL ml_add_connection_script(  
  'ver1',  
  'upload_statistics',  
  'INSERT INTO upload_tables_audit (  
    id,  
    user_name,  
    table,  
    warnings,  
    errors,  
    inserted_rows,  
    deleted_rows,  
    updated_rows,  
    conflicted_updates,  
    ignored_inserts,  
    ignored_deletes,  
    ignored_updates,  
    bytes,  
    deadlocks )  
VALUES (  
  ut_audit.nextval,  
  {ml s.username},  
  {ml s.table},  
  {ml s.warnings},  
  {ml s.errors},  
  {ml s.inserted_rows},  
  {ml s.deleted_rows},  
  {ml s.updated_rows},  
  {ml s.conflicted_updates},  
  {ml s.ignored_inserts},  
  {ml s.ignored_deletes},  
  {ml s.ignored_updates},  
  {ml s.bytes},  
  {ml s.deadlocks} )' )
```

Nachdem statistische Daten in die Audit-Tabelle eingefügt wurden, können Sie sie zur Überwachung der Synchronisationen einsetzen und ggf. Optimierungen durchführen.

### Java-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens uploadStatisticsTable als das Skript für das Tabellenereignis upload\_statistics, wenn die Skriptversion ver1 synchronisiert wird.

```
CALL ml_add_java_table_script(  
  'ver1',  
  'table1',  
  'upload_statistics',  
  'ExamplePackage.ExampleClass.uploadStatisticsTable' )
```

Im Folgenden sehen Sie die Java-Beispielmethode `uploadStatisticsTable`. Sie protokolliert einige Statistiken im MobiLink-Nachrichtenlog. (Das Protokollieren von Statistiken im MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
package ExamplePackage;
public class ExampleClass {
    String _curUser = null;
    public void uploadStatisticsTable(
        String user,
        String table,
        int warnings,
        int errors,
        int insertedRows,
        int deletedRows,
        int updatedRows,
        int conflictedInserts,
        int conflictedDeletes,
        int conflictedUpdates,
        int ignoredInserts,
        int ignoredDeletes,
        int ignoredUpdates,
        int bytes,
        int deadlocks ) {
        java.lang.System.out.println( "updated rows:" +
            updatedRows );
    }
}
```

### .NET-Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine .NET-Methode namens `UploadTableStats` als das Skript für das Tabellenereignis `upload_statistics`, wenn die Skriptversion `ver1` und die Tabelle `table1` synchronisiert werden.

```
CALL ml_add_dnet_table_script(
    'ver1',
    'table1',
    'upload_statistics',
    'TestScripts.Test.UploadTableStats'
)
```

Im Folgenden sehen Sie die .NET-Beispielmethode `uploadStatisticsTable`. Sie protokolliert einige Statistiken im MobiLink-Nachrichtenlog. (Das Protokollieren von Statistiken im MobiLink-Meldungslog ist während der Entwicklung möglicherweise nützlich, verlangsamt aber einen Produktionsserver.)

```
namespace TestScripts {
public class Test {
    string _curUser = null;
    public void UploadTableStats(
        string user,
        string table,
        int warnings,
        int errors,
        int insertedRows,
        int deletedRows,
        int updatedRows,
        int conflictInserts,
        int conflictDeletes,
        int conflictUpdates,
        int ignoredInserts,
        int ignoredDeletes,
        int ignoredUpdates,

```

```
int bytes,
int deadlocks ) {
System.Console.WriteLine( "updated rows: " +
updatedRows );
}}}
```

upload\_update (Tabellenereignis)

Ein Datenskript, das ein Ereignis liefert, das der MobiLink-Server während der Verarbeitung des Uploads zur Verarbeitung der in der entfernten Datenbank aktualisierten Zeilen verwendet.

Parameter

In SQL-Skripten können Sie Ereignisparameter mit ihrem Namen oder mit einem Fragezeichen angeben. Die Verwendung von Fragezeichen ist veraltet und es wird empfohlen, benannte Parameter zu verwenden. Sie können nicht Namen und Fragezeichen in einem Skript mischen. Wenn Sie Fragezeichen verwenden, müssen die Parameter in der unten gezeigten Reihenfolge stehen und sind nur dann optional, wenn keine nachfolgenden Parameter angegeben wurden. (Sie müssen beispielsweise Parameter 1 verwenden, wenn Sie Parameter 2 verwenden möchten.) Wenn Sie benannte Parameter verwenden, können Sie beliebige Teilmengen der Parameter in beliebiger Reihenfolge angeben.

| Parameter              | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für<br>SQL) |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| <i>s.Skriptversion</i> | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                      |
| <i>r.column-1</i>      | Erforderlich. Der erste Nicht-Primärschlüssel-Spaltenwert aus dem neuen Spaltenwert (im Post-Image), referenziert nach Spaltenname oder Spaltennummer.                                                                                                                       | 1                                    |
| ...                    | ...                                                                                                                                                                                                                                                                          | ...                                  |
| <i>r.column-M</i>      | Erforderlich. Der letzte Nicht-Primärschlüssel-Spaltenwert aus dem neuen Spaltenwert (im Post-Image), referenziert nach Spaltenname oder Spaltennummer.                                                                                                                      | M                                    |
| <i>r.pk-column-1</i>   | Erforderlich. Der erste Primärschlüssel-Spaltenwert aus dem neuen Spaltenwert (im Post-Image), referenziert nach Spaltenname oder Spaltennummer.                                                                                                                             | M + 1                                |
| ...                    | ...                                                                                                                                                                                                                                                                          | ...                                  |

| Parameter            | Beschreibung                                                                                                                                                                                                                                                                 | Reihenfolge<br>(veraltet für<br>SQL) |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------|
| <b>r.pk-column-N</b> | Erforderlich. Der letzte Primärschlüssel-Spaltenwert aus dem neuen Spaltenwert (im Post-Image), referenziert nach Spaltenname oder Spaltennummer.                                                                                                                            | M + N                                |
| <b>o.column-N</b>    | Optional. Der erste Nicht-Primärschlüssel-Spaltenwert aus dem alten Spaltenwert (im Pre-Image), referenziert nach Spaltenname oder Spaltennummer.                                                                                                                            | M + N + 1                            |
| ...                  | ...                                                                                                                                                                                                                                                                          | ...                                  |
| <b>o.column-M</b>    | Optional. Der letzte Nicht-Primärschlüssel-Spaltenwert aus dem alten Spaltenwert (im Pre-Image), referenziert nach Spaltenname oder Spaltennummer.                                                                                                                           | M + N + M                            |
| s.script_version     | VARCHAR(128). Optionaler IN-Parameter, mit dem angegeben wird, dass der MobiLink-Server die bei der aktuellen Synchronisation verwendete Skriptversionszeichenfolge an diesen Parameter übergibt. Fragezeichen können nicht verwendet werden, um diesen Parameter anzugeben. | Nicht anwendbar                      |

### Standardaktion

Keine.

### Bemerkungen

Die WHERE-Klausel muss alle Primärschlüsselspalten enthalten, die synchronisiert werden, und kann optional auch die Nicht-Primärschlüssel-Spalten enthalten. Die SET-Klausel muss alle Nicht-Primärschlüsselspalten enthalten, die synchronisiert werden.

Sie können benannte Parameter in einer beliebigen Reihenfolge verwenden. Derselbe benannte Parameter kann in demselben Skript beliebig oft verwendet werden. In ein Skript mit benannten Parametern können Sie auch eine Teilmenge der Spalten angeben.

Das upload\_update-Skript für die Tabelle MyTable kann beispielsweise folgendermaßen geschrieben werden:

```
UPDATE MyTable
  SET column_2 = { ml r.column_2 }, column_1 = { ml r.column_1 }, ...,
  column_M = { ml r.column_M }
  WHERE pk_column_1 = { ml r.pk_column_1 } AND ... AND pk_column_N = { ml
  r.pk_column_N }
```

Für jede Tabelle in der entfernten Datenbank kann ein upload\_update-Skript vorhanden sein.

Dieses Skript muss in SQL implementiert werden. Hinweise zur Java- oder .NET-Verarbeitung von Zeilen finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

Um mit dem upload\_update-Skript Konflikte festzustellen, müssen Sie alle Nicht-Primärschlüsselspalten in die WHERE-Klausel einbeziehen.

```
UPDATE table-name
SET col1 = {ml r.col1}, col2 = {ml r. col2 } ...
WHERE pk1 = {ml r.pk1} AND pk2 = {ml r.pk2} ...
      AND col1 = {ml o.col1} AND col2 = {ml o.col2} ...
```

In dieser Anweisung sind col1 und col2 Nicht-Primärschlüsselspalten, pk1 und pk2 hingegen sind Primärschlüsselspalten. Die an die zweite Gruppe von Nicht-Primärschlüsselspalten übergebenen Werte sind das Pre-Image der aktualisierten Zeile. Die WHERE-Klausel vergleicht die alten Werte, die aus der entfernten Datenbank eingelesen wurden, mit den aktuellen Werten in der konsolidierten Datenbank. Wenn die Werte nicht übereinstimmen, wird die Aktualisierung ignoriert, sodass die bereits in der konsolidierten Datenbank enthaltenen Werte erhalten bleiben.

Dieses Skript muss in SQL implementiert werden. Hinweise zur Java- oder .NET-Verarbeitung von Zeilen finden Sie unter [„Direkte Zeilenbehandlung“ auf Seite 627](#).

### Siehe auch

- [„Datenskripten“ auf Seite 283](#)
- [„Skriptparameter“ auf Seite 236](#)
- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)
- [„Konflikterkennung mit upload\\_update-Skripten“ auf Seite 128](#)
- [„Konfliktlösung mit upload\\_update-Skripten“ auf Seite 130](#)
- [„upload\\_delete \(Tabellenereignis\)“ auf Seite 435](#)
- [„upload\\_fetch \(Tabellenereignis\)“ auf Seite 437](#)
- [„upload\\_insert \(Tabellenereignis\)“ auf Seite 441](#)

### SQL-Beispiel

Mit diesem Beispiel werden Aktualisierungen verarbeitet, die in der Tabelle Customer in der entfernten Datenbank vorgenommen wurden. Mit dem Skript werden die Werte in der Tabelle Customer in der konsolidierten Datenbank aktualisiert.

```
CALL ml_add_table_script(
    'ver1',
    'table1',
    'upload_update',
    'UPDATE Customer
     SET name = {ml r.name}, rep_id = {ml r.rep_id}
     WHERE cust_id = {ml o.cust_id}')
```

Im folgenden Beispiel wird eine ähnliche Aktualisierung ausgeführt. Dabei werden jedoch die alten Werte (aus dem Pre-Image) verwendet, um sicherzustellen, dass die Aktualisierung nur erfolgt, wenn es keinen Konflikt gibt. Wenn es einen Konflikt gibt, wird die Aktualisierung in dieser "Der Erste gewinnt"-Konfliktlösungsrichtlinie ignoriert.

```
CALL ml_add_table_script(
    'ver1',
    'table1',
    'upload_update',
    'UPDATE Customer
     SET name = {ml r.name}, rep_id = {ml r.rep_id}
     WHERE cust_id = {ml o.cust_id}
```

```
AND name = {ml o.name}  
AND rep_id = {ml o.rep_id}')
```



---

# MobiLink-Server-APIs

Dieser Abschnitt beschreibt die MobiLink-Server-APIs für Java und .NET.

## Schreiben eines Synchronisationsskripts in Java

Die Aktionen des MobiLink-Servers können mithilfe von Synchronisationsskripten gesteuert werden. Sie können diese Skripten in SQL, .NET oder Java implementieren. Die Java-Synchronisationslogik kann auf dieselbe Weise wie SQL-Logikfunktionen eingesetzt werden. Der MobiLink-Server kann beim Eintritt von MobiLink-Ereignissen Java-Methoden ebenso wie SQL-Skripten aufrufen. Eine Java-Methode kann eine SQL-Zeichenfolge an MobiLink zurückgeben.

Dieser Abschnitt erläutert, wie Sie Java-Synchronisationslogik einrichten, entwickeln und anwenden. Außerdem werden eine Beispielanwendung und die API für JAVA des MobiLink-Servers beschrieben.

### Siehe auch

- „Optionen zum Erstellen serverseitiger Synchronisationslogik“ [*MobiLink - Erste Orientierung*]
- „Schreiben von Synchronisationsskripten“ auf Seite 231

## Java-Synchronisationslogik einrichten

Verwenden Sie die folgende Prozedur, um Synchronisationsskripten in Java zu implementieren.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Kontext und Bemerkungen

Wenn Sie SQL Anywhere installieren, legt das Installationsprogramm automatisch den Speicherort der API für Java-Klassen von MobiLink fest. Wenn Sie den MobiLink-Server starten, schließt er diese Klassen automatisch in die classpath-Variable ein. Die MobiLink-Server-API für Java-Klassen befindet sich in %SQLANY16%\java\mlscript.jar.

### Aufgabe

1. Erstellen Sie Ihre Klassen. Schreiben Sie für jedes einzelne erforderliche Synchronisationsskript eine Methode. Diese Methoden müssen öffentlich sein. Die Klasse muss im Paket öffentlich sein.

Siehe „Methoden“ auf Seite 468.

Jede Klasse mit nichtstatischen Methoden muss über einen öffentlichen Konstruktor verfügen. Der MobiLink-Server instanziert automatisch jede einzelne Klasse, wenn eine Methode in der betreffenden Klasse zum ersten Mal aufgerufen wird.

Siehe „Konstruktoren“ auf Seite 467.

2. Beim Kompilieren der Klasse müssen Sie die JAR-Datei *java\mlscript.jar* einbeziehen.

Beispiel:

```
javac MyClass.java -classpath "C:\Program Files\SQL Anywhere 16\java\mlscript.jar"
```

3. In den MobiLink-Systemtabellen Ihrer konsolidierten Datenbank geben Sie für jedes einzelne Skript den Namen des aufzurufenden Pakets, der Klasse und der Methode an. Pro Skriptversion ist eine Klasse zulässig.

Sie können diese Angaben beispielsweise mit der gespeicherten Prozedur `ml_add_java_connection_script` oder `ml_add_java_table_script` in die MobiLink-Systemtabellen einfügen.

Wird z.B. die folgende SQL-Anweisung in einer SQL Anywhere-Datenbank ausgeführt, dann wird festgelegt, dass "MyPackage.MyClass.myMethod" für die Skriptversion `ver1` immer dann ausgeführt werden soll, wenn auf Verbindungsebene das Ereignis `authenticate_user` eintritt. Bei der festgelegten Methode muss es sich um den vollständigen qualifizierten Namen einer öffentlichen Java-Methode handeln. Der Name berücksichtigt die Groß- und Kleinschreibung.

```
call ml_add_java_connection_script('ver1',
'authenticate_user', 'myPackage.myClass.myMethod')
```

Weitere Hinweise zur Erstellung von Skripten finden Sie unter:

- [Systemprozeduren zum Hinzufügen oder Löschen von Skripten auf Seite 648](#)
- [„ml\\_add\\_java\\_connection\\_script-Systemprozedur“ auf Seite 655](#)
- [„ml\\_add\\_java\\_table\\_script-Systemprozedur“ auf Seite 656](#)

4. Weisen Sie den MobiLink-Server an, Klassen zu laden. Ein wesentlicher Bestandteil der Java-Synchronisationslogik besteht darin, der Java VM mitzuteilen, wo sich die Java-Klassen befinden. Dies kann auf zwei Wegen geschehen:

- Verwenden Sie die `mlsrv16`-Option `-sl java -cp`, um eine Reihe von Verzeichnissen oder Jar-Dateien festzulegen, in denen nach Klassen gesucht werden soll. Führen Sie zum Beispiel folgenden Befehl aus:

```
mlsrv16 -c "DSN=consolidated1" -sl java (-cp %classpath%;c:\local\Java\myclasses.jar)
```

Der MobiLink-Server fügt automatisch den Speicherort der API für Java-Klassen des MobiLink-Servers (*java\mlscript.jar*) an die Gruppe der Verzeichnisse oder JAR-Dateien an. Die Option `-sl java` erzwingt auch, dass die Java Virtual Machine beim Start des Servers geladen wird.

Weitere Hinweise zu den verfügbaren Java-Optionen finden Sie unter [„mlsrv16-Option -sl java“ auf Seite 79](#).

- Legen Sie den Klassenpfad explizit fest. Verwenden Sie eine Anweisung wie die folgende, um den Klassenpfad für benutzerdefinierte Klassen festzulegen:

```
SET classpath=%classpath%;c:\local\Java\myclasses.jar
```

Wenn Ihr Systemklassenpfad die Klassen Ihrer Java-Synchronisationslogik umfasst, müssen Sie in der Befehlszeile des MobiLink-Servers nichts ändern.

Mit der Java-Option `-sl` können Sie erzwingen, dass die Java VM beim Start des Servers geladen wird. Andernfalls wird die Java VM bei der Ausführung der ersten Java-Methode gestartet.

Weitere Hinweise zu den verfügbaren Java-Optionen finden Sie unter [„mlsrv16-Option -sl java“ auf Seite 79](#).

5. Wenn Sie unter Unix eine bestimmte JRE laden möchten, legen Sie `LD_LIBRARY_PATH` (`LIBPATH` unter IBM AIX, `SHLIB_PATH` unter HP-UX) so fest, dass das Verzeichnis mit der JRE einbezogen wird. Das Verzeichnis muss vor allen SQL Anywhere-Installationsverzeichnissen aufgeführt sein.

## Ergebnisse

Die Java-Synchronisationslogik wird eingerichtet.

### Siehe auch

- [„JavaSynchronisationslogik“ auf Seite 465](#)
- [„Java-Synchronisationsbeispiel“ auf Seite 473](#)
- [„Referenz der MobiLink-Server-API für Java“ auf Seite 477](#)
- [„Optionen zum Erstellen serverseitiger Synchronisationslogik“ \[\*MobiLink - Erste Orientierung\*\]](#)
- [„Schreiben von Synchronisationsskripten“ auf Seite 231](#)

## JavaSynchronisationslogik

Zum Schreiben von Java-Synchronisationslogik sind Kenntnisse über die MobiLink-Ereignisse, über Java sowie über die API für Java des MobiLink-Servers erforderlich.

Eine vollständige Beschreibung der API-Schnittstelle finden Sie unter [„Referenz der MobiLink-Server-API für Java“ auf Seite 477](#).

Java-Synchronisationslogik kann zur Aufrechterhaltung der Statusinformationen und zur Implementierung der Logik vor und nach Upload- und Download-Ereignissen eingesetzt werden. Ein in Java geschriebenes `begin_synchronization`-Skript könnte z.B. den MobiLink Benutzernamen in einer Variablen speichern. Später im Synchronisationsprozess aufgerufene Skripten können dann auf diese Variable zugreifen. Sie können auch mit Java auf Zeilen in der konsolidierten Datenbank zugreifen, und zwar bevor und nachdem sie festgeschrieben wurden.

Der Einsatz von Java verringert die Abhängigkeit von der konsolidierten Datenbank. Das Aufrüsten der konsolidierten Datenbank auf eine neuere Version oder der Wechsel auf ein anderes Datenbank-Managementsystem beeinträchtigt das Verhalten in geringerem Umfang.

### Direkte Zeilenbehandlung

Mit der direkten Zeilenbehandlung in MobiLink übermitteln Sie entfernte Daten an jede zentrale Datenquelle, Anwendung oder einen Webdienst. Die direkte Zeilenbehandlung verwendet spezielle Klassen in den APIs für Java oder .NET des MobiLink-Servers, um direkt auf synchronisierte Daten zuzugreifen.

Siehe [„Direkte Zeilenbehandlung“ auf Seite 627](#).

## Klasseninstanzen

Der MobiLink-Server instanziert Ihre Klassen auf der Verbindungsebene. Wenn ein Ereignis erreicht wurde, für das Sie eine nichtstatische Java-Methode geschrieben haben, erstellt der MobiLink-Server automatisch eine Instanz der Klasse, falls dies noch nicht bei der aktuellen Verbindung geschehen ist.

Siehe „[Konstruktoren](#)“ auf Seite 541.

Alle Methoden, die für eine Skriptversion einem Ereignis auf der Verbindungs- oder Tabellenebene direkt zugeordnet sind, **müssen zu derselben Klasse gehören**.

Nachdem eine Klasse instanziiert wurde, bleibt sie für jede einzelne Datenbankverbindung bestehen, bis die Verbindung geschlossen wird. Dieselbe Instanz kann so auch für mehrere aufeinander folgende Synchronisationssitzungen verwendet werden. Wenn die Informationen in öffentlichen oder privaten Variablen nicht explizit gelöscht werden, bleiben sie für alle Synchronisationen, die über dieselbe Verbindung ausgeführt werden, bestehen.

Sie können auch statische Klassen bzw. Variablen benutzen. In diesem Fall stehen die Werte in allen Verbindungen zur Verfügung.

Der MobiLink-Server löscht Ihre Klassen nur dann automatisch, wenn die Verbindung zur konsolidierten Datenbank geschlossen wird.

## Transaktionen

Die normalen Regeln im Hinblick auf Transaktionen beziehen sich auf Java-Methoden. Start und Dauer von Datenbanktransaktionen sind wesentlich für den Synchronisationsprozess. Transaktionen dürfen nur vom MobiLink-Server gestartet oder beendet werden. Wenn Sie bei einer Synchronisationsverbindung und innerhalb einer Java-Methode Transaktionen ausdrücklich festschreiben oder zurücksetzen, verletzen Sie die Integrität des Synchronisationsprozesses und es können Fehler verursacht werden.

Diese Regeln betreffen nur die vom MobiLink-Server hergestellten Datenbankverbindungen und insbesondere die von Methoden zurückgegebenen SQL-Anweisungen. Wenn Ihre Klassen andere Datenbankverbindungen erstellen, verwenden Sie vorhandene Verwaltungsregeln, um Klassen zu verwalten, die von anderen Datenbankverbindungen erstellt wurden.

## SQL-Java-Datentypen

Folgende Tabelle enthält SQL-Datentypen und die jeweils entsprechenden Java-Datentypen.

| SQL-Datentyp | Entsprechender Java-Datentyp |
|--------------|------------------------------|
| VARCHAR      | java.lang.String             |
| CHAR         | java.lang.String             |
| INTEGER      | int oder Integer             |

| SQL-Datentyp    | Entsprechender Java-Datentyp       |
|-----------------|------------------------------------|
| BINARY          | byte[ ]                            |
| BIGINT          | long                               |
| TIMESTAMP       | java.sql.Timestamp                 |
| INOUT INTEGER   | ianywhere.ml.script.InOutInteger   |
| INOUT VARCHAR   | ianywhere.ml.script.InOutString    |
| INOUT CHAR      | ianywhere.ml.script.InOutString    |
| INOUT BYTEARRAY | ianywhere.ml.script.InOutByteArray |
| INOUT TIMESTAMP | java.sql.Timestamp                 |

Der MobiLink-Server fügt das `ianywhere.ml.script`-Paket automatisch dem Klassenpfad hinzu, falls es noch nicht vorhanden ist. Wenn Sie Ihre Klasse kompilieren, müssen Sie jedoch den Pfad `%SQLANY16%\java\mlscript.jar` hinzufügen.

## Konstrukturen

Der Konstruktor Ihrer Klasse kann eine von zwei Signaturen aufweisen:

```
public MyScriptClass(ianywhere.ml.script.DBConnectionContext sc)
```

oder

```
public MyScriptClass()
```

Der an Sie weitergegebene Synchronisationskontext gilt für die Verbindung, über die der MobiLink-Server den aktuellen Benutzer synchronisiert.

Die Methode `DBConnectionContext.getConnection` gibt dieselbe Datenbankverbindung zurück, die MobiLink benutzt, um den aktuellen Benutzer zu synchronisieren. Sie können Anweisungen in dieser Verbindung ausführen, dürfen die Transaktion jedoch weder festschreiben noch zurücksetzen. Der MobiLink-Server verwaltet die Transaktionen.

Der MobiLink-Server zieht Konstrukturen mit der ersten Signatur vor. Er verwendet den Konstruktor ohne Argument nur, falls kein Konstruktor mit der ersten Signatur vorhanden ist.

### Siehe auch

- [DBConnectionContext-Schnittstelle \[MobiLink-Server Java\] auf Seite 477](#)

## Methoden

Im Allgemeinen implementieren Sie für jedes einzelne Synchronisationsereignis eine Methode. Diese Methoden müssen öffentlich sein. Wenn sie nicht öffentlich sind, kann der MobiLink-Server sie nicht benutzen und nicht feststellen, ob sie überhaupt vorhanden sind.

Die Namen der Methoden sind unerheblich, sofern sie mit den in der Tabelle `ml_script` in der konsolidierten Datenbank angegebenen Namen übereinstimmen. In den Beispielen in der Dokumentation sind die Methodennamen jedoch dieselben wie in den MobiLink-Ereignissen, weil der Java-Code durch diese Benennungskonvention leichter lesbar wird.

Die Signatur Ihrer Methode sollte mit der Signatur des Skripts für das betreffende Ereignis übereinstimmen, nur dass Sie die Parameterliste verkürzen können, wenn Sie die Werte am Ende der Liste nicht benötigen. Sie sollten nur die benötigten Parameter akzeptieren, weil die Weitergabe der Parameter Overhead verursacht.

Methoden können jedoch nicht überladen werden. In der Systemtabelle `ml_script` kann nur ein Methodenprototyp per Klasse gespeichert werden.

### Methoden registrieren

Nach der Erstellung einer Methode müssen Sie sie registrieren. Durch das Registrieren der Methode wird eine Referenz der Methode in den MobiLink-Systemtabellen in der konsolidierten Datenbank erstellt, sodass die Methode im Falle eines Ereignisses aufgerufen wird. Sie registrieren Methoden auf die gleiche Art, mit der Sie Synchronisationsskripten hinzufügen. Anstatt jedoch der MobiLink-Systemtabelle das gesamte SQL-Skript hinzuzufügen, fügen Sie lediglich den Methodennamen hinzu.

Siehe „[Hinzufügungen und Löschungen von Skripten](#)“ auf Seite 258.

### Rückgabewerte

Methoden, die für einen Upload oder Download auf SQL-Basis aufgerufen werden, müssen eine gültige SQL-Anweisung zurückgeben. Der Rückgabetyt dieser Methoden muss `java.lang.String` sein. Andere Rückgabetypen sind nicht zulässig.

Der Rückgabetyt für alle anderen Skripten muss entweder `java.lang.String` oder `void` sein. Es sind keine anderen Typen zulässig. Wenn der Rückgabetyt eine Zeichenfolge und nicht Null ist, geht der MobiLink-Server davon aus, dass die Zeichenfolge eine gültige SQL-Anweisung enthält, und führt die Anweisung in der konsolidierten Datenbank aus, ebenso wie er es in einem gewöhnlichen SQL-Synchronisationsskript getan hätte. Wenn eine Methode normalerweise eine Zeichenfolge zurückgibt, bei Rückgabe jedoch keine SQL-Anweisung in der Datenbank ausgeführt werden soll, kann sie auch Null zurückgeben.

## Debuggen von Java-Klassen

MobiLink bietet verschiedene Informationen und Funktionen, die bei der Fehlersuche im Java-Code von Nutzen sein können. In diesem Abschnitt wird beschrieben, wo Sie diese Informationen finden und wie Sie sie benutzen können.

## Informationen in der Logdatei des MobiLink-Servers

Der MobiLink-Server schreibt Nachrichten in eine Nachrichtenlogdatei. Die Logdatei des Servers enthält die folgenden Informationen:

- Die Java-Laufzeitumgebung. Sie können die Option `-jrepath` verwenden, um eine bestimmte JRE anzufordern, wenn Sie den MobiLink-Server starten. Der Standardpfad ist der Pfad der mit SQL Anywhere 16 installierten JRE.
- Der Suchpfad der geladenen Standard-Java-Klassen. Falls Sie diese nicht ausdrücklich angeben, fügt der MobiLink-Server sie automatisch Ihrem Klassenpfad hinzu, bevor die Java VM aufgerufen wird.
- Die vollständig angegebenen Namen der aufgerufenen spezifischen Methoden. Mit diesen Angaben können Sie prüfen, ob der MobiLink-Server die richtigen Methoden aufruft.
- Alle von Java-Methoden an `java.lang.System.out` oder `java.lang.System.err` ausgegebenen Informationen werden in die Logdatei des MobiLink-Servers umgeleitet.
- Der `mlsrv16`-Befehlszeilenparameter `-verbose` kann verwendet werden.

Siehe „[mlsrv16-Option -v](#)“ auf Seite 86.

## Umgang mit einem Java-Debugger

Für die Fehlersuche in Ihren Java-Klassen können Sie einen standardmäßigen Java-Debugger benutzen. Geben Sie die erforderlichen Parameter in der `mlsrv16`-Befehlszeile mit der Option `-sl java` an.

Siehe „[mlsrv16-Option -sl java](#)“ auf Seite 79.

Wenn Sie einen Debugger angeben, stoppt die Java VM und wartet auf eine Verbindung von einem Java-Debugger.

## Informationen von Java ausgeben

Alternativ dazu können Sie wählen, Ihren Java-Methoden mithilfe von `java.lang.System.err` oder `java.lang.System.out` Anweisungen hinzuzufügen, die Informationen in das MobiLink-Nachrichtenlog ausgeben. Auf diese Weise können Sie den Verarbeitungsfortschritt und das Verhalten der Klassen protokollieren.

### Hinweis

Informationen auf diese Weise auszugeben, ist ein nützliches Hilfsmittel zur Überwachung, für eine Produktionsumgebung jedoch nicht empfehlenswert.

Dieselbe Methode kann genutzt werden, um zufällig ausgewählte Synchronisationsdaten zu protokollieren oder statistische Daten zur Verwendung Ihrer Skripten zu sammeln.

## Eigenen Testtreiber schreiben

Sie können Ihren eigenen Treiber schreiben, um Ihre Java-Klassen zu testen. Diese Vorgehensweise kann nützlich sein, weil damit die Aktionen Ihrer Java-Methoden vom Rest des MobiLink-Systems isoliert werden.

## MobiLink-Server-Fehlerbehandlung in Java

Wenn das Durchsuchen des Logs nicht ausreicht, können Sie Ihre Anwendungen mithilfe von Programmen überwachen. Sie können z.B. Nachrichten eines bestimmten Typs in einer E-Mail versenden.

Sie können Methoden erstellen, die einer Klasse übergeben werden, die jeden Fehler oder jede Warnung umfasst, die im Log gespeichert werden. Dies kann Ihnen helfen, einen MobiLink-Server zu überwachen und mit der Audit-Funktion zu überprüfen.

Der folgende Code installiert einen LogListener für alle Warnungen und schreibt die Informationen in eine Datei.

```
class TestLogListener implements LogListener {
    FileOutputStream _out_file;

    public TestLogListener(FileOutputStream out_file) {
        _out_file = out_file;
    }

    public void messageLogged(ServerContext sc, LogMessage msg) {
        String type;
        String user;

        try {
            if (msg.getType() == LogMessage.ERROR) {
                type = "ERROR";
            } else if (msg.getType() == LogMessage.WARNING) {
                type = "WARNING";
            }
            else {
                type = "UNKNOWN!!!";
            }

            user = msg.getUser();
            if (user == null) {
                user = "NULL";
            }

            _out_file.write(("Caught msg type="
                + type
                + " user=" + user
                + " text=" + msg.getText()
                + "\n").getBytes());
            _out_file.flush();
        }
        catch(Exception e) {
            // Print some error output to the MobiLink log.
            e.printStackTrace();
        }
    }
}
```

Der folgende Code registriert TestLogListener zum Empfang von Warnmeldungen. Rufen Sie diesen Code dort auf, von wo aus Sie Zugriff auf den ServerContext haben, z.B. in einem Klassenkonstruktor oder einem Synchronisationsskript.

```
// ServerContext serv_context;
serv_context.addWarningListener(
    new MyLogListener(ll_out_file)
);
```

**Siehe auch**

- [ServerContext.addErrorListener-Methode \[MobiLink-Server Java\] auf Seite 507](#)
- [ServerContext.removeErrorListener-Methode \[MobiLink-Server Java\] auf Seite 512](#)
- [ServerContext.addWarningListener-Methode \[MobiLink-Server Java\] auf Seite 509](#)
- [ServerContext.removeWarningListener-Methode \[MobiLink-Server Java\] auf Seite 513](#)
- [LogListener-Schnittstelle \[MobiLink-Server Java\] auf Seite 495](#)
- [LogMessage-Klasse \[MobiLink-Server Java\] auf Seite 496](#)

## Benutzerdefinierte Startklassen

Sie können Startklassen definieren, die beim Serverstart automatisch geladen werden. Diese Funktion gestattet es Ihnen, Java-Code zu schreiben, der ausgeführt wird, wenn der MobiLink-Server die Java VM startet, also vor der ersten Synchronisation. Das bedeutet, dass Sie vor der Synchronisationsanforderung eines Benutzers Verbindungen oder Cachedaten erstellen können.

Hierzu verwenden Sie die Option DMLStartClasses der mlsrv16-Option -sl java. Die mlsrv16-Befehlszeile kann beispielsweise Folgendes enthalten. Damit wird bewirkt, dass mycl1 und mycl2 als Startklassen geladen werden.

```
-sl java(-DMLStartClasses=com.test.mycl1,com.test.mycl2)
```

Klassen werden in der Reihenfolge geladen, in der sie aufgelistet sind. Wenn dieselbe Klasse mehr als einmal aufgelistet ist, wird mehr als eine Instanz erstellt.

Alle Startklassen müssen "public" sein und einen öffentlichen Konstruktor haben, der keine Argumente oder ein Argument vom Typ `ianywhere.ml.script.ServerContext` akzeptiert.

Die Namen der geladenen Startklassen werden mit der Meldung "Java-Startklasse geladen: *Klassenname*" in der MobiLink-Protokolldatei ausgegeben.

Weitere Hinweise zu den Optionen der Java VM finden Sie unter „[mlsrv16-Option -sl java](#)“ auf Seite 79.

Welche Startklassen beim Serverstart erstellt werden, finden Sie unter [ServerContext.getStartClassInstances-Methode \[MobiLink-Server Java\] auf Seite 511](#).

**Beispiel**

Im Folgenden sehen Sie eine Vorlage für eine Startklasse. Sie startet einen Daemon-Thread, der Ereignisse verarbeitet und eine Datenbankverbindung erstellt. (Nicht alle Startklassen müssen einen Thread erstellen, aber wenn ein Thread angestoßen wird, muss es sich um einen Daemon-Thread handeln.)

```
import ianywhere.ml.script.*;
import java.sql.*;

public class StartTemplate extends
    Thread implements ShutdownListener {
    ServerContext    _sc;
    Connection       _conn;
    boolean          _exit_loop;

    public StartTemplate(ServerContext sc) throws SQLException {
```

```
// Perform setup first so that an exception
// causes MobiLink startup to fail.
_sc = sc;

// Create a connection for use later.
_conn = _sc.makeConnection();

_exit_loop = false;
setDaemon(true);
start();
}

public void run() {
    _sc.addShutdownListener(this);

    // run() cannot throw exceptions.
    try {
        handlerLoop();
        _conn.close();
        _conn = null;
    }
    catch(Exception e) {

        // Print some error output to the MobiLink log.
        e.printStackTrace();

        // This thread shuts down and so does not
        // need to be notified of shutdown.
        _sc.removeShutdownListener(this);

        // Ask server to shutdown so that this fatal
        // error is fixed.
        _sc.shutdown();
    }
    // Shortly after return this thread no longer exists.
    return;
}

// stop our event handler loop
public void shutdownPerformed(ServerContext sc) {
    try {
        // Wait max 10 seconds for thread to die.
        join(10*1000);
    }
    catch(Exception e) {
        // Print some error output to the MobiLink log.
        e.printStackTrace();
    }
}

private void handlerLoop() throws InterruptedException {
    while (!_exit_loop) {
        // Handle events in this loop. Sleep not
        // needed, block on event queue.
        sleep(1 * 1000);
    }
}
}
```

## Java-Synchronisationsbeispiel

Java-Synchronisationslogik arbeitet mit MobiLink und gemeinsam verwendeten Java-Klassen und bietet Ihnen Flexibilität beim Deployment von Anwendungen im Zusammenhang mit dem MobiLink-Server. Der folgende Abschnitt ist eine Einführung in diesen erweiterten Funktionsbereich anhand eines einfachen Beispiels.

In diesem Abschnitt wird ein funktionierendes Beispiel für Java-Synchronisationslogik beschrieben. Bevor Sie versuchen, diese Klasse einzusetzen, bzw. Ihre eigene Klasse zu schreiben, stellen Sie mithilfe der folgenden Checkliste sicher, dass alle Bestandteile am richtigen Ort vorhanden sind, bevor Sie die Klasse kompilieren.

- Planen Sie die Funktionen, z.B. mit Pseudocode.
- Erstellen Sie eine Übersicht der Datenbanktabellen und –spalten.
- Konfigurieren Sie die konsolidierte Datenbank für Java, indem Sie sicherstellen, dass in den MobiLink-Systemtabellen Sprachtyp und Speicherort der Java-Synchronisationsmethoden angegeben sind.  
Siehe „[Java-Synchronisationslogik einrichten](#)“ auf Seite 463.
- Erstellen Sie eine Liste der zugeordneten Java-Klassen, die beim Ausführen Ihrer Java-Klassen aufgerufen werden.
- Speichern Sie Ihre Java-Klassen an einem Speicherort, der in der Classpath-Variablen für MobiLink enthalten ist.

### Plan

Die Java-Synchronisationslogik für dieses Beispiel verweist auf die zugeordneten Java-Dateien und –Klassen, die die für die Ausführung des Beispiels benötigten Funktionen enthalten. Das Beispiel zeigt, wie Sie eine CustEmpScripts-Klasse erstellen. Sie erfahren, wie ein Synchronisationskontext für die Verbindung eingerichtet wird. Schließlich liefert das Beispiel Java-Methoden für folgende Aufgaben:

- einen MobiLink-Benutzer authentifizieren.
- für jede einzelne Datenbanktabelle Download- und Uploadvorgänge mit einem Cursor durchführen.

### Schema

Die zu synchronisierenden Tabellen sind emp und cust. Die Tabelle emp umfasst drei Spalten mit den Namen emp\_id, emp\_name und manager. Die Tabelle cust enthält die drei Spalten cust\_id, cust\_name und emp\_id. Alle Spalten in den einzelnen Tabellen werden synchronisiert. Die Zuordnung von der konsolidierten zur entfernten Datenbank ist so, dass Tabellennamen und Spaltennamen in beiden Datenbanken identisch sind. Eine zusätzliche Tabelle, eine Audit-Tabelle, wird der konsolidierten Datenbank hinzugefügt.

### Java-Klassen-Dateien

Die in diesem Beispiel enthaltenen Dateien befinden sich im Verzeichnis *Samples\MobiLink\JavaAuthentication*.

## Setup

Der folgende Code richtet die Java-Synchronisationslogik ein. Über die Import-Anweisungen wird der Java VM der Speicherort der benötigten Dateien mitgeteilt. Mit der öffentlichen Klassenanweisung wird die Klasse deklariert.

```
// Use a package when you create your own script.
import ianywhere.ml.script.InOutInteger;
import ianywhere.ml.script.DBConnectionContext;
import ianywhere.ml.script.ServerContext;
import java.sql.*;

public class CustEmpScripts {
    // Context for this synchronization connection.
    DBConnectionContext _conn_context;

    // Same connection MobiLink uses for sync.
    // Do not commit or close this.
    Connection _sync_connection;
    Connection _audit_connection;

    //Get a user id given the user name. On audit connection.
    PreparedStatement _get_user_id_pstmt;

    // Add record of user logins added. On audit connection.
    PreparedStatement _insert_login_pstmt;

    // Prepared statement to add a record to the audit table.
    // On audit connection.
    PreparedStatement _insert_audit_pstmt;

    // ...
}
```

Der Konstruktor CustEmpScripts passt alle vorbereiteten Anweisungen für die Methode authenticateUser an. Er richtet Mitgliederdaten ein.

```
public CustEmpScripts(DBConnectionContext cc) throws SQLException {
    try {
        _conn_context = cc;
        _sync_connection = _conn_context.getConnection();

        ServerContext serv_context =
            _conn_context.getServerContext();
        _audit_connection = serv_context.makeConnection();

        // Get the prepared statements ready.
        _get_user_id_pstmt =
            _audit_connection.prepareStatement(
                "select user_id from ml_user where name = ?"
            );

        _insert_login_pstmt =
            _audit_connection.prepareStatement(
                "INSERT INTO login_added(ml_user, add_time) "
                + "VALUES (?, { fn CONVERT({ fn NOW() }, SQL_VARCHAR) })"
            );

        _insert_audit_pstmt =
            _audit_connection.prepareStatement(
                "INSERT INTO login_audit(ml_user_id, audit_time,
```

```

audit_action) "
+ "VALUES (?, { fn CONVERT({ fn NOW() },
SQL_VARCHAR) }, ?)"
);
}
catch(SQLException e) {
    freeJDBCResources();
    throw e;
}
catch(Error e) {
    freeJDBCResources();
    throw e;
}
}
}

```

Die Methode `finalize` bereinigt JDBC-Ressourcen, falls `end_connection` nicht aufgerufen wird. Sie ruft die Methode `freeJDBCResources` auf, wodurch zugewiesener Speicher freigegeben und die Audit-Verbindung geschlossen wird.

```

protected void finalize() throws SQLException, Throwable {
    super.finalize();
    freeJDBCResources();
}

private void freeJDBCResources() throws SQLException {
    if (_get_user_id_pstmt != null) {
        _get_user_id_pstmt.close();
    }
    if (_insert_login_pstmt != null) {
        _insert_login_pstmt.close();
    }
    if (_insert_audit_pstmt != null) {
        _insert_audit_pstmt.close();
    }
    if (_audit_connection != null) {
        _audit_connection.close();
    }
    _conn_context      = null;
    _sync_connection   = null;
    _audit_connection  = null;
    _get_user_id_pstmt = null;
    _insert_login_pstmt = null;
    _insert_audit_pstmt = null;
}

```

Die Methode `endConnection` bereinigt die Ressourcen, wenn diese nicht mehr benötigt werden.

```

public void endConnection() throws SQLException {
    freeJDBCResources();
}

```

Die Methode `authenticateUser` weiter unten genehmigt alle Benutzerlogins und protokolliert die Benutzerdaten in Datenbanktabellen. Wenn ein Benutzer nicht in der Tabelle `ml_user` enthalten ist, wird er in `login_added` erfasst. Wenn die Benutzer-ID in `ml_user` enthalten ist, wird sie in `login_audit` erfasst. In einem echten System muss auch das `user_password` berücksichtigt werden. Der Einfachheit halber werden in diesem Beispiel alle Benutzer genehmigt. Die Methode `endConnection` gibt eine SQL-Ausnahmebedingung aus, falls einer der Datenbankvorgänge mit einer Ausnahme fehlschlägt.

```

public void authenticateUser(
    InOutInteger authentication_status,

```

```

        String user_name) throws SQLException
    {

        boolean new_user;
        int user_id;

        // Get ml_user id.
        _get_user_id_pstmt.setString(1, user_name);

        ResultSet user_id_rs =
            _get_user_id_pstmt.executeQuery();
        new_user = !user_id_rs.next();
        if (!new_user) {
            user_id = user_id_rs.getInt(1);
        }
        else {
            user_id = 0;
        }

        user_id_rs.close();
        user_id_rs = null;

        // In this tutorial always allow the login.
        authentication_status.setValue(1000);

        if (new_user) {
            _insert_login_pstmt.setString(1, user_name);
            _insert_login_pstmt.executeUpdate();
            java.lang.System.out.println("user: " + user_name + " added. ");
        }
        else {
            _insert_audit_pstmt.setInt(1, user_id);
            _insert_audit_pstmt.setString(2, "LOGIN ALLOWED");
            _insert_audit_pstmt.executeUpdate();
        }
        _audit_connection.commit();
        return;
    }

```

Die folgenden Methoden benutzen SQL-Anweisungen als Cursor in den Datenbanktabellen. Da es sich dabei um Cursor-Skripten handelt, müssen sie eine SQL-Zeichenfolge zurückgeben.

```

public static String empUploadInsertStmt() {
    return("INSERT INTO emp(emp_id, emp_name) VALUES(?, ?)");
}

public static String empUploadDeleteStmt() {
    return("DELETE FROM emp WHERE emp_id = ?");
}

public static String empUploadUpdateStmt() {
    return("UPDATE emp SET emp_name = ? WHERE emp_id = ?");
}

public static String empDownloadCursor() {
    return("SELECT emp_id, emp_name FROM emp");
}

public static String custUploadInsertStmt() {
    return("INSERT INTO cust(cust_id, emp_id, cust_name) VALUES (?, ?, ?)");
}

public static String custUploadDeleteStmt() {

```

```

        return("DELETE FROM cust WHERE cust_id = ?");
    }

    public static String custUploadUpdateStmt() {
        return("UPDATE cust SET emp_id = ?, cust_name = ? WHERE cust_id = ?");
    }

    public static String custDownloadCursor() {
        return("SELECT cust_id, emp_id, cust_name FROM cust");
    }

```

Mit dem folgenden Befehl kompilieren Sie den Code:

```
javac -cp %sqlany16%\java\mlscript.jar CustEmpScripts.java
```

Führen Sie den MobiLink-Server mit der Position von CustEmpScripts.class in der Classpath-Variablen aus. Im Folgenden finden Sie den Ausschnitt einer Befehlszeile:

```
mlsrv16 ... -sl java (-cp <class_location>)
```

## Referenz der MobiLink-Server-API für Java

Dieser Abschnitt beschreibt die MobiLink Java-Schnittstellen und -Klassen und die ihnen zugeordneten Methoden und Konstruktoren. Zur Verwendung dieser Klassen referenzieren Sie die Assembly *mlscript.jar*, die sich in *%SQLANY16%\java* befindet.

### Paket

```
ianywhere.ml.script
```

## DBConnectionContext-Schnittstelle

Ruft Informationen über die aktuelle Datenbankverbindung ab.

### Syntax

```
public interface DBConnectionContext
```

### Mitglieder

Alle Mitglieder der DBConnectionContext-Schnittstelle, einschließlich aller geerbten Mitglieder.

| Name                                    | Beschreibung                                                                     |
|-----------------------------------------|----------------------------------------------------------------------------------|
| <a href="#">getConnection-Methode</a>   | Gibt die vorhandene Verbindung zur konsolidierten MobiLink-Datenbank zurück.     |
| <a href="#">getDownloadData-Methode</a> | Gibt das DownloadData-Objekt für die aktuelle Synchronisation zurück.            |
| <a href="#">getNetworkData-Methode</a>  | Gibt Informationen über die Netzwerkdatenströme für eine Synchronisation zurück. |

| Name                                     | Beschreibung                                                                                   |
|------------------------------------------|------------------------------------------------------------------------------------------------|
| <a href="#">getProperties-Methode</a>    | Gibt die Eigenschaften für diese Verbindung basierend auf der Skriptversion zurück.            |
| <a href="#">getRemoteID-Methode</a>      | Liefert die entfernte ID der Datenbank, die zurzeit über diese Verbindung synchronisiert wird. |
| <a href="#">getServerContext-Methode</a> | Gibt den Serverkontext für diesen MobiLink-Server zurück.                                      |
| <a href="#">getVersion-Methode</a>       | Gibt die Versionszeichenfolge für diese Verbindung zurück.                                     |

### Bemerkungen

Diese Informationen werden an den Konstruktor von Klassen übergeben, die Skripten enthalten. Wenn Kontext für einen Hintergrund-Thread oder über die Verbindungsdauer hinaus benötigt wird, verwenden Sie die **ServerContext**-Schnittstelle.

Weitere Hinweise zu Konstruktoren finden Sie unter „[Konstruktoren](#)“ auf Seite 467.

#### Hinweis

Eine **DBConnectionContext**-Instanz sollte nicht außerhalb des Threads verwendet werden, der Ihren Java-Code aufruft.

### Siehe auch

- [ServerContext-Schnittstelle \[MobiLink-Server Java\]](#) auf Seite 505

### Beispiel

Im folgenden Beispiel wird gezeigt, wie Sie eine **DBConnectionContext**-Instanz auf Klassenebene erstellen, die in Ihren Synchronisationsskripten verwendet werden kann. Die Methode **DBConnectionContext.getConnection** ruft eine **DBConnection**-Instanz ab, die die aktuelle Verbindung zur konsolidierten MobiLink-Datenbank darstellt.

```
using iAnywhere.MobiLink.Script;
using System.Data;

public class OrderProcessor {
    DBConnectionContext _cc;

    public OrderProcessor( DBConnectionContext cc ) {
        _cc = cc;
    }

    // The method used for the handle_DownloadData event.
    public void HandleEvent() {
        DBConnection my_connection = _cc.GetConnection();
        // ...
    }

    // ...
}
```

## getConnection-Methode

Gibt die vorhandene Verbindung zur konsolidierten MobiLink-Datenbank zurück.

### Syntax

```
java.sql.Connection DBConnectionContext.getConnection()  
throws SQLException
```

### Rückgabe

Eine vorhandene Verbindung als JDBC-Verbindung.

### Ausnahmen

- **java.sql.SQLException** Wird ausgelöst, wenn beim Binden der vorhandenen Verbindung als JDBC-Verbindung ein Fehler aufgetreten ist.

### Bemerkungen

Diese Verbindung ist dieselbe, die MobiLink bei der Ausführung von SQL-Skripten für diese Synchronisation verwendet.

Diese Verbindung darf nicht auf eine Weise festgeschrieben, getrennt oder geändert werden, die die Nutzung dieser Verbindung durch den MobiLink-Server beeinträchtigen könnte. Die zurückgegebene Verbindung gilt nur für die Dauer der zugrunde liegenden MobiLink-Verbindung.

#### Hinweis

Verwenden Sie die Verbindung nicht, nachdem das Ereignis end\_connection für die Verbindung aufgerufen wurde.

### Siehe auch

- [ServerContext.makeConnection-Methode \[MobiLink-Server Java\] auf Seite 511](#)
- [DBConnectionContext-Schnittstelle \[MobiLink-Server Java\] auf Seite 477](#)

## getDownloadData-Methode

Gibt das DownloadData-Objekt für die aktuelle Synchronisation zurück.

### Syntax

```
DownloadData DBConnectionContext.getDownloadData()
```

### Rückgabe

DownloadData für die aktuelle Synchronisation. Null, wenn diese Synchronisation keinen Download enthält.

### Bemerkungen

Verwenden Sie die DownloadData-Klasse, um den Download für direkte Zeilenbehandlung zu erstellen. Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

**Siehe auch**

- [DownloadData-Schnittstelle \[MobiLink-Server Java\] auf Seite 483](#)

**Beispiel**

Im folgenden Beispiel wird gezeigt, wie Sie eine **DownloadData**-Instanz für die aktuelle Synchronisation mit der **DBConnectionContext.getDownloadData**-Methode abrufen. In diesem Beispiel wird davon ausgegangen, dass Sie eine **DBConnectionContext**-Instanz namens **\_cc** erstellt haben.

```
// The method used for the handle_DownloadData event.
public void HandleDownload() throws SQLException {
    // Get the DownloadData for the current synchronization
    DownloadData my_dd = _cc.getDownloadData();
    // ...
}

// ...
```

## getNetworkData-Methode

Gibt Informationen über die Netzwerkdatenströme für eine Synchronisation zurück.

**Syntax**

```
NetworkData DBConnectionContext.getNetworkData( )
```

**Rückgabe**

Informationen über die Netzwerkdatenströme, die für die Anforderung verwendet werden, oder NULL, wenn die Sammlung nicht aktiviert wurde.

**Bemerkungen**

Diese Methode ist nützlich, wenn die Benutzerauthentifizierung bei einem anderen Server im Unternehmen erfolgt, der das clientseitige Zertifikat und HTTP-Header verwendet.

Zum Aktivieren einer Sammlung von Netzwerk-Datenströmen fügen Sie den -x Parametern `collect_network_data = 1` hinzu. Diese Option fügt für die Synchronisation zusätzlichen Speicher-Overhead zum Speichern der Daten hinzu.

**Siehe auch**

- [NetworkData-Schnittstelle \[MobiLink-Server Java\] auf Seite 501](#)

## getProperties-Methode

Gibt die Eigenschaften für diese Verbindung basierend auf der Skriptversion zurück.

**Syntax**

```
Properties DBConnectionContext.getProperties( )
```

## Rückgabe

Die Eigenschaften für diese Verbindung.

## Bemerkungen

Die Eigenschaften werden in der Tabelle **ml\_property** gespeichert. Weitere Hinweise finden Sie unter „[ml\\_add\\_property-Systemprozedur](#)“ auf Seite 664.

Weitere Hinweise zu **java.util.Properties** finden Sie in Ihrer Java Software Development Kit-Dokumentation.

## Beispiel

Das folgende Beispiel zeigt, wie Sie die Eigenschaften für **DBConnectionContext** ausgeben. Bei diesem Beispiel wird vorausgesetzt, dass eine **DBConnectionContext**-Instanz namens **\_cc** vorhanden ist.

```
// The method used to output the connection properties.
public void outputProperties() {
    // Output the properties for the current synchronization
    java.util.Properties properties = _cc.getProperties();
    System.out.println(properties.toString());
}
```

## getRemoteID-Methode

Liefert die entfernte ID der Datenbank, die zurzeit über diese Verbindung synchronisiert wird.

## Syntax

```
String DBConnectionContext.getRemoteID()
```

## Rückgabe

Die entfernte ID.

## Bemerkungen

Weitere Hinweise zu entfernten IDs finden Sie unter „[Entfernte IDs](#)“ [*MobiLink - Clientadministration*].

## Beispiel

Das folgende Beispiel zeigt, wie Sie die entfernte ID für ein **DBConnectionContext** ausgeben. Bei diesem Beispiel wird vorausgesetzt, dass eine **DBConnectionContext**-Instanz namens **\_cc** vorhanden ist.

```
// The method used to output the remote ID.
public void outputRemoteID() {
    // output the Remote ID for the current synchronization
    String remoteID = _cc.getRemoteID();
    System.out.println(remoteID);
}
```

## getServerContext-Methode

Gibt den Serverkontext für diesen MobiLink-Server zurück.

**Syntax**

```
ServerContext DBConnectionContext.getServerContext()
```

**Rückgabe**

Der MobiLink-Server-Kontext.

**Siehe auch**

- [ServerContext-Schnittstelle \[MobiLink-Server Java\] auf Seite 505](#)

**Beispiel**

Das folgende Beispiel zeigt, wie Sie die **ServerContext**-Instanz für ein **DBConnectionContext** abrufen. Bei diesem Beispiel wird vorausgesetzt, dass eine **DBConnectionContext**-Instanz namens **\_cc** vorhanden ist.

```
// A method that uses a ServerContext instance to shut down the server
public void shutDownServer() {
    ServerContext context = _cc.getServerContext();
    context.shutdown();
}
```

## getVersion-Methode

Gibt die Versionszeichenfolge für diese Verbindung zurück.

**Syntax**

```
String DBConnectionContext.getVersion()
```

**Rückgabe**

Die Skriptversion

**Bemerkungen**

Weitere Hinweise finden Sie unter „[ml\\_add\\_property-Systemprozedur](#)“ auf Seite 664.

**Beispiel**

Das folgende Beispiel zeigt, wie Sie die Skriptversion abrufen und abhängig von ihrem Wert fortfahren. Bei diesem Beispiel wird vorausgesetzt, dass eine **DBConnectionContext**-Instanz namens **\_cc** vorhanden ist.

```
// A method that uses the script version
public void handleEvent() {
    // ...

    String version = _cc.getVersion();
    if (version.equals("My Version 1")) {
        // ...
    } else if (version.equals("My Version 2")) {
        // ...
    }
}
// ...
```

## DownloadData-Schnittstelle

Verpackt Datendownload-Vorgänge für die direkte Zeilenbehandlung.

### Syntax

```
public interface DownloadData
```

### Mitglieder

Alle Mitglieder der DownloadData-Schnittstelle, einschließlich aller geerbten Mitglieder.

| Name                                           | Beschreibung                                                        |
|------------------------------------------------|---------------------------------------------------------------------|
| <a href="#">getDownloadTableByName-Methode</a> | Ruft die benannte Downloadtabelle für diese Synchronisation ab.     |
| <a href="#">getDownloadTables-Methode</a>      | Ruft ein Array aller Downloadtabellen in dieser Synchronisation ab. |

### Bemerkungen

Verwenden Sie die **DBConnectionContext.getDownloadData**-Methode, um eine **DownloadData**-Instanz abzurufen.

Verwenden Sie die **getDownloadTables**- und **getDownloadTableByName**-Methoden, um **DownloadTableData**-Instanzen zurückzugeben

Diese Download-Daten sind über **DBConnectionContext** verfügbar. Vor dem begin\_synchronization-Ereignis oder nach dem end\_download-Ereignis kann nicht auf die Download-Daten zugegriffen werden. In einer Synchronisation mit "Nur Upload" ist der Zugriff auf **DownloadData** ungültig. Weitere Hinweise finden Sie unter „[handle\\_DownloadData \(Verbindungsereignis\)](#)“ auf Seite 371.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

### Siehe auch

- [DBConnectionContext.getDownloadData-Methode \[MobiLink-Server Java\] auf Seite 479](#)
- [DownloadTableData-Schnittstelle \[MobiLink-Server Java\] auf Seite 485](#)

### Beispiel

Im folgenden Beispiel wird gezeigt, wie Sie eine **DownloadData**-Instanz für die aktuelle Synchronisation mit der **DBConnectionContext.getDownloadData**-Methode abrufen:

```
DBConnectionContext _cc;

// Your class constructor.
public OrderProcessor(DBConnectionContext cc) {
    _cc = cc;
}

// The method used for the handle_DownloadData event.
public void handleDownload() throws SQLException {
    // Get the DownloadData for the current synchronization.
```

```
DownloadData my_dd = _cc.getDownloadData();  
    // ...  
}
```

## getDownloadTableByName-Methode

Ruft die benannte Downloadtabelle für diese Synchronisation ab.

### Syntax

```
DownloadTableData DownloadData.getDownloadTableByName(String table_name)
```

### Parameter

- **table\_name** Der Name der Tabelle, deren Daten Sie herunterladen möchten.

### Rückgabe

Eine **DownloadTableData**-Instanz, die die angegebene Tabelle darstellt, oder Null, wenn keine Tabelle mit dem angegebenen Namen für die aktuelle Synchronisation vorhanden ist.

### Bemerkungen

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter [„Direkte Zeilenbehandlung“ auf Seite 627](#).

### Siehe auch

- [DownloadData-Schnittstelle \[MobiLink-Server Java\] auf Seite 483](#)
- [DownloadTableData-Schnittstelle \[MobiLink-Server Java\] auf Seite 485](#)
- [DBConnectionContext-Schnittstelle \[MobiLink-Server Java\] auf Seite 477](#)

### Beispiel

Im folgenden Beispiel wird mit der **getDownloadTableByName**-Methode eine **DownloadTableData**-Instanz für die **remoteOrders**-Tabelle zurückgegeben. In diesem Beispiel wird davon ausgegangen, dass Sie eine **DBConnectionContext**-Instanz namens **\_cc** erstellt haben.

```
// The method used for the handle_DownloadData event.  
public void handleDownload() throws SQLException {  
    // Get the DownloadData for the current synchronization.  
    DownloadData my_dd = _cc.getDownloadData();  
  
    // Get the DownloadTableData for the remoteOrders table.  
    DownloadTableData my_download_table =  
    my_dd.getDownloadTableByName("remoteOrders");  
  
    // ...  
}
```

## getDownloadTables-Methode

Ruft ein Array aller Downloadtabellen in dieser Synchronisation ab.

## Syntax

```
DownloadTableData[] DownloadData.getDownloadTables()
```

## Rückgabe

Ein Array von **DownloadTableData**-Objekten für die aktuelle Synchronisation. Die Reihenfolge der Tabellen im Array ist die gleiche wie die entfernte Upload-Reihenfolge.

## Bemerkungen

Die in dieser Tabelle durchgeführten Vorgänge werden an die entfernte Datenbank gesendet.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

## Siehe auch

- [DownloadData-Schnittstelle \[MobiLink-Server Java\] auf Seite 483](#)
- [DownloadTableData-Schnittstelle \[MobiLink-Server Java\] auf Seite 485](#)
- [DBConnectionContext-Schnittstelle \[MobiLink-Server Java\] auf Seite 477](#)

## Beispiel

Im nachfolgenden Beispiel wird mit der **DownloadData.getDownloadTables**-Methode ein Array von **DownloadTableData**-Objekten für die aktuelle Synchronisation abgerufen. Bei diesem Beispiel wird vorausgesetzt, dass eine **DBConnectionContext**-Instanz namens **\_cc** vorhanden ist.

```
// The method used for the handle_DownloadData event.
public void handleDownload() throws SQLException {
    // Get the DownloadData for the current synchronization.
    DownloadData my_dd = _cc.getDownloadData();

    // Get an array of tables to set download operations.
    DownloadTableData[] download_tables = my_dd.getDownloadTables();

    // Get the first table in the DownloadTableData array.
    DownloadTableData my_download_table = download_tables[0];

    // ...
}
```

## DownloadTableData-Schnittstelle

Kapselt Informationen für eine Downloadtabelle für eine Synchronisation.

## Syntax

```
public interface DownloadTableData
```

## Mitglieder

Alle Mitglieder der DownloadTableData-Schnittstelle, einschließlich aller geerbten Mitglieder.

| Name                                               | Beschreibung                                                                                                                                                                        |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getDeletePreparedStatement-Methode</a> | Gibt eine <b>java.sql.PreparedStatement</b> -Instanz zurück, mit der der Benutzer dem Download Löschvorgänge hinzufügen kann.                                                       |
| <a href="#">getLastDownloadTime-Method</a>         | Gibt die letzte Downloadzeit für diese Tabelle zurück.                                                                                                                              |
| <a href="#">getMetaData-Methode</a>                | Ruft die Metadaten für die <b>DownloadTableData</b> -Instanz ab.                                                                                                                    |
| <a href="#">getName-Methode</a>                    | Gibt den Tabellennamen für die DownloadTableData-Instanz zurück.                                                                                                                    |
| <a href="#">getUpsertPreparedStatement-Methode</a> | Gibt eine <b>java.sql.PreparedStatement</b> -Instanz zurück, mit der der Benutzer dem Download einer Synchronisation upsert-Vorgänge (Einfügen oder Aktualisieren) hinzufügen kann. |

### Bemerkungen

Richten Sie mit dieser Schnittstelle die Datenvorgänge ein, die auf den Client geladen werden.

Sie können mit der **DownloadData**-Schnittstelle **DownloadTableData**-Instanzen für die aktuelle Synchronisation abrufen. Sie können mit den Methoden **getUpsertPreparedStatement** und **getDeletePreparedStatement** mit Java vorbereitete Anweisungen für Einfügungen, Aktualisierungen bzw. Löschungen abrufen.

Sie können die DELETE-Anweisung mit allen Primärschlüssel auf NULL ausführen, damit der entfernte Client die Tabelle kürzt.

Die **java.sql.PreparedStatement.executeUpdate**-Methode registriert einen Vorgang für den Download. Weitere Hinweise zu **java.sql.PreparedStatement** finden Sie in Ihrer Java Software Development Kit-Dokumentation.

#### Hinweis

Richten Sie alle Spaltenwerte für vorbereitete Einfüge- und Aktualisierungsanweisungen (Insert und Update). Für Löschvorgänge (Delete) müssen Sie Primärschlüsselwerte festlegen. Die vorbereiteten Anweisungen zum Löschen einerseits und zum Einfügen und Aktualisieren andererseits können nicht gleichzeitig geöffnet sein.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

### Siehe auch

- [DownloadData-Schnittstelle \[MobiLink-Server Java\] auf Seite 483](#)

### Beispiel

In diesem Beispiel wird vorausgesetzt, dass die MobiLink-Client-Datenbanken eine Tabelle namens **remoteOrders** enthalten, die mit folgender SQL-Anweisung erstellt wurde:

```
CREATE TABLE remoteOrders (
    pk INT NOT NULL,
    coll VARCHAR(200),
    PRIMARY KEY (pk)
);
```

Im folgenden Beispiel wird mit der **DownloadData.getDownloadTableByName**-Methode eine **DownloadTableData**-Instanz zurückgegeben, die die Tabelle **remoteOrders** darstellt. Bei diesem Beispiel wird vorausgesetzt, dass eine **DBConnectionContext**-Instanz namens **\_cc** vorhanden ist.

```
// The method used for the handle_DownloadData event.
public void handleDownload() throws SQLException {

    // Get the DownloadData for the current synchronization.
    DownloadData my_dd = _cc.getDownloadData();

    // Get the DownloadTableData for the remoteOrders table.
    DownloadTableData td = my_dd.getDownloadTableByName("remoteOrders");

    // User defined-methods to set download operations.
    setDownloadInserts(td);
    setDownloadDeletes(td);
    // ...
}
```

In diesem Beispiel verwendet die **SetDownloadInserts**-Methode **GetUpsertCommand**, um einen Befehl für die einzufügenden oder zu aktualisierenden Zeilen abzurufen. **IDbCommand** enthält die Parameter, die auf die Werte gesetzt werden, die in die entfernte Datenbank eingefügt werden sollen.

```
void setDownloadInserts(DownloadTableData td) {
    java.sql.PreparedStatement insert_ps = td.getUpsertPreparedStatement();

    // The following method calls are the same as the following SQL
    statement:
    // INSERT INTO remoteOrders(pk, coll) values(2300, "truck");
    insert_ps.setInt(1, 2300);
    insert_ps.setString(2, "truck");

    int update_result = insert_ps.executeUpdate();
    if (update_result == 0) {
        // Insert was filtered because it was uploaded
        // in the same synchronization.
    }
    else {
        // Insert was not filtered.
    }
}
```

Die **setDownloadDeletes**-Methode ruft mit **DownloadTableData.getDeletePreparedStatement** eine vorbereitete Anweisung für zu löschende Zeilen ab. Die **java.sql.PreparedStatement.setInt**-Methode legt die Primärschlüsselwerte für Zeilen fest, die Sie in der entfernten Datenbank löschen möchten. Die **java.sql.PreparedStatement.executeUpdate**-Methode registriert die Zeilenwerte für den Download.

```
void setDownloadDeletes(DownloadTableData td) {
    java.sql.PreparedStatement delete_ps = td.getDeletePreparedStatement();

    // The following method calls are the same as the following SQL
    statement:
    // DELETE FROM remoteOrders where pk=2300;
    delete_ps.setInt(1, 2300);
}
```

```
        delete_ps.executeUpdate();  
    }
```

## getDeletePreparedStatement-Methode

Gibt eine **java.sql.PreparedStatement**-Instanz zurück, mit der der Benutzer dem Download Löschvorgänge hinzufügen kann.

### Syntax

```
java.sql.PreparedStatement DownloadTableData.getDeletePreparedStatement()  
    throws java.sql.SQLException
```

### Rückgabe

Eine **java.sql.PreparedStatement**-Instanz zum Hinzufügen von Löschvorgängen zum Download.

### Ausnahmen

- **java.sql.SQLException** Wird ausgelöst, wenn es Probleme beim Abrufen der gelöschten **java.sql.PreparedStatement**-Instanz gibt.

### Bemerkungen

Die vorbereitete Anweisung gilt für die **DownloadTableData**-Instanz und enthält einen Parameter für jede Primärschlüsselspalte in der Tabelle.

Um einen Löschvorgang in den Download einzubeziehen, richten Sie alle Spalten in **java.sql.PreparedStatement** ein und rufen anschließend die **java.sql.PreparedStatement.executeUpdate**-Methode auf.

Setzen Sie alle Parameter auf NULL, damit die entfernte Datenbank die Tabelle kürzt.

#### Hinweis

Sie müssen alle Primärschlüsselwerte festlegen, um Download-Löschvorgänge durchzuführen, oder alle Primärschlüsselwerte auf NULL setzen, um Kürzungsvorgänge durchzuführen.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

### Siehe auch

- [DownloadTableData-Schnittstelle \[MobiLink-Server Java\] auf Seite 485](#)

### Beispiel

Im folgenden Beispiel ruft die **setDownloadDeletes**-Methode mit **getDeletePreparedStatement** eine vorbereitete Anweisung für zu löschende Zeilen ab. Die **java.sql.PreparedStatement.setInt**-Methode legt die Primärschlüsselwerte für Zeilen fest, die Sie in der entfernten Datenbank löschen möchten. Die **java.sql.PreparedStatement.executeUpdate**-Methode legt die Zeilenwerte im Download fest.

```
void setDownloadDeletes(DownloadTableData td) {  
    java.sql.PreparedStatement delete_ps = td.getDeletePreparedStatement();  
    // This is the same as executing the following SQL statement:
```

```

        // DELETE FROM remoteOrders where pk=2300;
        delete_ps.setInt(1, 2300);
        delete_ps.executeUpdate();
        delete_ps.close();
    }

```

## getLastDownloadTime-Method

Gibt die letzte Downloadzeit für diese Tabelle zurück.

### Syntax

```
java.sql.Timestamp DownloadTableData.getLastDownloadTime()
```

### Rückgabe

Die letzte Downloadzeit für diese Tabelle.

### Bemerkungen

Dies ist dieselbe letzte Downloadzeit, die an mehrere der tabellenweisen Download-Ereignisse übergeben wurde.

Die letzte Downloadzeit dient zur Generierung der Download-Daten der Tabelle für eine bestimmte Synchronisation.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

### Siehe auch

- [DownloadTableData-Schnittstelle \[MobiLink-Server Java\] auf Seite 485](#)

### Beispiel

Im folgenden Abschnitt wird gezeigt, wie Sie eine Tabelle im Download mithilfe der letzten Downloadzeit mit Einfügungen füllen. Beachten Sie, dass bei diesem Beispiel vorausgesetzt wird, dass eine **DBConnectionContext**-Instanz namens **\_cc** vorhanden ist.

```

// The method used for the handle_DownloadData event.
public void handleDownload() throws SQLException {
    // Get the DownloadData for the current synchronization.
    DownloadData my_dd = _cc.getDownloadData();

    // Get the DownloadTableData for the remoteOrders table.
    DownloadTableData td = my_dd.getDownloadTableByName("remoteOrders");

    // Get the inserts given a last download time.
    ResultSet inserts_rs =
    makeInsertsFromTimestamp(td.getLastDownloadTime());

    // Fill the DownloadTableData using the inserts resultset.
    setDownloadInsertsFromRS(td, inserts_rs);
    inserts_rs.close();

    // ...
}

```

## getMetaData-Methode

Ruft die Metadaten für die **DownloadTableData**-Instanz ab.

### Syntax

```
java.sql.ResultSetMetaData DownloadTableData.getMetaData()
```

### Rückgabe

Die Metadaten für die **DownloadTableData**-Instanz.

### Bemerkungen

Die Metadaten sind ein standardmäßiges **java.sql.ResultSetMetaData**-Objekt.

Falls die Metadaten Spaltennameninformationen enthalten sollen, geben Sie auf dem Client an, dass mit dem Upload Spaltennamen gesendet werden sollen.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

Weitere Hinweise zu **java.sql.ResultSetMetaData** finden Sie in Ihrer Java Software Development Kit-Dokumentation.

### Siehe auch

- [DownloadTableData-Schnittstelle \[MobiLink-Server Java\] auf Seite 485](#)

### Beispiel

Das folgende Beispiel zeigt, wie Sie die Anzahl der Spalten abrufen, die in der Abfrage für die **DownloadTableData**-Instanz verwendet werden:

```
import java.sql.ResultSetMetaData;

// The method used to return the number of columns in a DownloadTableData
instance query
public int getNumColumns(DownloadTableData td) {
    ResultSetMetaData rsmd = td.getMetaData();
    return rsmd.getColumnCount();
}
```

## getName-Methode

Gibt den Tabellennamen für die **DownloadTableData**-Instanz zurück.

### Syntax

```
String DownloadTableData.getName()
```

### Rückgabe

Der Tabellenname für die **DownloadTableData**-Instanz.

## Bemerkungen

Sie können den Tabellennamen auch mit der **java.sql.ResultSetMetaData**-Instanz aufrufen, die von der **getMetaData**-Methode zurückgegeben wird.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

## Siehe auch

- [DownloadTableData-Schnittstelle \[MobiLink-Server Java\] auf Seite 485](#)
- [DownloadTableData.getMetaData-Methode \[MobiLink-Server Java\] auf Seite 490](#)

## Beispiel

Das folgende Beispiel zeigt, wie Sie den Tabellennamen für die **DownloadTableData**-Instanz ausgeben. Bei diesem Beispiel wird vorausgesetzt, dass eine **DBConnectionContext**-Instanz namens **\_cc** vorhanden ist.

```
// The method used for the handle_DownloadData event
public void handleDownload() throws SQLException {

    // Get the DownloadData for the current synchronization.
    DownloadData my_dd = _cc.getDownloadData();

    // Get the DownloadTableData for the remoteOrders table.
    DownloadTableData td = my_dd.getDownloadTableByName("remoteOrders");

    // Print the table name to standard output (remoteOrders)
    System.out.println(td.getName());

    // User defined-methods to set download operations.
    setDownloadInserts(td);
    setDownloadDeletes(td);

    // ...
}
```

## getUpsertPreparedStatement-Methode

Gibt eine **java.sql.PreparedStatement**-Instanz zurück, mit der der Benutzer dem Download einer Synchronisation upsert-Vorgänge (Einfügen oder Aktualisieren) hinzufügen kann.

### Syntax

```
java.sql.PreparedStatement DownloadTableData.getUpsertPreparedStatement( )
throws java.sql.SQLException
```

### Rückgabe

Eine **java.sql.PreparedStatement**-Instanz zum Hinzufügen von Upsert-Vorgängen zum Download.

### Ausnahmen

- **java.sql.SQLException** Wird ausgelöst, wenn es Probleme beim Abrufen der Upsert-**java.sql.PreparedStatement**-Instanz gibt.

## Bemerkungen

Die vorbereitete Anweisung gilt für die `DownloadTableData`-Instanz und enthält einen Parameter für jede Spalte in der Tabelle.

Um einen Einfüge- oder Aktualisierungsvorgang in den Download einzubeziehen, legen Sie alle Spaltenwerte in **`java.sql.PreparedStatement`** fest und rufen anschließend die **`java.sql.PreparedStatement.executeUpdate`**-Methode auf. Beim Aufruf von **`java.sql.PreparedStatement.executeUpdate`** in der vorbereiteten Anweisung wird der Wert 0 zurückgegeben, wenn der Einfüge- oder Aktualisierungsvorgang gefiltert wurde. Der Wert 1 wird zurückgegeben, wenn der Vorgang nicht gefiltert wurde. Ein Vorgang wird gefiltert, wenn er in derselben Synchronisation hochgeladen wurde.

### Hinweis

Legen Sie alle Spaltenwerte für Download-Einfüge- und Download-Aktualisierungsvorgänge fest.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

## Siehe auch

- [DownloadTableData-Schnittstelle \[MobiLink-Server Java\] auf Seite 485](#)

## Beispiel

Im folgenden Beispiel verwendet die **`setDownloadInserts`**-Methode **`getUpsertPreparedStatement`**, um eine vorbereitete Anweisung für Zeilen abzurufen, die Sie einfügen oder aktualisieren möchten. Die **`java.sql.PreparedStatement.setInt`**- und die **`PreparedStatement.setString`**-Methode legen die Spaltenwerte fest, und die **`PreparedStatement.executeUpdate`**-Methode legt die Zeilenwerte im Download fest

```
void setDownloadInserts(DownloadTableData td) {
    java.sql.PreparedStatement insert_ps = td.getUpsertPreparedStatement();

    // This is the same as executing the following SQL statement:
    // INSERT INTO remoteOrders(pk, coll) VALUES (2300, "truck");
    insert_ps.setInt(1, 2300);
    insert_ps.setString(2, "truck");

    int update_result = insert_ps.executeUpdate();
    if (update_result == 0) {
        // Insert was filtered because it was uploaded
        // in the same synchronization.
    }
    else {
        // Insert was not filtered.
    }
    insert_ps.close();
}
```

## InOutInteger-Schnittstelle

Wird an Methoden übergeben, um die Funktion eines an ein SQL-Skript übergebenen I/O-Parameters zu aktivieren.

## Syntax

```
public interface InOutInteger
```

## Mitglieder

Alle Mitglieder der InOutInteger-Schnittstelle, einschließlich aller geerbten Mitglieder.

| Name                             | Beschreibung                                    |
|----------------------------------|-------------------------------------------------|
| <a href="#">getValue-Methode</a> | Gibt den Wert dieses Ganzzahlparameters zurück. |
| <a href="#">setValue-Methode</a> | Legt den Wert dieses Ganzzahlparameters fest.   |

## Beispiel

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens **handleError** als Skript für das **handle\_error connection**-Ereignis, wenn die Skriptversion **ver1** synchronisiert wird.

```
CALL ml_add_java_connection_script(
  'ver1',
  'handle_error',
  'ExamplePackage.ExampleClass.handleError'
)
```

Dies ist die Java-Beispielmethode **handleError**. Sie verarbeitet einen Fehler basierend auf den Daten, die übergeben werden. Außerdem legt sie den sich daraus ergebenden Fehlercode fest.

```
public String handleError(
    ianywhere.ml.script.InOutInteger actionCode,
    int errorCode,
    String errorMessage,
    String user,
    String table)
{
    int new_ac;
    if (user == null) {
        new_ac = handleNonSyncError(errorCode, errorMessage);
    } else if (table == null) {
        new_ac = handleConnectionError(errorCode, errorMessage, user);
    }
    else {
        new_ac = handleTableError(errorCode, errorMessage, user, table);
    }

    // Keep the most serious action code.
    if (actionCode.getValue() < new_ac) {
        actionCode.setValue(new_ac);
    }
}
```

## getValue-Methode

Gibt den Wert dieses Ganzzahlparameters zurück.

**Syntax**

```
int InOutInteger.getValue()
```

**Rückgabe**

Der Wert dieser Ganzzahl.

**Siehe auch**

- [InOutInteger-Schnittstelle \[MobiLink-Server Java\] auf Seite 492](#)

**setValue-Methode**

Legt den Wert dieses Ganzzahlparameters fest.

**Syntax**

```
void InOutInteger.setValue(int new_value)
```

**Parameter**

- **new\_value** Der Wert für diese Ganzzahl.

**Siehe auch**

- [InOutInteger-Schnittstelle \[MobiLink-Server Java\] auf Seite 492](#)

**InOutString-Schnittstelle**

Wird an Methoden übergeben, um die Funktion eines an ein SQL-Skript übergebenen I/O-Parameters zu aktivieren.

**Syntax**

```
public interface InOutString
```

**Mitglieder**

Alle Mitglieder der InOutString-Schnittstelle, einschließlich aller geerbten Mitglieder.

| Name                             | Beschreibung                                         |
|----------------------------------|------------------------------------------------------|
| <a href="#">getValue-Methode</a> | Gibt den Wert dieses Zeichenfolgenparameters zurück. |
| <a href="#">setValue-Methode</a> | Legt den Wert dieses Zeichenfolgeparameters fest.    |

**Beispiel**

Der folgende Aufruf einer MobiLink-Systemprozedur registriert eine Java-Methode namens **modifyUser** als Skript für das **modify\_user**-Verbindungsereignis , wenn die Skriptversion **ver1** synchronisiert wird.

```
CALL ml_add_java_connection_script(  
    'ver1',
```

```
'modify_user',  
'ExamplePackage.ExampleClass.modifyUser'  
)
```

Dies ist die Java-Beispielmethode **modifyUser**. Sie erhält die Benutzer-ID von der Datenbank und verwendet sie dann, um den Benutzernamen festzulegen.

```
public String modifyUser(InOutString io_user_name) throws SQLException {  
    Statement uid_select = curConn.createStatement();  
    ResultSet uid_result = uid_select.executeQuery(  
        "SELECT rep_id FROM SalesRep WHERE name = '"  
        + io_user_name.getValue() + "' "  
    );  
    uid_result.next();  
    io_user_name.setValue(java.lang.Integer.toString(uid_result.getInt(1)));  
    uid_result.close();  
    uid_select.close();  
    return (null);  
}
```

## getValue-Methode

Gibt den Wert dieses Zeichenfolgenparameters zurück.

### Syntax

```
String InOutString.getValue()
```

### Rückgabe

Der Wert dieses Zeichenfolgeparameters.

### Siehe auch

- [InOutString-Schnittstelle \[MobiLink-Server Java\] auf Seite 494](#)

## setValue-Methode

Legt den Wert dieses Zeichenfolgeparameters fest.

### Syntax

```
void InOutString.setValue(String new_value)
```

### Parameter

- **new\_value** Der Wert für diese Zeichenfolge.

### Siehe auch

- [InOutString-Schnittstelle \[MobiLink-Server Java\] auf Seite 494](#)

## LogListener-Schnittstelle

Dient zum Erfassen von Nachrichten, die in das Log geschrieben werden.

**Syntax**

```
public interface LogListener
```

**Mitglieder**

Alle Mitglieder der LogListener-Schnittstelle, einschließlich aller geerbten Mitglieder.

| Name                                  | Beschreibung                                                      |
|---------------------------------------|-------------------------------------------------------------------|
| <a href="#">messageLogged-Methode</a> | Wird aufgerufen, wenn eine Nachricht in das Log geschrieben wird. |

**Bemerkungen**

Weitere Hinweise finden Sie unter „[MobiLink-Server-Fehlerbehandlung in Java](#)“ auf Seite 470.

## messageLogged-Methode

Wird aufgerufen, wenn eine Nachricht in das Log geschrieben wird.

**Syntax**

```
void LogListener.messageLogged(ServerContext sc, LogMessage message)
```

**Parameter**

- **sc** Kontext für den Server, der die Nachricht schreibt.
- **message** Die **LogMessage**, die an das MobiLink-Log gesendet wird.

**Bemerkungen**

Weitere Hinweise finden Sie unter „[MobiLink-Server-Fehlerbehandlung in Java](#)“ auf Seite 470.

## LogMessage-Klasse

Speichert die Daten, die einer Lognachricht zugeordnet sind.

**Syntax**

```
public class LogMessage
```

**Mitglieder**

Alle Mitglieder der LogMessage-Klasse, einschließlich aller geerbten Mitglieder.

| Name                            | Beschreibung                                    |
|---------------------------------|-------------------------------------------------|
| <a href="#">getText-Methode</a> | Ruft den dieser Nachricht zugeordneten Text ab. |
| <a href="#">getType-Methode</a> | Ruft diesen Nachrichtentyp ab.                  |

| Name                             | Beschreibung                                                    |
|----------------------------------|-----------------------------------------------------------------|
| <a href="#">getUser-Methode</a>  | Ruft den Benutzernamen ab, der dieser Nachricht zugeordnet ist. |
| <a href="#">ERROR-Variable</a>   | Zeigt an, dass die Logmeldung ein Fehler ist.                   |
| <a href="#">INFO-Variable</a>    | Zeigt an, dass das Meldungslog Informationen enthält.           |
| <a href="#">WARNING-Variable</a> | Zeigt an, dass die Logmeldung eine Warnung ist.                 |

### Bemerkungen

Weitere Hinweise finden Sie unter „[MobiLink-Server-Fehlerbehandlung in Java](#)“ auf Seite 470.

## getText-Methode

Ruft den dieser Nachricht zugeordneten Text ab.

### Syntax

```
String LogMessage.getText( )
```

### Rückgabe

Der Haupttext dieser Meldung

### Siehe auch

- [LogMessage.ERROR-Variable \[MobiLink-Server Java\]](#) auf Seite 498
- [LogMessage.INFO-Variable \[MobiLink-Server Java\]](#) auf Seite 499
- [LogMessage.WARNING-Variable \[MobiLink-Server Java\]](#) auf Seite 500

## getType-Methode

Ruft diesen Nachrichtentyp ab.

### Syntax

```
int LogMessage.getType( )
```

### Rückgabe

Der Typ dieser Nachricht, der **LogMessage.ERROR**, **LogMessage.INFO** oder **LogMessage.WARNING** sein kann.

### Siehe auch

- [LogMessage.ERROR-Variable \[MobiLink-Server Java\]](#) auf Seite 498
- [LogMessage.INFO-Variable \[MobiLink-Server Java\]](#) auf Seite 499
- [LogMessage.WARNING-Variable \[MobiLink-Server Java\]](#) auf Seite 500

## getUser-Methode

Ruft den Benutzernamen ab, der dieser Nachricht zugeordnet ist.

### Syntax

```
String LogMessage.getUser()
```

### Rückgabe

Der Benutzer, der dieser Nachricht zugeordnet ist. Dieser Wert kann Null sein, wenn die Nachricht keinen Benutzer hat.

### Siehe auch

- [LogMessage.ERROR-Variable \[MobiLink-Server Java\] auf Seite 498](#)
- [LogMessage.INFO-Variable \[MobiLink-Server Java\] auf Seite 499](#)
- [LogMessage.WARNING-Variable \[MobiLink-Server Java\] auf Seite 500](#)

## ERROR-Variable

Zeigt an, dass die Logmeldung ein Fehler ist.

### Syntax

```
final int LogMessage.ERROR
```

### Siehe auch

- [LogMessage-Klasse \[MobiLink-Server Java\] auf Seite 496](#)

### Beispiel

Der folgende Code installiert ein **LogListener**-Objekt für alle Fehlermeldungen und schreibt dann die Informationen in eine Datei.

```
class ErrorLogListener implements LogListener {
    FileOutputStream _outFile;

    public ErrorLogListener(FileOutputStream outFile) {
        _outFile = outFile;
    }

    public void messageLogged(ServerContext sc, LogMessage msg) {
        String user;

        try {
            if (msg.getType() != LogMessage.ERROR) {
                //this class deals exclusively with errors
                return;
            }

            user = msg.getUser();
            if (user == null) {
                user = "NULL";
            }

            _outFile.write(("Caught error"
```

```

        + " user=" + user
        + " text=" + msg.getText()
        + "\n").getBytes()
    );
    _outFile.flush();
}
catch(Exception e) {
    // Print some error output to the MobiLink log.
    e.printStackTrace();
}
}
}

```

Der folgende Code registriert ein **ErrorLogListener**-Objekt zum Empfang von Fehlermeldungen. Rufen Sie diesen Code dort auf, von wo aus Sie Zugriff auf den **ServerContext** haben, z.B. in einem Klassenkonstruktor oder einem Synchronisationsskript.

```

ServerContext serv_context;
FileOutputStream outFile
serv_context.addErrorListener(new ErrorLogListener(outFile));

```

## INFO-Variable

Zeigt an, dass das Meldungslog Informationen enthält.

### Syntax

```
final int LogMessage.INFO
```

### Siehe auch

- [LogMessage-Klasse \[MobiLink-Server Java\] auf Seite 496](#)
- [ServerContext.addInfoListener-Methode \[MobiLink-Server Java\] auf Seite 507](#)

### Beispiel

Der folgende Code installiert ein **LogListener**-Objekt für alle Informationsmeldungen und schreibt die Informationen in eine Datei.

```

class InfoLogListener implements LogListener {
    FileOutputStream _outFile;

    public InfoLogListener(FileOutputStream outFile) {
        _outFile = outFile;
    }

    public void messageLogged(ServerContext sc, LogMessage msg) {
        String user;

        try {
            if (msg.getType() != LogMessage.INFO) {
                // this class deals exclusively with info
                return;
            }

            user = msg.getUser();
            if (user == null) {
                user = "NULL";
            }
        }
    }
}

```

```
        _outFile.write(("Caught info"
            + " user=" + user
            + " text=" + msg.getText()
            + "\n").getBytes()
        );
        _outFile.flush();
    }
    catch(Exception e) {
        // Print some info output to the MobiLink log.
        e.printStackTrace();
    }
}
```

Der folgende Code registriert ein **InfoLogListener**-Objekt zum Empfang von Informationsmeldungen. Rufen Sie diesen Code dort auf, von wo aus Sie Zugriff auf den **ServerContext** haben, z.B. in einem Klassenkonstruktor oder einem Synchronisationsskript.

```
ServerContext serv_context;
FileOutputStream outFile
serv_context.addInfoListener(new InfoLogListener(outFile));
```

## WARNING-Variable

Zeigt an, dass die Logmeldung eine Warnung ist.

### Syntax

```
final int LogMessage.WARNING
```

### Siehe auch

- [LogMessage-Klasse \[MobiLink-Server Java\] auf Seite 496](#)

### Beispiel

Der folgende Code installiert ein **LogListener**-Objekt für alle Warnungen und schreibt dann die Informationen in eine Datei.

```
class WarningLogListener implements LogListener {
    FileOutputStream _outFile;

    public WarningLogListener(FileOutputStream outFile) {
        _outFile = outFile;
    }

    public void messageLogged(ServerContext sc, LogMessage msg) {
        String user;

        try {
            if (msg.getType() != LogMessage.WARNING) {
                // This class deals exclusively with warnings.
                return;
            }
            user = msg.getUser();

            if (user == null) {
                user = "NULL";
            }
        }
    }
}
```

```

        _outFile.write(("Caught warning"
            + " user=" + user
            + " text=" + msg.getText()
            + "\n").getBytes()
        );
        _outFile.flush();
    }
    catch(Exception e) {
        // Print some warning output to the MobiLink log.
        e.printStackTrace();
    }
}
}

```

Der folgende Code registriert ein **WarningLogListener**-Objekt zum Empfang von Warnmeldungen. Rufen Sie diesen Code dort auf, von wo aus Sie Zugriff auf den **ServerContext** haben, z.B. in einem Klassenkonstruktor oder einem Synchronisationsskript.

```

ServerContext serv_context;
FileOutputStream outFile
serv_context.addWarningListener(new WarningLogListener(outFile));

```

## NetworkData-Schnittstelle

Enthält Informationen über die Netzwerkdatenströme für eine Synchronisation.

### Syntax

```
public interface NetworkData
```

### Mitglieder

Alle Mitglieder der NetworkData-Schnittstelle, einschließlich aller geerbten Mitglieder.

| Name                                        | Beschreibung                                                                                             |
|---------------------------------------------|----------------------------------------------------------------------------------------------------------|
| <a href="#">getCertificateChain-Methode</a> | Gibt ein java.security.cert.CertPath-Objekt zurück, das alle vom Client geschickten Zertifikate enthält. |
| <a href="#">getHTTPHeaders-Methode</a>      | Gibt eine Zuordnungsobjekt zurück, das Header-Namen einer Liste von Header-Werten zuordnet.              |
| <a href="#">getHTTPHeaderValue-Methode</a>  | Gibt den Wert des letzten vom Server empfangenen Headers mit dem angegebenen Namen zurück.               |
| <a href="#">getHTTPHeaderValues-Methode</a> | Gibt alle vom Server empfangenen Header-Werte zurück, die mit dem angegebenen Namen verknüpft sind.      |
| <a href="#">isEndToEndEncrypted-Methode</a> | Ermittelt, ob diese Synchronisation eine Punkt-zu-Punkt-Verschlüsselung hat.                             |
| <a href="#">isHTTP-Methode</a>              | Ermittelt, ob die Synchronisation HTTP oder HTTPS verwendet.                                             |

| Name                          | Beschreibung                                       |
|-------------------------------|----------------------------------------------------|
| <a href="#">isTLS-Methode</a> | Ermittelt, ob diese Synchronisation TLS verwendet. |

## Bemerkungen

Diese Schnittstelle ist nützlich, wenn die Benutzerauthentifizierung bei einem anderen Server im Unternehmen erfolgt, der das clientseitige Zertifikat und HTTP-Header verwendet.

Wenn Sie das Sammeln von Netzwerk-Datenstromdaten aktivieren möchten, fügen Sie den Parametern vom Typ -x den Wert `collect_network_data=1` hinzu. Dadurch wird für jede Synchronisation zusätzlicher Speicher-Overhead zum Speichern der Daten erforderlich. Wenn Sie TLS oder HTTPS mit clientseitigen Zertifikaten verwenden, fügen Sie `trusted_certificates=<Zertifikatdatei>=""` hinzu, damit der Server den Client während des TLS-Handshakes auffordert, ein Zertifikat zu senden. Dies ist jedoch mit Zeitaufwand und Netzwerkkosten verbunden.

Sie erhalten ein `NetworkData`-Objekt durch Aufruf der `getNetworkData`-Methode der `DBConnectionContext`-Klasse. Wenn Sie HTTP oder HTTPS verwenden, enthält es die Header-Daten für die letzte HTTP-Anforderung, die vom Server vor dem Aufruf der Authentifizierungsskripten empfangen wurden.

## Siehe auch

- [DBConnectionContext-Schnittstelle \[MobiLink-Server Java\] auf Seite 477](#)

## Beispiel

Das folgende Beispiel zeigt, wie ein `NetworkData`-Objekt vom `DBConnectionContext`-Objekt abgerufen und die Daten ausgegeben werden.

```
public class OrderProcessor {
    DBConnectionContext _cc;

    public OrderProcessor( DBConnectionContext cc ) {
        _cc = cc;
    }

    // The method used for the authenticate_user event.
    public void AuthUser() {
        NetworkData nd = _cc.getNetworkData();
        if( nd != null ) {
            if( nd.isHTTP() ) {
                System.out.println( "http" );
                String user_agent = nd.getHeaderValue( "user-agent" );
                System.out.println( "  user-agent: " +
user_agent.substring( 0, user_agent.indexOf( '/' ) ) );
            } else {
                System.out.println( "no http" );
            }
            if( nd.isTLS() ) {
                System.out.println( "tls" );
                CertPath certs = nd.getCertificateChain();
                if( certs != null ) {
                    System.out.println( "  client-side cert:" );
                    int n = 1;
                    for( Certificate c : certs.getCertificates() ) {
                        System.out.println( "    cert " + n++ );
                    }
                }
            }
        }
    }
}
```

```

        X509Certificate c509 = (X509Certificate) c;
        System.out.println( "      Subject: " +
c509.getSubjectX500Principal().getName() );
        System.out.println( "      Issuer: " +
c509.getIssuerX500Principal().getName() );
    }
    } else {
        System.out.println( "    no client cert" );
    }
    } else {
        System.out.println( "no tls" );
    }
    if( nd.isEndToEndEncrypted() ) {
        System.out.println( "e2ee" );
    } else {
        System.out.println( "no e2ee" );
    }
    } else {
        System.out.println( "NULL networkdata" );
    }
    }
}

```

## getCertificateChain-Methode

Gibt ein `java.security.cert.CertPath`-Objekt zurück, das alle vom Client geschickten Zertifikate enthält.

### Syntax

```
java.security.cert.CertPath NetworkData.getCertificateChain()
```

### Rückgabe

Ein `CertPath`, das die X.509-Zertifikate enthält, die den Client identifizieren; gibt NULL zurück, wenn keine solchen Zertifikate enthalten waren.

### Bemerkungen

Die Zertifikate sind alle `java.security.cert.X509Certificate`-Objekte.

Diese Methode gibt nur dann einen anderen Wert als NULL zurück, wenn die `isTLS`-Methode TRUE zurückgibt, der Client mit dem `identity`-Datenstromparameter ein Zertifikat liefert und die `trusted_certificates`-Option auf dem Server gesetzt ist. Ein `CertPath`-Wert, der nicht NULL ist, enthält die Zertifikate, sortiert vom selbstsignierten Zertifikat zum Peer-Zertifikat.

### Siehe auch

- [trusted\\_certificates-Beispiel für Java auf Seite 101](#)

## getHTTPHeaders-Methode

Gibt eine Zuordnungsobjekt zurück, das Header-Namen einer Liste von Header-Werten zuordnet.

### Syntax

```
Map< String, List< String > > NetworkData.getHTTPHeaders()
```

### Rückgabe

Eine Zuordnung, die alle vom Server empfangenen Header enthält.

### Siehe auch

- [NetworkData.getHTTPHeaderValue-Methode \[MobiLink-Server Java\] auf Seite 504](#)
- [NetworkData.getHTTPHeaderValues-Methode \[MobiLink-Server Java\] auf Seite 504](#)

## getHTTPHeaderValue-Methode

Gibt den Wert des letzten vom Server empfangenen Headers mit dem angegebenen Namen zurück.

### Syntax

```
String NetworkData.getHTTPHeaderValue(String name)
```

### Parameter

- **name** Der Headername, für den der Wert zurückgegeben werden soll.

### Rückgabe

Der letzte Header-Wert, der dem angegebenen Header-Namen zugeordnet ist.

### Siehe auch

- [NetworkData.getHTTPHeaderValues-Methode \[MobiLink-Server Java\] auf Seite 504](#)
- [NetworkData.getHTTPHeaders Methode \[MobiLink-Server Java\] auf Seite 503](#)

## getHTTPHeaderValues-Methode

Gibt alle vom Server empfangenen Header-Werte zurück, die mit dem angegebenen Namen verknüpft sind.

### Syntax

```
List< String > NetworkData.getHTTPHeaderValues(String name)
```

### Parameter

- **name** Der Headername, für den die Werte zurückgegeben werden sollen.

### Rückgabe

Die Header-Werte, die dem angegebenen Header-Namen zugeordnet sind.

### Siehe auch

- [NetworkData.getHTTPHeaderValue-Methode \[MobiLink-Server Java\] auf Seite 504](#)
- [NetworkData.getHTTPHeaders Methode \[MobiLink-Server Java\] auf Seite 503](#)

## isEndToEndEncrypted-Methode

Ermittelt, ob diese Synchronisation eine Punkt-zu-Punkt-Verschlüsselung hat.

### Syntax

```
boolean NetworkData.isEndToEndEncrypted( )
```

### Rückgabe

TRUE, wenn diese Synchronisation die Punkt-zu-Punkt-Verschlüsselung verwendet, sonst FALSE.

## isHTTP-Methode

Ermittelt, ob die Synchronisation HTTP oder HTTPS verwendet.

### Syntax

```
boolean NetworkData.isHTTP( )
```

### Rückgabe

TRUE, wenn diese Synchronisation HTTP oder HTTPS verwendet, sonst FALSE.

## isTLS-Methode

Ermittelt, ob diese Synchronisation TLS verwendet.

### Syntax

```
boolean NetworkData.isTLS( )
```

### Rückgabe

TRUE, wenn diese Synchronisation TLS verwendet, sonst FALSE.

## ServerContext-Schnittstelle

Eine Instanziierung des gesamten Kontextes, der für die Dauer des MobiLink-Servers vorhanden ist.

### Syntax

```
public interface ServerContext
```

### Mitglieder

Alle Mitglieder der ServerContext-Schnittstelle, einschließlich aller geerbten Mitglieder.

| Name                                     | Beschreibung                                                                                                            |
|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| <a href="#">addErrorListener-Methode</a> | Fügt das angegebene LogListener-Objekt hinzu, das eine Benachrichtigung erhalten soll, wenn ein Fehler ausgegeben wird. |

| Name                                           | Beschreibung                                                                                                                                                                  |
|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">addInfoListener-Methode</a>        | Fügt das angegebene <b>LogListener</b> -Objekt der Liste der Listener-Objekte hinzu, die eine Benachrichtigung erhalten sollen, wenn eine Infomeldung ausgegeben wird.        |
| <a href="#">addShutdownListener-Methode</a>    | Fügt das angegebene <b>ShutdownListener</b> -Objekt hinzu, das die Benachrichtigung erhalten soll, bevor der Serverkontext zerstört wird.                                     |
| <a href="#">addWarningListener-Methode</a>     | Fügt das angegebene <b>LogListener</b> -Objekt hinzu, das eine Benachrichtigung erhalten soll, wenn eine Warnung ausgegeben wird.                                             |
| <a href="#">getProperties-Methode</a>          | Gibt die Eigenschaftengruppen für eine bestimmte Komponente und Eigenschaftengruppe zurück.                                                                                   |
| <a href="#">getPropertiesByVersion-Methode</a> | Gibt die Eigenschaftengruppe zurück, die der Skriptversion zugeordnet ist.                                                                                                    |
| <a href="#">getPropertySetNames-Methode</a>    | Gibt die Liste der Namen der Eigenschaftengruppen für eine bestimmte Komponente zurück:                                                                                       |
| <a href="#">getStartClassInstances-Methode</a> | Ruft ein Array der Startklassen ab, die beim Serverstart konstruiert wurden.                                                                                                  |
| <a href="#">makeConnection-Methode</a>         | Öffnet eine neue Serververbindung und gibt sie zurück.                                                                                                                        |
| <a href="#">removeErrorListener-Methode</a>    | Löscht das angegebene <b>LogListener</b> -Objekt aus der Liste der Listener-Objekte, die eine Benachrichtigung erhalten sollen, wenn ein Fehler ausgegeben wird.              |
| <a href="#">removeInfoListener-Methode</a>     | Löscht das angegebene <b>LogListener</b> -Objekt aus der Liste der Listener-Objekte, die eine Benachrichtigung erhalten sollen, wenn eine Infomeldung ausgegeben wird.        |
| <a href="#">removeShutdownListener-Methode</a> | Entfernt das angegebene <b>ShutdownListener</b> -Objekt aus der Liste der Listener-Objekte, die Benachrichtigungen erhalten, bevor dieser <b>ServerContext</b> zerstört wird. |
| <a href="#">removeWarningListener-Methode</a>  | Löscht das angegebene <b>LogListener</b> -Objekt aus der Liste der Listener-Objekte, die eine Benachrichtigung erhalten sollen, wenn eine Warnung ausgegeben wird.            |
| <a href="#">shutdown-Methode</a>               | Zwingt den Server, herunterzufahren.                                                                                                                                          |

## Bemerkungen

Dieser Kontext kann wie statische Daten gespeichert und in einem Hintergrund-Thread benutzt werden. Er ist für die Dauer der von MobiLink aufgerufenen Java VM gültig.

Zum Zugriff auf eine **ServerContext**-Instanz verwenden Sie die **DBConnectionContext.getServerContext**-Methode.

## addErrorListener-Methode

Fügt das angegebene **LogListener**-Objekt hinzu, das eine Benachrichtigung erhalten soll, wenn ein Fehler ausgegeben wird.

### Syntax

```
void ServerContext.addErrorListener(LogListener ll)
```

### Parameter

- **ll** Das **LogListener**-Objekt, das bei einem Fehler benachrichtigt werden soll.

### Bemerkungen

Wenn ein Fehler ausgegeben wird, wird die **LogListener.messageLogged(ServerContext, LogMessage)**-Methode aufgerufen.

### Siehe auch

- [LogListener.messageLogged-Methode \[MobiLink-Server Java\] auf Seite 496](#)
- [LogMessage-Klasse \[MobiLink-Server Java\] auf Seite 496](#)

## addInfoListener-Methode

Fügt das angegebene **LogListener**-Objekt der Liste der Listener-Objekte hinzu, die eine Benachrichtigung erhalten sollen, wenn eine Infomeldung ausgegeben wird.

### Syntax

```
void ServerContext.addInfoListener(LogListener ll)
```

### Parameter

- **ll** Das **LogListener**-Objekt, das bei einer Infomeldung benachrichtigt werden soll.

### Bemerkungen

Die **LogListener.messageLogged**-Methode wird aufgerufen.

### Siehe auch

- [LogListener.messageLogged-Methode \[MobiLink-Server Java\] auf Seite 496](#)

### Beispiel

Der folgende Code registriert ein **MyLogListener**-Objekt, das Benachrichtigungen über informative Meldungen erhalten soll:

```
// ServerContext serv_context;  
serv_context.addInfoListener(new MyLogListener(ll_out_file));
```

```
// The following code shows an example of processing those messages:
class MyLogListener implements LogListener {
    FileOutputStream _out_file;
    public TestLogListener(FileOutputStream out_file) {
        _out_file = out_file;
    }

    public void messageLogged(ServerContext sc, LogMessage msg) {
        String type;
        String user;

        try {
            if (msg.getType() == LogMessage.ERROR) {
                type = "ERROR";
            } else if (msg.getType() == LogMessage.WARNING) {
                type = "WARNING";
            } else if (msg.getType() == LogMessage.INFO) {
                type = "INFO";
            } else {
                type = "UNKNOWN!!!";
            }

            user = msg.getUser();
            if (user == null) {
                user = "NULL";
            }
            _out_file.write(("Caught msg type="
                + type
                + " user=" + user
                + " text=" +msg.getText()
                + "\n").getBytes()
            );
            _out_file.flush();
        }
        catch(Exception e) {

            // if we print the exception from processing an info message,
            // we may recurse indefinitely
            if (msg.getType() != LogMessage.ERROR) {
                // Print some error output to the MobiLink log.
                e.printStackTrace();
            }
        }
    }
}
```

## addShutdownListener-Methode

Fügt das angegebene **ShutdownListener**-Objekt hinzu, das die Benachrichtigung erhalten soll, bevor der Serverkontext zerstört wird.

### Syntax

```
void ServerContext.addShutdownListener(ShutdownListener sl)
```

### Parameter

- **sl** Das ShutdownListener-Objekt, das bei einem Shutdown benachrichtigt werden soll.

## Bemerkungen

Beim Herunterfahren wird die **ShutdownListener.shutdownPerformed(ServerContext)**-Methode aufgerufen.

## Siehe auch

- [ShutdownListener-Schnittstelle \[MobiLink-Server Java\] auf Seite 516](#)
- [ShutdownListener.shutdownPerformed-Methode \[MobiLink-Server Java\] auf Seite 517](#)

## addWarningListener-Methode

Fügt das angegebene **LogListener**-Objekt hinzu, das eine Benachrichtigung erhalten soll, wenn eine Warnung ausgegeben wird.

## Syntax

```
void ServerContext.addWarningListener(LogListener ll)
```

## Parameter

- **ll** Das LogListener-Objekt, das bei einer Warnung benachrichtigt werden soll.

## Bemerkungen

Die **LogListener.messageLogged(ServerContext, LogMessage)**-Methode wird aufgerufen.

## Siehe auch

- [LogMessage-Klasse \[MobiLink-Server Java\] auf Seite 496](#)
- [LogListener.messageLogged-Methode \[MobiLink-Server Java\] auf Seite 496](#)

## getProperties-Methode

Gibt die Eigenschaftengruppen für eine bestimmte Komponente und Eigenschaftengruppe zurück.

## Syntax

```
Properties ServerContext.getProperties(String component, String set)
```

## Parameter

- **component** Die Komponente.
- **group** Die Eigenschaftengruppe.

## Rückgabe

Die Eigenschaftengruppe, die möglicherweise leer ist.

## Bemerkungen

Diese Eigenschaften werden in der **ml\_property**-Systemtabelle gespeichert. Weitere Hinweise finden Sie unter „**ml\_add\_property**-Systemprozedur“ auf Seite 664.

### Beispiel

Das folgende Beispiel zeigt, wie Sie eine Liste aller Eigenschaften einer **ServerContext**-Instanz anzeigen.

```
import java.util.*;
ServerContext serverContext;
PrintStream out
Properties prop = serverContext.getProperties();
prop.list(out);
```

## getPropertiesByVersion-Methode

Gibt die Eigenschaftengruppe zurück, die der Skriptversion zugeordnet ist.

### Syntax

```
Properties ServerContext.getPropertiesByVersion(String script_version)
```

### Parameter

- **script\_version** Die Skriptversion, für die zugehörige Eigenschaften zurückgegeben werden sollen.

### Rückgabe

Gibt die Eigenschaftengruppe zurück, die der angegebenen Skriptversion zugeordnet ist.

### Bemerkungen

Diese werden in der **ml\_property**-Systemtabelle gespeichert. Die Skriptversion ist in der **property\_set\_name**-Spalte gespeichert, wenn **component\_name** **ScriptVersion** ist. Weitere Hinweise finden Sie unter „[ml\\_add\\_property-Systemprozedur](#)“ auf Seite 664.

### Beispiel

Das folgende Beispiel zeigt, wie Sie eine Liste aller Eigenschaften einer **ServerContext**-Instanz anzeigen, die einer bestimmten Skriptversion zugeordnet ist:

```
import java.util.*;
ServerContext serverContext;
PrintStream out
Properties prop = serverContext.getPropertiesByVersion("MyScriptVersion");
prop.list(out);
```

## getPropertySetNames-Methode

Gibt die Liste der Namen der Eigenschaftengruppen für eine bestimmte Komponente zurück:

### Syntax

```
Iterator ServerContext.getPropertySetNames(String component)
```

### Parameter

- **component** Der Name der Komponente, für die Eigenschaftennamen aufgeführt werden sollen.

## Rückgabe

Die Liste der Namen der Eigenschaftengruppen für eine bestimmte Komponente.

## Bemerkungen

Diese Eigenschaften werden in der **ml\_property**-Systemtabelle gespeichert. Weitere Hinweise finden Sie unter „[ml\\_add\\_property-Systemprozedur](#)“ auf Seite 664.

## Beispiel

Das folgende Beispiel zeigt, wie Sie eine Liste aller Eigenschaften einer **ServerContext**-Instanz anzeigen, die einer bestimmten Komponente zugeordnet ist:

```
import java.util.*;
ServerContext serverContext;
PrintStream out
Properties prop = serverContext.getPropertySetNames("Component Name");
prop.list(out);
```

## getStartClassInstances-Methode

Ruft ein Array der Startklassen ab, die beim Serverstart konstruiert wurden.

### Syntax

```
Object[] ServerContext.getStartClassInstances()
```

## Rückgabe

Ein Array von Startklassen, die beim Serverstart erstellt wurden, oder ein Array mit der Länge Null, wenn keine Startklassen vorhanden sind.

## Bemerkungen

Weitere Hinweise finden Sie unter „[Benutzerdefinierte Startklassen](#)“ auf Seite 471.

## Beispiel

Das folgende Beispiel zeigt die Verwendung der **getStartClassInstances**- Methode:

```
Object objs[] = sc.getStartClassInstances();
int i;
for (i=0; i < objs.length; i += 1) {
    if (objs[i] instanceof MyClass) {
        // Use class.
    }
}
```

## makeConnection-Methode

Öffnet eine neue Serververbindung und gibt sie zurück.

### Syntax

```
java.sql.Connection ServerContext.makeConnection() throws SQLException
```

## Rückgabe

Die neu erstellte Serververbindung.

## Ausnahmen

- **java.sql.SQLException** Wird ausgegeben, wenn beim Öffnen einer neuen Verbindung ein Fehler aufgetreten ist.

## Bemerkungen

Der Eigentümer dieser Verbindung ist der Benutzer des Java-Codes. Die Verbindung muss vom Benutzer festgeschrieben und geschlossen werden.

Um auf den Serverkontext zuzugreifen, wenden Sie die **DBConnectionContext.getServerContext-** Methode auf **DBConnectionContext** für die aktuelle Verbindung an.

### Hinweis

Das Öffnen einer Verbindung kann kostenträchtig sein. Schreiben Sie die Logik so, dass die Anzahl von Aufrufen dieser Methode minimiert wird.

## removeErrorListener-Methode

Löscht das angegebene **LogListener**-Objekt aus der Liste der Listener-Objekte, die eine Benachrichtigung erhalten sollen, wenn ein Fehler ausgegeben wird.

### Syntax

```
void ServerContext.removeErrorListener(LogListener ll)
```

### Parameter

- **ll** Das Listener-Objekt soll keine Benachrichtigen mehr erhalten.

### Beispiel

Der folgende Code entfernt ein **LogListener**-Objekt aus der Liste der Fehler-Listener-Objekte.

```
ServerContext serverContext;  
LogListener myErrorListener  
serverContext.removeErrorListener(myErrorListener);
```

## removeInfoListener-Methode

Löscht das angegebene **LogListener**-Objekt aus der Liste der Listener-Objekte, die eine Benachrichtigung erhalten sollen, wenn eine Infomeldung ausgegeben wird.

### Syntax

```
void ServerContext.removeInfoListener(LogListener ll)
```

**Parameter**

- **ll** Das Listener-Objekt soll keine Benachrichtigen mehr erhalten.

**Beispiel**

Der folgende Code entfernt ein **LogListener**-Objekt aus der Liste der Informations-Listener-Objekte:

```
ServerContext serverContext;  
LogListener myInfoListener  
serverContext.removeInfoListener(myInfoListener);
```

## removeShutdownListener-Methode

Entfernt das angegebene **ShutdownListener**-Objekt aus der Liste der Listener-Objekt, die Benachrichtigungen erhalten, bevor dieser **ServerContext** zerstört wird.

**Syntax**

```
void ServerContext.removeShutdownListener(ShutdownListener sl)
```

**Parameter**

- **sl** Das Listener-Objekt soll keine Benachrichtigen mehr erhalten.

**Beispiel**

Der folgende Code löscht ein **ShutdownListener**-Objekt aus der Liste der Listener-Objekte, die eine Benachrichtigung erhalten sollen, bevor der **ServerContext** zerstört wird:

```
ServerContext serverContext;  
ShutdownListener myShutdownListener  
serverContext.removeShutdownListener(myShutdownListener);
```

## removeWarningListener-Methode

Löscht das angegebene **LogListener**-Objekt aus der Liste der Listener-Objekte, die eine Benachrichtigung erhalten sollen, wenn eine Warnung ausgegeben wird.

**Syntax**

```
void ServerContext.removeWarningListener(LogListener ll)
```

**Parameter**

- **ll** Das Listener-Objekt soll keine Benachrichtigen mehr erhalten.

**Beispiel**

Der folgende Code entfernt ein **LogListener**-Objekt aus der Liste der Warnungs-Listener-Objekte.

```
ServerContext serverContext;  
LogListener myWarningListener  
serverContext.removeWarningListener(myWarningListener);
```

# shutdown-Methode

Zwingt den Server, herunterzufahren.

## Syntax

```
void ServerContext.shutdown()
```

## Bemerkungen

Für registrierte **ShutdownListener**-Objekte wird die **shutdownPerformed**-Methode aufgerufen.

## Siehe auch

- [ShutdownListener.shutdownPerformed-Methode \[MobiLink-Server Java\] auf Seite 517](#)

## Beispiel

Der folgende Code zwingt den Server herunterzufahren:

```
ServerContext serverContext;  
serverContext.shutdown();
```

# ServerException-Klasse

Wird ausgegeben, wenn eine Fehlerbedingung eingetreten ist, die die Fortsetzung der Synchronisation auf dem Server nicht zulässt.

## Syntax

```
public class ServerException
```

## Mitglieder

Alle Mitglieder der **ServerException**-Klasse, einschließlich aller geerbten Mitglieder.

| Name                                        | Beschreibung                           |
|---------------------------------------------|----------------------------------------|
| <a href="#">ServerException-Konstruktor</a> | Erstellt eine <b>ServerException</b> . |

## Bemerkungen

Die Meldung dieser Ausnahme führt dazu, dass der MobiLink-Server heruntergefahren wird.

## Beispiel

Das folgende Beispiel zeigt, wie Sie eine **ServerException** ausgeben, wenn ein schwerwiegendes Problem auftritt, und den MobiLink-Server herunterfahren:

```
public void handleUpload(UploadData ud)  
    throws SQLException, IOException, ServerException  
{  
  
    UploadedTableData tables[] = ud.getUploadedTables();  
    if (tables == null) {  
        throw new ServerException("Failed to read uploaded tables");  
    }  
}
```

```

    }

    for (int i = 0; i < tables.length; i++) {
        UploadedTableData currentTable = tables[i];
        println("table " + java.lang.Integer.toString(i)
            + " name: " + currentTable.getName());

        // Print out delete result set.
        println("Deletes");
        printRSInfo(currentTable.getDeletes());

        // Print out insert result set.
        println("Inserts");
        printRSInfo(currentTable.getInserts());

        // print out update result set
        println("Updates");
        printUpdateRSInfo(currentTable.getUpdates());
    }
}

```

## ServerException-Konstruktor

Konstruiert eine `ServerException`.

### Überladungsliste

| Name                                                | Beschreibung                                                                       |
|-----------------------------------------------------|------------------------------------------------------------------------------------|
| <a href="#">ServerException()-Konstruktor</a>       | Konstruiert eine <b>ServerException</b> ohne detaillierte Meldung.                 |
| <a href="#">ServerException(String)-Konstruktor</a> | Konstruiert eine <b>ServerException</b> mit der angegebenen detaillierten Meldung. |

## ServerException()-Konstruktor

Konstruiert eine **ServerException** ohne detaillierte Meldung.

### Syntax

```
ServerException.ServerException()
```

## ServerException(String)-Konstruktor

Konstruiert eine **ServerException** mit der angegebenen detaillierten Meldung.

### Syntax

```
ServerException.ServerException(String s)
```

### Parameter

- **s** Eine ausführliche Meldung.

# ShutdownListener-Schnittstelle

Die Listener-Schnittstelle für das Erfassen des Herunterfahrens des Servers.

## Syntax

```
public interface ShutdownListener
```

## Mitglieder

Alle Mitglieder der ShutdownListener-Schnittstelle, einschließlich aller geerbten Mitglieder.

| Name                                      | Beschreibung                                                                                        |
|-------------------------------------------|-----------------------------------------------------------------------------------------------------|
| <a href="#">shutdownPerformed-Methode</a> | Wird aufgerufen, bevor der <b>ServerContext</b> durch das Herunterfahren des Servers zerstört wird. |

## Bemerkungen

Mit dieser Schnittstelle können Sie sicherstellen, dass alle Threads, Verbindungen und anderen Ressourcen bereinigt werden, bevor der MobiLink-Server gestoppt wird.

## Beispiel

Das folgende Beispiel zeigt, wie Sie ein **ShutdownListener**-Objekt für die **ServerContext**-Instanz installieren:

```
class MyShutdownListener implements ShutdownListener {
    FileOutputStream _outFile;
    public MyShutdownListener(FileOutputStream outFile) {
        _outFile = outFile;
    }

    public void shutdownPerformed(ServerContext sc) {
        // Add shutdown code
        try {
            _outFile.write(("Shutting Down" + "\n").getBytes());
            _outFile.flush();
        }
        catch(Exception e) {
            // Print some error output to the MobiLink log.
            e.printStackTrace();
        }
        // ...
    }
}
```

Der folgende Code registriert ein **MyShutdownListener**-Objekt. Rufen Sie diesen Code dort auf, von wo aus Sie Zugriff auf den **ServerContext** haben, z.B. in einem Klassenkonstruktor oder einem Synchronisationsskript.

```
ServerContext serv_context;
FileOutputStream outFile;
serv_context.addShutdownListener(new MyShutdownListener(outFile));
```

## shutdownPerformed-Methode

Wird aufgerufen, bevor der **ServerContext** durch das Herunterfahren des Servers zerstört wird.

### Syntax

```
void ShutdownListener.shutdownPerformed(ServerContext sc)
```

### Parameter

- **sc** Der Kontext für den Server, der heruntergefahren wird.

## SpatialUtilities-Klasse

Eine Sammlung von statischen Methoden für die Arbeit mit räumlichen Werten.

### Syntax

```
public class SpatialUtilities
```

### Mitglieder

Alle Mitglieder der SpatialUtilities-Klasse, einschließlich aller geerbten Mitglieder.

| Name                                       | Beschreibung                                                                                                                         |
|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">CreateSpatialValue-Methode</a> | Gibt ein neues Byte-Array mit einem für den Download formatierten räumlichen Wert zurück.                                            |
| <a href="#">getBytes-Methode</a>           | Gibt ein neues Byte-Array mit denselben räumlichen Daten wie das angegebene Byte-Array zurück, wobei jedoch die SRID entfernt wurde. |
| <a href="#">getSRID-Methode</a>            | Gibt die SRID für den angegebenen räumlichen Wert zurück.                                                                            |
| <a href="#">setSRID-Methode</a>            | Speichert die angegebene SRID in den ersten vier Byte des angegebenen Byte-Arrays.                                                   |

## CreateSpatialValue-Methode

Gibt ein neues Byte-Array mit einem für den Download formatierten räumlichen Wert zurück.

### Syntax

```
byte[] SpatialUtilities.createSpatialValue(
    int srid,
    byte[] spatial_value
)
```

### Parameter

- **srid** Die SRID.

- **spatial\_value** Die räumlichen Daten.

### Rückgabe

Der für den Download formatierte räumliche Wert.

### Bemerkungen

Die ersten vier Byte enthalten die angegebene SRID im Little-Endian-System, der Rest sind die räumlichen Daten, die im angegebenen Byte-Array weitergegeben werden.

## getBytes-Methode

Gibt ein neues Byte-Array mit denselben räumlichen Daten wie das angegebene Byte-Array zurück, wobei jedoch die SRID entfernt wurde.

### Syntax

```
byte[] SpatialUtilities.getBytes(byte[] spatial_value)
```

### Parameter

- **spatial\_value** Ein räumlicher Wert, dessen SRID entfernt werden muss.

### Rückgabe

Das neue Byte-Array.

## getSRID-Methode

Gibt die SRID für den angegebenen räumlichen Wert zurück.

### Syntax

```
int SpatialUtilities.getSRID(byte[] spatial_value)
```

### Parameter

- **spatial\_value** Der hochgeladene Wert. Die ersten vier Byte müssen die angegebene SRID in Little-Endian enthalten.

### Rückgabe

Die SRID.

## setSRID-Methode

Speichert die angegebene SRID in den ersten vier Byte des angegebenen Byte-Arrays.

### Syntax

```
void SpatialUtilities.setSRID(byte[] spatial_value, int srid)
```

### Parameter

- **spatial\_value** Das Array, in dem die SRID gespeichert werden soll.
- **srid** Die zu speichernde SRID.

## SynchronizationException-Klasse

Wird ausgegeben, um darauf hinzuweisen, dass eine Fehlerbedingung eingetreten ist, die den Abschluss der aktuellen Synchronisation nicht zulässt.

### Syntax

```
public class SynchronizationException
```

### Mitglieder

Alle Mitglieder der SynchronizationException-Klasse, einschließlich aller geerbten Mitglieder.

| Name                                                 | Beschreibung                                    |
|------------------------------------------------------|-------------------------------------------------|
| <a href="#">SynchronizationException-Konstruktor</a> | Erstellt eine <b>SynchronizationException</b> . |

### Bemerkungen

Das Auftreten dieser Ausnahmebedingung bewirkt, dass der MobiLink-Server die aktuelle Synchronisation zurücksetzen und die Verbindung neu initialisieren muss.

### Beispiel

Das folgende Beispiel zeigt, wie Sie eine **SynchronizationException** ausgeben, wenn ein Problem auftritt, das dazu führt, dass der MobiLink-Server zurückgesetzt wird.

```
public void handleUpload(UploadData ud)
    throws SQLException, IOException, SynchronizationException
{
    UploadedTableData tables[] = ud.getUploadedTables();

    for (int i = 0; i < tables.length; i++) {
        UploadedTableData currentTable = tables[i];
        println("table " + java.lang.Integer.toString(i)
            + " name: " + currentTable.getName());

        // Print out delete result set.
        println("Deletes");
        printRSInfo(currentTable.getDeletes());

        // Print out insert result set.
        println("Inserts");
        printRSInfo(currentTable.getInserts());

        // print out update result set
        println("Updates");
        printUpdaterSInfo(currentTable.getUpdates());

        if (/* Reason for Sync failure */) {
```

```
        }  
        }  
        }  
        throw new SynchronizationException("Sync Failed");  
    }  
}
```

# SynchronizationException-Konstruktor

Konstruiert eine SynchronizationException.

## Überladungsliste

| Name                                                         | Beschreibung                                                                                |
|--------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <a href="#">SynchronizationException()-Konstruktor</a>       | Konstruiert eine <b>SynchronizationException</b> ohne detaillierte Meldung.                 |
| <a href="#">SynchronizationException(String)-Konstruktor</a> | Konstruiert eine <b>SynchronizationException</b> mit der angegebenen detaillierten Meldung. |

## SynchronizationException()-Konstruktor

Konstruiert eine **SynchronizationException** ohne detaillierte Meldung.

### Syntax

```
SynchronizationException.SynchronizationException()
```

## SynchronizationException(String)-Konstruktor

Konstruiert eine **SynchronizationException** mit der angegebenen detaillierten Meldung.

### Syntax

```
SynchronizationException.SynchronizationException(String s)
```

### Parameter

- **s** Eine ausführliche Meldung.

# TimestampWithTimeZone-Klasse

Ein **java.sql.Timestamp** mit Methoden zum Abrufen und Festlegen der Zeitzone.

### Syntax

```
public class TimestampWithTimeZone
```

### Basisklassen

- Timestamp

## Mitglieder

Alle Mitglieder der `TimestampWithTimeZone`-Klasse, einschließlich aller geerbten Mitglieder.

| Name                                              | Beschreibung                                                                                                                                                                                                                                                                        |
|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">TimestampWithTimeZone-Konstruktor</a> | Erstellt einen neuen <b>TimestampWithTimeZone</b> mit den angegebenen Werten für Jahr, Monat, Tag, Stunde, Minute, Sekunde, Nanosekunde, Zeitzone-Stunde und Zeitzone-Minute.                                                                                                       |
| <a href="#">equals-Methode</a>                    | Gibt TRUE zurück, wenn <b>o</b> mit dem Timestamp-Teil dieses Objekts übereinstimmt und entweder <b>o</b> ein <b>TimestampWithTimeZone</b> mit demselben Zeitzone-Offset ist oder <b>o</b> kein <b>TimestampWithTimeZone</b> ist dieses Objekt einen Zeitzone-Offset von 00:00 hat. |
| <a href="#">getTimeZoneOffsetHours-Methode</a>    | Ruft den Stundenteil des Zeitzone-Offsets ab.                                                                                                                                                                                                                                       |
| <a href="#">getTimeZoneOffsetMinutes-Methode</a>  | Ruft den Minutenteil des Zeitzone-Offsets ab.                                                                                                                                                                                                                                       |
| <a href="#">setTimeZoneOffsetHours-Methode</a>    | Legt den Stundenteil des Zeitzone-Offsets fest.                                                                                                                                                                                                                                     |
| <a href="#">setTimeZoneOffsetMinutes-Methode</a>  | Legt den Minutenteil des Zeitzone-Offsets fest.                                                                                                                                                                                                                                     |
| <a href="#">toString-Methode</a>                  | Gibt die Zeichenfolge für diesen Zeitstempel im Format <b>yyyy-mm-dd hh:mm:ss.ffffff Shh:mm</b> zurück, wobei <b>S</b> für das Vorzeichen des Stundenfelds steht.                                                                                                                   |
| <a href="#">toTimestampWithTimeZone-Methode</a>   | Konvertiert den angegebenen <b>Timestamp</b> in einen <b>TimestampWithTimeZone</b> .                                                                                                                                                                                                |
| <a href="#">valueOf-Methode</a>                   | Konvertiert eine Zeichenfolge in einen <b>TimestampWithTimeZone</b> -Wert.                                                                                                                                                                                                          |

## Bemerkungen

Verwenden Sie diese Klasse, wenn Sie den Zeitzone-Offset für `TIMESTAMP WITH TIME ZONE`-Spalten unter Verwendung der MobiLink-API für die direkte Zeilenbehandlung angeben.

**PreparedStatement**- und **ResultSet**-Objekte aus anderen JDBC-Treibern als der MobiLink-direct row-API behandeln diese Klasse als normalen **Timestamp**.

## TimestampWithTimeZone-Konstruktor

Erstellt einen neuen **TimestampWithTimeZone** mit den angegebenen Werten für Jahr, Monat, Tag, Stunde, Minute, Sekunde, Nanosekunde, Zeitzone-Stunde und Zeitzone-Minute.

## Syntax

```
TimestampWithTimeZone.TimestampWithTimeZone(  
    int year,  
    int month,  
    int date,  
    int hour,  
    int minute,  
    int second,  
    int nano,  
    int tz_hour,  
    int tz_minute  
)
```

## Parameter

- **year** Das Jahr minus 1900.
- **month** Eine Ganzzahl zwischen 0 und 11.
- **date** Eine Ganzzahl zwischen 1 und 31.
- **hour** Eine Ganzzahl zwischen 0 und 23.
- **minute** Eine Ganzzahl zwischen 0 und 59.
- **second** Eine Ganzzahl zwischen 0 und 59.
- **nano** Eine Ganzzahl zwischen 0 und 999.999.999.
- **tz\_hour** Eine Ganzzahl zwischen -14 und 14.
- **tz\_minute** Eine Ganzzahl zwischen 0 und 59.

## Ausnahmebedingungen

- **java.lang.IllegalArgumentException** Wird ausgelöst, wenn **tz\_minute** sich nicht im Bereich 0-59 oder **tz\_hour** sich nicht im entsprechenden Bereich befindet.

## equals-Methode

Gibt TRUE zurück, wenn ein Vergleichsobjekt mit dem Timestamp-Teil dieses Objekts übereinstimmt und entweder das Vergleichsobjekt ein TimestampWithTimeZone-Wert mit demselben Zeitzone-Offset ist oder das Vergleichsobjekt kein TimestampWithTimeZone-Wert ist und dieses Objekt einen Zeitzone-Offset von 00:00 hat.

## Überladungsliste

| Name                                      | Beschreibung                                                                                                                                                                                                                                                                        |
|-------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">equals(Object)-Methode</a>    | Gibt TRUE zurück, wenn <b>o</b> mit dem Timestamp-Teil dieses Objekts übereinstimmt und entweder <b>o</b> ein <b>TimestampWithTimeZone</b> mit demselben Zeitzone-Offset ist oder <b>o</b> kein <b>TimestampWithTimeZone</b> ist dieses Objekt einen Zeitzone-Offset von 00:00 hat. |
| <a href="#">equals(Timestamp)-Methode</a> | Gibt TRUE zurück, wenn <b>o</b> mit dem Timestamp-Teil dieses Objekts übereinstimmt und entweder <b>o</b> ein <b>TimestampWithTimeZone</b> mit demselben Zeitzone-Offset ist oder <b>o</b> kein <b>TimestampWithTimeZone</b> ist dieses Objekt einen Zeitzone-Offset von 00:00 hat. |

### equals(Object)-Methode

Gibt TRUE zurück, wenn **o** mit dem Timestamp-Teil dieses Objekts übereinstimmt und entweder **o** ein **TimestampWithTimeZone** mit demselben Zeitzone-Offset ist oder **o** kein **TimestampWithTimeZone** ist dieses Objekt einen Zeitzone-Offset von 00:00 hat.

#### Syntax

```
boolean TimestampWithTimeZone.equals(Object o)
```

#### Parameter

- **o** Das Objekt zum Vergleich.

#### Rückgabe

TRUE, wenn **o** gleich diesem Objekt ist, sonst FALSE.

#### Siehe auch

- [TimestampWithTimeZone.equals-Methode \[MobiLink-Server Java\] auf Seite 522](#)

### equals(Timestamp)-Methode

Gibt TRUE zurück, wenn **o** mit dem Timestamp-Teil dieses Objekts übereinstimmt und entweder **o** ein **TimestampWithTimeZone** mit demselben Zeitzone-Offset ist oder **o** kein **TimestampWithTimeZone** ist dieses Objekt einen Zeitzone-Offset von 00:00 hat.

#### Syntax

```
boolean TimestampWithTimeZone.equals(Timestamp o)
```

#### Parameter

- **o** Das Objekt zum Vergleich.

## Rückgabe

TRUE, wenn o gleich diesem Objekt ist, sonst FALSE.

## Siehe auch

- [TimestampWithTimeZone.equals-Methode \[MobiLink-Server Java\] auf Seite 522](#)

## getTimeZoneOffsetHours-Methode

Ruft den Stundenteil des Zeitzonen-Offsets ab.

### Syntax

```
int TimestampWithTimeZone.getTimeZoneOffsetHours( )
```

## Rückgabe

Der Stundenteil des Zeitzonen-Offset.

## getTimeZoneOffsetMinutes-Methode

Ruft den Minutenteil des Zeitzonen-Offsets ab.

### Syntax

```
int TimestampWithTimeZone.getTimeZoneOffsetMinutes( )
```

## Rückgabe

Der Minutenteil des Zeitzonen-Offset.

## setTimeZoneOffsetHours-Methode

Legt den Stundenteil des Zeitzonen-Offsets fest.

### Syntax

```
void TimestampWithTimeZone.setTimeZoneOffsetHours(int tz_hour)
```

### Parameter

- **tz\_hour** Der neue Stundenteil des Zeitzonen-Offset.

### Ausnahmebedingungen

- **java.lang.IllegalArgumentException** Wird ausgelöst, wenn **tz\_hour** sich nicht im Bereich -14 - 14 befindet.

## setTimeZoneOffsetMinutes-Methode

Legt den Minutenteil des Zeitzonen-Offsets fest.

**Syntax**

```
void TimestampWithTimeZone.setTimeZoneOffsetMinutes(int tz_minute)
```

**Parameter**

- **tz\_minute** Der neue Minutenteil des Zeitzonen-Offset.

**Ausnahmebedingungen**

- **java.lang.IllegalArgumentException** Wird ausgelöst, wenn **tz\_minute** sich nicht im Bereich 0-59 befindet.

## toString-Methode

Gibt die Zeichenfolge für diesen Zeitstempel im Format **yyyy-mm-dd hh:mm:ss.ffffff Shh:mm** zurück, wobei **S** für das Vorzeichen des Stundenfelds steht.

**Syntax**

```
String TimestampWithTimeZone.toString()
```

**Rückgabe**

Die Zeichenfolge für diesen Zeitstempel.

## toTimestampWithTimeZone-Methode

Konvertiert den angegebenen **Timestamp** in einen **TimestampWithTimeZone**.

**Syntax**

```
TimestampWithTimeZone TimestampWithTimeZone.toTimestampWithTimeZone(  
    Timestamp ts  
)
```

**Parameter**

- **ts** Der zu konvertierende Timestamp-Wert.

**Rückgabe**

**ts**, wenn **ts** eine Instanz von **TimestampWithTimeZone** ist. Andernfalls erstellt sie einen neuen **TimestampWithTimeZone**, der **ts** mit einem Zeitzonen-Offset von 00:00 entspricht.

## valueOf-Methode

Konvertiert eine Zeichenfolge in einen **TimestampWithTimeZone**-Wert.

**Syntax**

```
TimestampWithTimeZone TimestampWithTimeZone.valueOf(String val)
```

**Parameter**

- **val** Ein Zeitstempel mit Zeitzone im Format **yyyy-mm-dd hh:mm:ss[.f...][ [-+]hh:mm]**. Die Bruch- und Zeitzone-Teile können ausgelassen werden. Wenn die Zeitzone vorhanden ist, kann das Vorzeichen weggelassen werden.

**Rückgabe**

Das neue `TimestampWithTimeZone`-Objekt.

**Ausnahmebedingungen**

- **java.lang.IllegalArgumentException** Wird ausgegeben, wenn **val** nicht das richtige Format hat.

# UpdateResultSet-Schnittstelle

Ein Ergebnismengenobjekt, das spezielle Methoden zum Zugriff auf Pre-Image- (alt) und Post-Image-Werte (neu) einer bestimmten Zeile umfasst.

**Syntax**

```
public interface UpdateResultSet
```

**Mitglieder**

Alle Mitglieder der `UpdateResultSet`-Schnittstelle, einschließlich aller geerbten Mitglieder.

| Name                                    | Beschreibung                                                                                                  |
|-----------------------------------------|---------------------------------------------------------------------------------------------------------------|
| <a href="#">setNewRowValues-Methode</a> | Legt den Modus dieser Ergebnismenge so fest, dass neue Spaltenwerte zurückgegeben werden (Post-Update-Zeile). |
| <a href="#">setOldRowValues-Methode</a> | Legt den Modus dieser Ergebnismenge so fest, dass alte Spaltenwerte zurückgegeben werden (Pre-Update-Zeile).  |

**Bemerkungen**

Sie können die Aktualisierungsvorgänge für eine Uploadtransaktion für eine Tabelle halten.

Neue und alte Zeilen können aufgerufen werden, indem Sie den Modus des **ResultSet** auf "alt" oder "neu" setzen.

Verwenden Sie die **DownloadTableData.getUpdates**-Methode zum Abrufen einer **UpdateResultSet**-Instanz.

**UpdateResultSet** erweitert **java.sql.ResultSet** und fügt die **setNewRowValues**- und **setOldRowValues**-Methode hinzu. Andernfalls kann dies als reguläre **ResultSet** verwendet werden.

Weitere Hinweise zu **java.sql.ResultSet** finden Sie in Ihrer Java Software Development Kit-Dokumentation.

Weitere Hinweise zur Verarbeitung von Konflikten beim direkten Upload finden Sie unter [„Konflikte bei direkten Uploads“ auf Seite 632](#).

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter [„Direkte Zeilenbehandlung“ auf Seite 627](#).

**Siehe auch**

- [UploadedTableData.getUpdates-Methode \[MobiLink-Server Java\] auf Seite 535](#)

## setNewRowValues-Methode

Legt den Modus dieser Ergebnismenge so fest, dass neue Spaltenwerte zurückgegeben werden (Post-Update-Zeile).

**Syntax**

```
void UpdateResultSet.setNewRowValues()
```

**Bemerkungen**

Die Ergebnismenge stellt die neuesten aktualisierten Werte in der entfernten Clientdatenbank dar.

Dies ist der Standardmodus.

Weitere Hinweise zur Verarbeitung von Konflikten beim direkten Upload finden Sie unter [„Konflikte bei direkten Uploads“ auf Seite 632](#).

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter [„Direkte Zeilenbehandlung“ auf Seite 627](#).

**Beispiel**

Der folgende Code zeigt, wie Sie den Modus von **UpdateResultSet** festlegen, um neue Spaltenwerte zurückzugeben:

```
UpdateResultSet results  
results.setNewRowValues();
```

## setOldRowValues-Methode

Legt den Modus dieser Ergebnismenge so fest, dass alte Spaltenwerte zurückgegeben werden (Pre-Update-Zeile).

**Syntax**

```
void UpdateResultSet.setOldRowValues()
```

**Bemerkungen**

In diesem Modus stellt **UpdateResultSet** alte Spaltenwerte dar, die der Client bei der letzten Synchronisation abgerufen hat.

Weitere Hinweise zur Verarbeitung von Konflikten beim direkten Upload finden Sie unter [„Konflikte bei direkten Uploads“ auf Seite 632](#).

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

**Beispiel**

Der folgende Code zeigt, wie Sie den Modus von **UpdateResultSet** festlegen, um alte Spaltenwerte zurückzugeben:

```
UpdateResultSet results
results.setOldRowValues();
```

# UploadData-Schnittstelle

Kapselt Upload-Vorgänge für die direkte Zeilenbehandlung.

**Syntax**

```
public interface UploadData
```

**Mitglieder**

Alle Mitglieder der UploadData-Schnittstelle, einschließlich aller geerbten Mitglieder.

| Name                                           | Beschreibung                                                                                   |
|------------------------------------------------|------------------------------------------------------------------------------------------------|
| <a href="#">getUploadedTableByName-Methode</a> | Gibt eine <b>UploadedTableData</b> -Instanz zurück, die die angegebene Tabelle darstellt.      |
| <a href="#">getUploadedTables-Methode</a>      | Gibt ein Array von <b>UploadedTableData</b> -Objekten für die aktuelle Synchronisation zurück. |

**Bemerkungen**

Eine **UploadData**-Instanz, die eine einzelne Uploadtransaktion darstellt, wird an das handle\_UploadData-Ereignis übergeben.

**Hinweis**

Sie müssen Uploadvorgänge zur direkten Zeilenbehandlung in der Methode verarbeiten, die für das Ereignis handle\_UploadData registriert ist. **UploadData** wird nach jedem Aufruf der registrierten Methode gelöscht. Erstellen Sie keine neue Instanz von **UploadData** für nachfolgende Ereignisse.

Mit der **getUploadedTables**- oder **getUploadedTableByName**-Methode rufen Sie **UploadedTableData**-Instanzen ab.

Eine Synchronisation enthält ein **UploadData**, außer wenn die entfernte Datenbank den transaktionalen Upload verwendet.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

Weitere Hinweise zur Verarbeitung von Konflikten beim direkten Upload finden Sie unter „[Konflikte bei direkten Uploads](#)“ auf Seite 632.

Weitere Hinweise zum `handle_UploadData`-Verbindungsereignis finden Sie unter „[handle\\_UploadData \(Verbindungsereignis\)](#)“ auf Seite 383.

#### Siehe auch

- [UploadedTableData-Schnittstelle \[MobiLink-Server Java\] auf Seite 530](#)

## getUploadedTableName-Methode

Gibt eine **UploadedTableData**-Instanz zurück, die die angegebene Tabelle darstellt.

#### Syntax

```
UploadedTableData UploadData.getUploadedTableName(String table_name)
```

#### Parameter

- **table\_name** Der Name der Uploadtabelle, deren Daten Sie hochladen möchten.

#### Rückgabe

Eine **UploadedTableData**-Instanz, die die angegebene Tabelle darstellt, oder Null, wenn keine Tabelle mit dem angegebenen Namen für die aktuelle Synchronisation vorhanden ist.

#### Siehe auch

- [UploadedTableData-Schnittstelle \[MobiLink-Server Java\] auf Seite 530](#)

#### Beispiel

Angenommen, Sie verwenden eine Methode namens **HandleUpload** für das `handle_UploadData`-Ereignis. Im folgenden Beispiel wird mit der **GetUploadedTableName**-Methode eine **UploadedTableData**-Instanz für **remoteOrderstable** zurückgegeben .

```
public void handleUpload(UploadData ud)
    throws SQLException, IOException
{
    UploadedTableData uploaded_t1 =
    ud.GetUploadedTableName("remoteOrders");
    //...
}
```

## getUploadedTables-Methode

Gibt ein Array von **UploadedTableData**-Objekten für die aktuelle Synchronisation zurück.

#### Syntax

```
UploadedTableData[] UploadData.getUploadedTables()
```

### Rückgabe

Ein Array von **UploadedTableData**-Objekten für die aktuelle Synchronisation. Die Reihenfolge der Tabellen im Array ist die gleiche wie die Upload-Reihenfolge für den Client.

### Bemerkungen

Die Reihenfolge der Tabellen im Array entspricht der Reihenfolge, die MobiLink für die SQL-Zeilenbehandlung benutzt. Diese Reihenfolge ist optimal für die Vermeidung von Verletzungen der referenziellen Integrität. Verwenden Sie diese Tabellenreihenfolge, wenn es sich bei Ihrer Datenquelle um eine relationale Datenbank handelt.

### Siehe auch

- [UploadedTableData-Schnittstelle \[MobiLink-Server Java\] auf Seite 530](#)

### Beispiel

Angenommen, Sie verwenden eine Methode namens **HandleUpload** für das `handle_UploadData`-Ereignis. Im folgenden Beispiel wird mit der **getUploadedTables**-Methode eine **UploadedTableData**-Instanz für die aktuelle Uploadtransaktion zurückgegeben.

```
public void handleUpload(UploadData ud)
    throws SQLException, IOException
{
    UploadedTableData tables[] = ud.getUploadedTables();
    //...
}
```

## UploadedTableData-Schnittstelle

Verpackt Tabellenvorgänge für die direkte Zeilenbehandlung.

### Syntax

```
public interface UploadedTableData
```

### Mitglieder

Alle Mitglieder der UploadedTableData-Schnittstelle, einschließlich aller geerbten Mitglieder.

| Name                                | Beschreibung                                                                                                                        |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getDeletes-Methode</a>  | Gibt ein <b>java.sql.ResultSet</b> -Objekt zurück, das Löschvorgänge darstellt, die von einem MobiLink-Client hochgeladen wurden.   |
| <a href="#">getInserts-Methode</a>  | Gibt ein <b>java.sql.ResultSet</b> -Objekt zurück, das Einfügevorgänge darstellt, die von einem MobiLink-Client hochgeladen wurden. |
| <a href="#">getMetaData-Methode</a> | Ruft die Metadaten für die <b>UploadedTableData</b> -Instanz ab.                                                                    |
| <a href="#">getName-Methode</a>     | Gibt den Tabellennamen für die <b>UploadedTableData</b> -Instanz zurück.                                                            |

| Name                               | Beschreibung                                                                                                                             |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">getUpdates-Methode</a> | Gibt ein <b>UpdateResultSet</b> -Objekt zurück, das Aktualisierungsvorgänge darstellt, die von einem MobiLink-Client hochgeladen wurden. |

## Bemerkungen

Mit einer **UploadedTableData**-Instanz können Sie die Einfüge-, Aktualisierung- und Löschvorgänge einer Tabelle für eine einzelne Uploadtransaktion abrufen. Mit den Methoden **getInserts**, **getUpdates** und **getDeletes** geben Sie standardmäßige JDBC-**java.sql.ResultSet**-Objekte zurück.

Weitere Hinweise zu **java.sql.ResultSet** und **java.sql.ResultSetMetaData** finden Sie in der Java Software Development Kit-Dokumentation.

Tabellenmetadaten können Sie mit der **getMetaData**-Methode oder mit den von **getInserts**, **getUpdates** und **getDeletes** zurückgegebenen Ergebnismengen aufrufen. Die Löschergebnismenge beinhaltet nur die Primärschlüsselspalten einer Tabelle.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

Weitere Hinweise zum `handle_UploadData`-Verbindungsereignis finden Sie unter „[handle\\_UploadData \(Verbindungsereignis\)](#)“ auf Seite 383.

## Siehe auch

- [UploadData-Schnittstelle \[MobiLink-Server Java\] auf Seite 528](#)

## Beispiel

Der folgende Code ruft die hochgeladenen Löschungen ab und gibt die jeweils erste Spalte aus:

```
void printFirstColOfDeletes( UploadedTableData tab_data )
{
    ResultSet deletes = tab_data.getDeletes();
    while( deletes.next() ) {
        java.lang.System.out.println( deletes.getString( 1 ) );
    }
    deletes.close();
}
```

Der folgende Code gibt den neuen und alten Wert der dritten Spalte jeder Aktualisierung aus:

```
void printThirdColOfUpdates( UploadedTableData tab_data )
{
    ResultSet updates = tab_data.getUpdates();
    while( updates.next() ) {
        updates.setOldRowValues();
        java.lang.System.out.println( "old row col: " +
updates.getString( 3 ) );
        updates.setNewRowValues();
        java.lang.System.out.println( "new row col: " +
updates.getString( 3 ) );
    }
    updates.close();
}
```

## getDeletes-Methode

Gibt ein **java.sql.ResultSet**-Objekt zurück, das Löschvorgänge darstellt, die von einem MobiLink-Client hochgeladen wurden.

### Syntax

```
java.sql.ResultSet UploadedTableData.getDeletes()
```

### Rückgabe

Ein **java.sql.ResultSet**-Objekt, das Löschvorgänge darstellt, die von einem MobiLink-Client hochgeladen wurden.

### Bemerkungen

Die Ergebnismenge enthält Primärschlüsselwerte für gelöschte Zeilen.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

Weitere Hinweise zum `handle_UploadData`-Verbindungsereignis finden Sie unter „[handle\\_UploadData \(Verbindungsereignis\)](#)“ auf Seite 383.

### Beispiel

Angenommen, der entfernte Client enthält eine Tabelle namens **remoteOrders**. Im folgenden Beispiel wird eine Ergebnismenge für gelöschte Zeilen mit der **DownloadTableData.getDeletes**-Methode abgerufen. In diesem Fall enthält die Löschergebnismenge eine einzelne Primärschlüsselspalte.

```
import ianywhere.ml.script.*;
import java.sql.*;

// The method used for the handle_UploadData event.
public void HandleUpload(UploadData ut)
    throws SQLException, IOException
{
    // Get an UploadedTableData for the remoteOrders table.
    UploadedTableData remoteOrdersTable =
ut.getUploadedTableByName("remoteOrders");

    // Get deletes uploaded by the MobiLink client.
    java.sql.ResultSet delete_rs = remoteOrdersTable.getDeletes();

    while (delete_rs.next()) {
        // Get primary key values for deleted rows.
        int deleted_id = delete_rs.getInt(1);

        // ...
    }
    delete_rs.close();
}
```

## getInserts-Methode

Gibt ein **java.sql.ResultSet**-Objekt zurück, das Einfügevorgänge darstellt, die von einem MobiLink-Client hochgeladen wurden.

**Syntax**

```
java.sql.ResultSet UploadedTableData.getInserts()
```

**Rückgabe**

Ein **java.sql.ResultSet**-Objekt, das Einfügevorgänge darstellt, die von einem MobiLink-Client hochgeladen wurden.

**Bemerkungen**

Jede Einfügung wird durch eine Zeile in der Ergebnismenge dargestellt.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter [„Direkte Zeilenbehandlung“ auf Seite 627](#).

**Beispiel**

Angenommen, der entfernte Client enthält eine Tabelle namens **remoteOrders**. Im folgenden Beispiel wird eine Ergebnismenge für eingefügte Zeilen mit der **DownloadTableData.getInserts**-Methode abgerufen. Der Code ermittelt die Auftragsmenge für jede Zeile in der aktuellen Uploadtransaktion.

```
import ianywhere.ml.script.*;
import java.sql.*;

// The method used for the handle_UploadData event
public void HandleUpload(UploadData ut)
    throws SQLException, IOException
{
    // Get an UploadedTableData instance representing the remoteOrders table.
    UploadedTableData remoteOrdersTable =
    ut.getUploadedTableByName("remoteOrders");

    // Get inserts uploaded by the MobiLink client.
    java.sql.ResultSet rs = remoteOrdersTable.getInserts();
    while (rs.next()) {
        // get the uploaded order_amount
        double order_amount = rs.getDouble("order_amount");

        // ...
    }
    rs.close();
}
```

**getMetaData-Methode**

Ruft die Metadaten für die **UploadedTableData**-Instanz ab.

**Syntax**

```
java.sql.ResultSetMetaData UploadedTableData.getMetaData()
```

**Rückgabe**

Die Metadaten für die **UploadedTableData**-Instanz.

## Bemerkungen

Die Metadaten sind eine standardmäßige **java.sql.ResultSetMetaData**-Instanz.

Wenn **ResultSetMetaData** Spaltennamen-Informationen enthalten soll, müssen Sie die Client-Option zum Senden von Spaltennamen angeben.

Weitere Hinweise zu **java.sql.ResultSetMetaData** finden Sie in Ihrer Java Software Development Kit-Dokumentation.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter [„Direkte Zeilenbehandlung“](#) auf Seite 627.

Weitere Hinweise zum `handle_UploadData`-Verbindungsereignis finden Sie unter [„handle\\_UploadData \(Verbindungsereignis\)“](#) auf Seite 383.

## Beispiel

Im folgenden Beispiel wird eine **java.sql.ResultSetMetaData**-Instanz für eine Uploadtabelle namens **remoteOrders** abgerufen. Der Code kompiliert unter Verwendung der Methoden **ResultSetMetaData.getColumnCount** und **ResultSetMetaData.getColumnLabel** eine Liste mit Spaltennamen.

```
import ianywhere.ml.script.*;
import java.sql.*;

// The method used for the handle_UploadData event.
public void HandleUpload(UploadData ut) {
    throws SQLException, IOException
    {
        // Get an UploadedTableData instance representing the remoteOrders table.
        UploadedTableData remoteOrdersTable =
        ut.getUploadedTableByName("remoteOrders");

        // get inserts uploaded by the MobiLink client
        java.sql.ResultSet rs = remoteOrdersTable.getInserts();

        // Obtain the result set metadata.
        java.sql.ResultSetMetaData md = rs.getMetaData();
        String columnHeading = "";

        // Compile a list of column names.
        for (int c=1; c <= md.getColumnCount(); c += 1) {
            columnHeading += md.getColumnLabel( c );

            if (c < md.getColumnCount()) {
                columnHeading += ", ";
            }
        }
        //...
    }
}
```

In diesem Fall verarbeitet eine Methode namens **HandleUpload** das `handle_UploadData`-Synchronisationsereignis.

## getName-Methode

Gibt den Tabellennamen für die **UploadedTableData**-Instanz zurück.

**Syntax**

```
String UploadedTableData.getName()
```

**Rückgabe**

Der Tabellennamen für die **UploadedTableData**-Instanz.

**Bemerkungen**

Sie können den Tabellennamen auch mit der **java.sql.ResultSetMetaData**-Instanz aufrufen, die von der **getMetaData**-Methode zurückgegeben wird.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

Weitere Hinweise zum handle\_UploadData-Verbindungsereignis finden Sie unter „[handle\\_UploadData \(Verbindungsereignis\)](#)“ auf Seite 383.

**Siehe auch**

- [UploadedTableData.getMetaData-Methode \[MobiLink-Server Java\] auf Seite 533](#)

**Beispiel**

Im folgenden Beispiel wird der Name jeder Uploadtabelle in einer einzelnen Uploadtransaktion abgerufen:

```
import anywhere.ml.script.*;
import java.sql.*;

// The method used for the handle_UploadData event.
public void HandleUpload(UploadData ud) {
    throws SQLException, IOException
    {
        int i;

        // Get UploadedTableData instances.
        UploadedTableData tables[] = ud.getUploadedTables();

        for (i=0; i<tables.length; i+=1) {
            // Get the table name.
            String table_name = tables[i].getName();

            // ...
        }
    }
}
```

**getUpdates-Methode**

Gibt ein **UpdateResultSet**-Objekt zurück, das Aktualisierungsvorgänge darstellt, die von einem MobiLink-Client hochgeladen wurden.

**Syntax**

```
UpdateResultSet UploadedTableData.getUpdates()
```

## Rückgabe

Ein **UpdateResultSet**-Objekt, das Aktualisierungsvorgänge darstellt, die von einem MobiLink-Client hochgeladen wurden.

## Bemerkungen

Jede Aktualisierung wird durch eine Zeile einschließlich aller Spaltenwerte dargestellt. **UpdateResultSet** erweitert **java.sql.ResultSet** mit speziellen Methoden zur MobiLink-Konflikterkennung.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

Weitere Hinweise zur Verarbeitung von Konflikten beim direkten Upload finden Sie unter „[Konflikte bei direkten Uploads](#)“ auf Seite 632.

Weitere Hinweise zum `handle_UploadData`-Verbindungsereignis finden Sie unter „[handle\\_UploadData \(Verbindungsereignis\)](#)“ auf Seite 383.

## Siehe auch

- [UpdateResultSet-Schnittstelle \[MobiLink-Server Java\] auf Seite 526](#)

## Beispiel

Angenommen, der entfernte Client enthält eine Tabelle namens **remoteOrders**. Im folgenden Beispiel wird eine Ergebnismenge für aktualisierte Zeilen mit der **getUpdates**-Methode abgerufen. Der Code ermittelt die Auftragsmenge für jede Zeile.

```
import ianywhere.ml.script.*;
import java.sql.*;

// The method used for the handle_UploadData event.
public void HandleUpload(UploadData ut)
    throws SQLException, IOException
{
    // Get an UploadedTableData instance representing the remoteOrders table.
    UploadedTableData remoteOrdersTable =
    ut.getUploadedTableByName("remoteOrders");

    // Get inserts uploaded by the MobiLink client.
    java.sql.ResultSet rs = remoteOrdersTable.getUpdates();
    while (rs.next()) {
        // Get the uploaded order_amount.
        double order_amount = rs.getDouble("order_amount");

        // ...
    }
    rs.close();
}
```

# Synchronisationsskripten in .NET

MobiLink unterstützt die Visual Studio-Programmiersprachen zum Schreiben von Synchronisationsskripten. Zum Schreiben von MobiLink-Skripten in .NET können Sie jede beliebige

Sprache verwenden, mit der Sie gültige .NET-Assemblys erstellen können. Insbesondere wurden folgende Sprachen getestet und dokumentiert:

- C#
- Visual Basic .NET
- C++

Die .NET-Synchronisationslogik kann auf dieselbe Weise wie SQL-Logikfunktionen eingesetzt werden: Der MobiLink-Server kann beim Eintritt von MobiLink-Ereignissen .NET-Methoden ebenso wie SQL-Skripten aufrufen. Eine .NET-Methode kann eine SQL-Zeichenfolge an MobiLink zurückgeben.

Dieser Abschnitt erläutert, wie Sie .NET-Synchronisationslogik für C#, Visual Basic .NET und C++ einrichten, entwickeln und anwenden. Außerdem werden eine Beispielanwendung und die API für .NET des MobiLink-Servers beschrieben.

### Siehe auch

- „Optionen zum Erstellen serverseitiger Synchronisationslogik“ [[MobiLink - Erste Orientierung](#)]
- „Schreiben von Synchronisationsskripten“ auf Seite 231

## Synchronisationsskripten in .NET implementieren

Wenn Sie Synchronisationsskripten in .NET implementieren, müssen Sie MobiLink angeben, wo sich die Pakete, Klassen und Methoden befinden, die in Ihren Assemblys enthalten sind.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Aufgabe

1. Erstellen Sie Ihre Klassen. Schreiben Sie für jedes einzelne erforderliche Synchronisationsereignis eine Methode. Diese Methoden müssen öffentlich sein.

Weitere Hinweise zu Methoden finden Sie unter „[Methoden](#)“ auf Seite 542.

Jede Klasse mit nichtstatischen Methoden muss über einen öffentlichen Konstruktor verfügen. Der MobiLink-Server instanziert automatisch jede einzelne Klasse, wenn eine Methode in der betreffenden Klasse zum ersten Mal für eine Verbindung aufgerufen wird.

Siehe „[Konstruktoren](#)“ auf Seite 541.

2. Erstellen Sie eine oder mehrere Assemblys. Referenzieren Sie beim Kompilieren die Datei *iAnywhere.MobiLink.Script.dll*, die ein Repository von API-Klassen des MobiLink-Servers enthält, die in Ihren eigenen .NET-Methoden verwendet werden sollen. *iAnywhere.MobiLink.Script.dll* befindet sich in `%SQLANYI6%\Assembly\V2`.

Sie können Ihre Klasse von der Befehlszeile aus kompilieren bzw. Visual Studio oder eine andere .NET-Entwicklungsumgebung verwenden.

Siehe „[Referenz zur MobiLink-Server-API für .NET](#)“ auf Seite 552.

3. Kompilieren Sie Ihr Projekt.

So kompilieren Sie beispielsweise von Visual Studio aus:

- a. Klicken Sie im **VS.NET Projektmenü** auf die Option **Vorhandenes Element hinzufügen**.
- b. Suchen Sie die Datei *iAnywhere.MobiLink.Script.dll*.  
In der Liste **Offen** klicken Sie auf **Datei verknüpfen**.

**Hinweis**

Verwenden Sie für Visual Studio immer die Dateiverknüpfungsmethode. Verwenden Sie nicht die Option "Verweis hinzufügen", um *iAnywhere.MobiLink.Script.dll* zu referenzieren. Die Option "Verweis hinzufügen" dupliziert die Datei *iAnywhere.MobiLink.Script.dll* im physischen Verzeichnis Ihrer Klassen-Assembly, wodurch Probleme für den MobiLink-Server entstehen.

- c. Verwenden Sie das Menü **Erstellen**, um Ihr Assembly zu erstellen.

Sie können auch wie nachfolgend beschrieben über die Befehlszeile kompilieren:

Ersetzen Sie *dll-Pfad* mit dem Pfad zur Datei *iAnywhere.MobiLink.Script.dll*, z.B. in C#:

```
csc /out:dll-pathout.dll /target:library /reference:dll-  
pathiAnywhere.MobiLink.Script.dll sync_v1.cs
```

4. In den MobiLink-Systemtabellen Ihrer konsolidierten Datenbank geben Sie für jedes einzelne Skript den Namen des aufzurufenden Pakets, der Klasse und der Methode an. Pro Skriptversion ist nicht mehr als eine Klasse zulässig.

Sie können diese Angaben beispielsweise mit der gespeicherten Prozedur `ml_add_dnet_connection_script` oder `ml_add_dnet_table_script` in die MobiLink-Systemtabellen einfügen. Wenn die nachfolgende SQL-Anweisung in einer SQL Anywhere-Datenbank ausgeführt wird, legt sie fest, dass `myNamespace.myClass.myMethod` immer dann ausgeführt werden soll, wenn auf Verbindungsebene das Ereignis `authenticate_user` eintritt.

```
CALL ml_add_dnet_connection_script(  
    'version1',  
    'authenticate_user',  
    'myNamespace.myClass.myMethod'  
)
```

**Hinweis**

Der vollständig angegebene Methodenname berücksichtigt die Groß- und Kleinschreibung.

Dieser Prozeduraufruf bewirkt, dass die Spalte `script_language` in der Systemtabelle `ml_script` das Wort **dnet** enthalten muss. Die Skriptspalte enthält den qualifizierten Namen einer öffentlichen .NET-Methode.

Siehe „[ml\\_add\\_dnet\\_connection\\_script-Systemprozedur](#)“ auf Seite 653 und „[ml\\_add\\_dnet\\_table\\_script-Systemprozedur](#)“ auf Seite 654.

Es besteht außerdem die Möglichkeit, diese Angaben mit Sybase Central hinzuzufügen.

Siehe „Hinzufügungen und Löschungen von Skripten“ auf Seite 258.

5. Weisen Sie den MobiLink-Server an, Assemblys zu laden und den CLR zu starten. Sie geben für MobiLink mit Optionen in der mlsrv16-Befehlszeile an, wo sich diese Assemblys befinden. Sie können zwischen zwei Optionen wählen:

- **-sl dnet verwenden (-MLAutoLoadPath)** Hiermit setzen Sie den Pfad auf das Stammverzeichnis der Anwendung und laden alle privaten Assemblys aus diesem Verzeichnis. In der Regel wird diese Option bevorzugt. Um beispielsweise alle Assemblys aus *dll-path* zu laden, geben Sie Folgendes ein:

```
mlsrv16 -c "DSN=consolidated1" -sl dnet(-MLAutoLoadPath=dll-path)
```

Wenn Sie die Option -MLAutoLoadPath verwenden, können Sie im Ereignisskript bei der Eingabe eines voll qualifizierten Methodennamens keine Domäne angeben.

Siehe „Laden von .NET-Assemblys“ auf Seite 548 und „mlsrv16-Option -sl dnet“ auf Seite 78.

- **-sl dnet verwenden (-MLDomConfigFile)** Diese Option erfordert eine Konfigurationsdatei, die Domänen- und Assembly-Einstellungen enthält. Verwenden Sie diese Option, wenn Sie gemeinsam genutzte Assemblys haben und nicht alle Assemblys in einem Verzeichnis laden möchten oder wenn Sie aus einem anderen Grund eine Konfigurationsdatei verwenden müssen.

Weitere Hinweise zum Laden von gemeinsam genutzten Assemblys finden Sie unter „Laden von .NET-Assemblys“ auf Seite 548.

Weitere Hinweise zur mlsrv16-Option -sl dnet finden Sie unter „mlsrv16-Option -sl dnet“ auf Seite 78.

#### Hinweis

Sie können die Option -MLAutoLoadPath oder -MLDomConfigFile verwenden, aber nicht beide.

## Ergebnisse

Die .NET-Synchronisationslogik wird eingerichtet.

## .NET-Synchronisationslogik

Zum Schreiben von .NET-Synchronisationslogik sind Kenntnisse über die MobiLink-Ereignisse, über .NET sowie über die API für .NET des MobiLink-Servers erforderlich.

Eine vollständige Beschreibung der API-Schnittstelle finden Sie unter „Referenz zur MobiLink-Server-API für .NET“ auf Seite 552.

.NET-Synchronisationslogik kann zur Aufrechterhaltung der Statusinformationen und zur Implementierung der Logik vor und nach Upload- und Download-Ereignissen eingesetzt werden. Ein in .NET geschriebenes begin\_synchronization-Skript könnte z.B. den MobiLink Benutzernamen in einer Variablen speichern. Später im Synchronisationsprozess aufgerufene Skripten können dann auf diese Variable zugreifen. Sie können auch mit .NET auf Zeilen in der konsolidierten Datenbank zugreifen, und zwar bevor und nachdem sie festgeschrieben wurden.

Der Einsatz von .NET verringert außerdem die Abhängigkeit von der konsolidierten Datenbank. Das Aufrüsten der konsolidierten Datenbank auf eine neuere Version oder der Wechsel auf ein anderes Datenbank-Managementsystem beeinträchtigt das Verhalten in geringerem Umfang.

### Direkte Zeilenbehandlung

Mit der direkten Zeilenbehandlung in MobiLink übermitteln Sie entfernte Daten an jede zentrale Datenquelle, Anwendung oder einen Webdienst. Die direkte Zeilenbehandlung verwendet spezielle Klassen in den APIs für Java oder .NET des MobiLink-Servers, um direkt auf synchronisierte Daten zuzugreifen.

Siehe „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

## Klasseninstanzen

Der MobiLink-Server instanziert Ihre Klassen auf der Datenbank-Verbindungsebene. Wenn ein Ereignis erreicht wurde, für das Sie eine nicht statische .NET-Methode geschrieben haben, baut der MobiLink-Server automatisch die Klasse auf, falls dies bei der aktuellen Datenbankverbindung noch nicht geschehen ist.

Siehe „[Konstruktoren](#)“ auf Seite 541.

#### Hinweis

Alle Methoden, die für eine Skriptversion einem Ereignis auf der Verbindungs- oder Tabellenebene direkt zugeordnet sind, müssen zu derselben Klasse gehören.

Nachdem eine Klasse instanziiert wurde, bleibt sie für jede einzelne Datenbankverbindung bestehen, bis die Verbindung geschlossen wird. Dieselbe Instanz kann so auch für mehrere aufeinander folgende Synchronisationssitzungen verwendet werden. Wenn die Informationen in öffentlichen oder privaten Variablen nicht explizit gelöscht werden, bleiben sie für alle Synchronisationen, die über dieselbe Verbindung ausgeführt werden, bestehen.

Sie können auch statische Klassen bzw. Variablen benutzen. In diesem Fall stehen die Werte in allen Verbindungen in einer Domäne zur Verfügung.

Der MobiLink-Server löscht Ihre Klassen nur dann automatisch, wenn die Verbindung zur konsolidierten Datenbank geschlossen wird.

## Transaktionen

Die normalen Regeln im Hinblick auf Transaktionen beziehen sich auf .NET-Methoden. Start und Dauer von Datenbanktransaktionen sind wesentlich für den Synchronisationsprozess. Transaktionen dürfen nur vom MobiLink-Server gestartet oder beendet werden. Wenn Sie bei einer Synchronisationsverbindung und innerhalb einer .NET-Methode Transaktionen ausdrücklich festschreiben oder zurücksetzen, verletzen Sie die Integrität des Synchronisationsprozesses und lösen damit eventuell in späterer Folge Fehler aus.

Diese Regeln betreffen nur die vom MobiLink-Server hergestellten Datenbankverbindungen und insbesondere die von Methoden zurückgegebenen SQL-Anweisungen.

## SQL-.NET-Datentypen

Die nachstehende Tabelle enthält SQL-Datentypen und die jeweils entsprechenden .NET-Datentypen.

| SQL-Datentyp    | Entsprechender .NET-Datentyp |
|-----------------|------------------------------|
| VARCHAR         | string                       |
| CHAR            | string                       |
| INTEGER         | int                          |
| BIGINT          | long                         |
| BINARY          | byte [ ]                     |
| TIMESTAMP       | DateTime                     |
| INOUT INTEGER   | ref int                      |
| INOUT VARCHAR   | ref string                   |
| INOUT CHAR      | ref string                   |
| INOUT BYTEARRAY | ref byte [ ]                 |
| INOUT TIMESTAMP | ref DateTime                 |

## Konstrukturen

Der Konstruktor Ihrer Klasse verwendet entweder keinen Parameter (leerer Parameter) oder den Parameter `iAnywhere.MobiLink.Script.DBConnectionContext`. Beispiel:

```
public ExampleClass(iAnywhere.MobiLink.Script.DBConnectionContext cc)
```

oder

```
public ExampleClass()
```

Der an Sie weitergegebene Synchronisationskontext gilt für die Verbindung, über die der MobiLink-Server den aktuellen Benutzer synchronisiert.

Die Methode `DBConnectionContext.GetConnection` gibt dieselbe Datenbankverbindung zurück, die MobiLink benutzt, um den aktuellen Benutzer zu synchronisieren. Sie können Anweisungen in dieser Verbindung ausführen, dürfen die Transaktion jedoch weder festschreiben noch zurücksetzen. Der MobiLink-Server verwaltet die Transaktionen.

Der MobiLink-Server verwendet den Konstruktor, der einen `iAnywhere.MobiLink.Script.DBConnectionContext-Parameter` (falls vorhanden) aufnimmt. Ist dies nicht der Fall, verwendet er den leeren Konstruktor.

Siehe [DBConnectionContext-Schnittstelle \[MobiLink-Server .NET\] auf Seite 565](#).

## Methoden

Im Allgemeinen implementieren Sie für jedes einzelne Synchronisationsereignis eine Methode. Diese Methoden müssen öffentlich sein. Wenn sie nicht öffentlich sind, kann der MobiLink-Server sie nicht benutzen und nicht feststellen, ob sie überhaupt vorhanden sind.

Die Namen der Methoden sind unerheblich, sofern sie mit den in der Tabelle `ml_script` in der konsolidierten Datenbank angegebenen Namen übereinstimmen. In den Beispielen in der Dokumentation sind die Methodennamen jedoch die gleichen wie in den MobiLink-Ereignissen. Diese Namenskonvention vereinfacht das Lesen des .NET-Codes.

Die Signatur Ihrer Methode sollte mit der Signatur des Skripts für das betreffende Ereignis übereinstimmen, nur dass Sie die Parameterliste verkürzen können, wenn Sie die Werte am Ende der Liste nicht benötigen. Sie sollten nur die benötigten Parameter akzeptieren, weil die Weitergabe der Parameter Overhead verursacht.

Methoden können jedoch nicht überladen werden. In der Systemtabelle `ml_script` kann nur ein Methodenprototyp per Klasse gespeichert werden.

### Werte registrieren

Methoden, die für einen Upload oder Download auf SQL-Basis aufgerufen werden, müssen eine gültige SQL-Anweisung zurückgeben. Der Rückgabetyt dieser Methoden muss `string` sein. Andere Rückgabetypen sind nicht zulässig.

Der Rückgabetyt für alle anderen Skripten muss entweder `string` oder `void` sein. Es sind keine anderen Typen zulässig. Wenn der Rückgabetyt eine Zeichenfolge und nicht `Null` ist, geht der MobiLink-Server davon aus, dass die Zeichenfolge eine gültige SQL-Anweisung enthält, und führt die Anweisung in der konsolidierten Datenbank aus, ebenso wie er es in einem gewöhnlichen SQL-Synchronisationsskript getan hätte. Wenn eine Methode normalerweise eine Zeichenfolge zurückgibt, bei Rückgabe jedoch keine SQL-Anweisung in der Datenbank ausgeführt werden soll, kann sie auch `Null` zurückgeben.

## Benutzerdefinierte Startklassen

Sie können Startklassen definieren, die beim Serverstart automatisch geladen werden. Diese Funktion gestattet es Ihnen, .NET-Code zu schreiben, der ausgeführt wird, wenn der MobiLink-Server CLR startet, also vor der ersten Synchronisation. Das bedeutet, dass Sie vor der ersten Synchronisationsanforderung eines Benutzers in der Serverinstanz Verbindungen oder Cachedaten erstellen können.

Hierzu verwenden Sie die Option `MLStartClasses` der `mlsrv16`-Option `-sl dnet`. Die `mlsrv16`-Befehlszeile kann beispielsweise Folgendes enthalten. Damit wird bewirkt, dass `mycl1` und `mycl2` als Startklassen geladen werden.

```
-sl dnet(-MLStartClasses=MyNameSpace.MyClass.mycl1,MyNameSpace.MyClass.mycl2)
```

Klassen werden in der Reihenfolge geladen, in der sie aufgelistet sind. Wenn dieselbe Klasse mehr als einmal aufgelistet ist, wird mehr als eine Instanz erstellt.

Alle Startklassen müssen öffentlich sein und einen öffentlichen Konstruktor haben, der entweder keine Argumente oder ein Argument vom Typ "MobiLink.Script.ServerContext" übernimmt.

Die Namen der geladenen Startklassen werden mit der Meldung ".NET-Startklasse geladen: *Klassenname*".

Weitere Hinweise zu .NET CLR finden Sie unter „[mlsrv16-Option -sl dnet](#)“ auf Seite 78.

Welche Startklassen beim Serverstart erstellt werden, finden Sie unter [ServerContext.GetStartClassInstances-Methode](#) [MobiLink-Server .NET] auf Seite 606.

## Beispiel

Im Folgenden sehen Sie eine Vorlage für eine Startklasse. Sie startet einen Daemon-Thread, der Ereignisse verarbeitet und eine Datenbankverbindung erstellt. (Nicht alle Startklassen müssen einen Thread erstellen, aber wenn ein Thread angestoßen wird, muss es sich um einen Daemon-Thread handeln.)

```
using System;
using System.IO;
using System.Threading;
using iAnywhere.MobiLink.Script;

namespace TestScripts {
    public class MyStartClass {
        ServerContext    _sc;
        bool              _exit_loop;
        Thread            _thread;
        OdbcConnection    _conn;

        public MyStartClass(ServerContext sc) {

            // Perform setup first so that an exception
            // causes MobiLink startup to fail.
            _sc = sc;

            // Create connection for use later.
            _conn = _sc.makeConnection();
            _exit_loop = false;
            _thread = new Thread(new ThreadStart(run)) ;
            _thread.IsBackground = true;
            _thread.Start();
        }

        public void run() {
            ShutdownCallback callback = new
            ShutdownCallback(shutdownPerformed);
            _sc.ShutdownListener += callback;

            // run() can't throw exceptions.
            try {
                handlerLoop();
                _conn.close();
                _conn = null;
            }
        }
    }
}
```

```
    }
    catch(Exception e) {
        // Print some error output to the MobiLink log.
        Console.Error.Write(e.ToString());

        // There is no need to be notified of shutdown.
        _sc.ShutdownListener -= callback;

        // Ask server to shut down so this fatal error can be fixed.
        _sc.Shutdown();
    }

    // Shortly after return, this thread no longer exists.
    return;
}

public void shutdownPerformed(ServerContext sc) {
    // Stop the event handler loop.
    try {
        _exit_loop = true;

        // Wait a maximum of 10 seconds for thread to die.
        _thread.Join(10*1000);
    }
    catch(Exception e) {
        // Print some error output to the MobiLink log.
        Console.Error.Write(e.ToString());
    }
}

private void handlerLoop() {
    while (!_exit_loop) {
        // Handle events in this loop.
        Thread.Sleep(1*1000);
    }
}
}
```

## Informationen aus .NET drucken

Sie können auch wählen, Ihren .NET-Methoden, die mithilfe von System.Console Informationen in das MobiLink-Log schreiben, Anweisungen hinzuzufügen. Damit können Sie das Verhalten Ihrer Klassen leichter verfolgen.

### Hinweis

Informationen auf diese Weise im MobiLink-Log auszugeben, ist ein nützliches Hilfsmittel zur Überwachung, für eine Produktionsumgebung jedoch nicht empfehlenswert.

Dieselbe Technik kann benutzt werden, um zufällig ausgewählte Synchronisationsdaten zu protokollieren oder statistische Daten über den Einsatz Ihrer Skripten zu sammeln.

## Fehlerbehandlung des MobiLink-Servers mit .NET

Wenn das Durchsuchen des Logs nicht ausreicht, können Sie Ihre Anwendungen mithilfe von Programmen überwachen. Sie können z.B. Nachrichten eines bestimmten Typs in einer E-Mail versenden.

Sie können Methoden erstellen, die einer Klasse übergeben werden, die jeden Fehler oder jede Warnung umfasst, die im Log gespeichert werden. Dies kann Ihnen helfen, einen MobiLink-Server zu überwachen und mit der Audit-Funktion zu überprüfen.

Der folgende Code installiert einen Listener für alle Fehlermeldungen und schreibt die Informationen in einen StreamWriter.

```
class TestLogListener {
    public TestLogListener(StreamWriter output_file) {
        _output_file = output_file;
    }

    public void errCallback(ServerContext sc, LogMessage lm) {
        string type;
        string user;

        if (lm.Type == LogMessage.MessageType.ERROR) {
            type = "ERROR";
        } else if (lm.Type == LogMessage.MessageType.WARNING) {
            type = "WARNING";
        }
        else {
            type = "INVALID TYPE!!";
        }
        if (lm.User == null) {
            user = "null";
        }
        else {
            user = lm.User;
        }

        _output_file.WriteLine("Caught msg type=" + type
            + " user=" + user
            + " text=" + lm.Text);
        _output_file.Flush();
    }
    StreamWriter _output_file;
}
```

Mit dem nachfolgenden Code registrieren Sie den TestLogListener. Rufen Sie diesen Code dort auf, von wo aus Sie Zugriff auf den ServerContext haben, z.B. in einem Klassenkonstruktor oder einem Synchronisationsskript.

```
// ServerContext serv_context;
TestLogListener errtll = new TestLogListener(log_listener_file);
serv_context.ErrorListener += new LogCallback(errtll.errCallback);
```

### Siehe auch

- [LogCallback-Delegat \[MobiLink-Server .NET\] auf Seite 623](#)
- [ServerContext.ErrorListener-Ereignis \[MobiLink-Server .NET\] auf Seite 607](#)
- [ServerContext.WarningListener-Ereignis \[MobiLink-Server .NET\] auf Seite 608](#)
- [LogMessage-Klasse \[MobiLink-Server .NET\] auf Seite 595](#)
- [LogMessage.MessageType-Enumeration \[MobiLink-Server .NET\] auf Seite 596](#)

## Fehlersuche in der .NET-Synchronisationslogik

Die folgende Prozedur kann verwendet werden, um ein Debugging Ihrer .NET-Skripten mit Visual Studio durchzuführen.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Aufgabe

1. Starten Sie Visual Studio.
2. Wählen Sie **Extras » An den Prozess anhängen**.
3. Im Feld **Verfügbare Prozesse** wählen Sie **mlsrv16.exe** und drücken Sie dann **Anfügen**.
4. Setzen Sie Ihre Breakpoints.
5. Starten Sie eine Synchronisation.

### Ergebnisse

Ein Debugging für das Skript ist möglich.

Die folgende Prozedur kann verwendet werden, um ein Debugging Ihrer .NET-Skripten mit Visual Studio durchzuführen.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Debuggen von .NET-Skripten

1. Kompilieren Sie den Code mit einer der folgenden Methoden bei aktivierten Fehlersucheinformationen:
  - Geben Sie in der csc-Befehlszeile die Option **/debug+** an.
  - Verwenden Sie Microsoft Visual Studio-Einstellungen, um die Ausgabe der Fehlersuche festzulegen.
    - Klicken Sie auf **Erstellen » Konfigurations-Manager**.  
Klicken Sie in der Liste **Konfiguration der aktuellen Projektmappe** auf die Option **Debug**.
    - Erstellen Sie Ihre Assembly.
2. Schließen Sie alle ausgeführten Instanzen von Visual Studio, die Ihre Quelldateien enthalten.
3. Starten Sie im nächsten Schritt eine neue Visual Studio-Instanz, um auf dem MobiLink-Server und in den .NET-Synchronisationsskripten nach Fehlern zu suchen. Starten Sie Visual Studio mit einer Befehlszeilenooption, um auf dem MobiLink-Server nach Fehlern zu suchen.

- Wechseln Sie an der Eingabeaufforderung ins Unterverzeichnis *Common7\IDE* Ihrer Visual Studio-Installation.
- Starten Sie devenv (die integrierte Entwicklungsumgebung von Visual Studio) mithilfe der Option /debugexe.

Geben Sie beispielsweise den folgenden Befehl ein, um auf dem MobiLink-Server nach Fehlern zu suchen. Vergessen Sie nicht, die mlsrv16-Optionen anzugeben, einschließlich der Verbindungszeichenfolge und der Option zum Laden der .NET-Assemblys.

Für 32-Bit-Windows-Umgebungen:

```
devenv /debugexe %sqlany16%\bin32\mlsrv16.exe -c ...
```

Für 64-Bit-Windows-Umgebungen:

```
devenv /debugexe %sqlany16%\bin64\mlsrv16.exe -c ...
```

Visual Studio startet und im Fenster des Projektmappen-Explorers wird *mlsrv16.exe* angezeigt.

#### 4. Richten Sie Microsoft Visual Studio für die Fehlersuche im .NET-Code ein:

- Im Fenster des Visual Studio-Projektmappen-Explorers klicken Sie mit der rechten Maustaste auf *mlsrv16.exe* und wählen dann "Eigenschaften".
- Ändern Sie den Debuggertyp vom automatischen in den gemischten oder verwalteten Modus, um sicherzustellen, dass Visual Studio nur in den .NET-Synchronisationsskripten nach Fehlern sucht. Ändern Sie in Visual Studio 2010 die Einstellung für "Debuggertyp" in "Verwaltet (v2.0, v1.1, v1.0)" bzw. "Verwaltet v4.0", abhängig von der Assembly-Version, die vom MobiLink-Server verwendet wird.

##### Hinweis

Um v4.0-Assemblys zu verwenden, müssen Sie die Option -clrVersion beim Laden des MobiLink-Servers explizit einbeziehen. Weitere Hinweise zur Option -clrVersion finden Sie unter „[mlsrv16-Option -sl dnet](#)“ auf Seite 78.

#### 5. Öffnen Sie die zugehörigen .NET-Quelldateien und setzen Sie Breakpoints.

Öffnen Sie die Quelldateien einzeln in der mlsrv16-Lösung. Öffnen Sie nicht die Original-Projektmappe oder -Projektdatei.

#### 6. Starten Sie MobiLink im Menü "Debuggen" oder indem Sie F5 drücken.

Speichern Sie *mlsrv16.sln*, wenn Sie dazu aufgefordert werden.

Wenn das Fenster "Keine symbolischen Informationen" angezeigt wird, klicken Sie auf **OK**, um die Fehlersuche dennoch auszuführen. Sie durchsuchen die verwalteten, von MobiLink aufgerufenen .NET-Synchronisationsskripten nach Fehlern und nicht den MobiLink-Server selbst.

#### 7. Führen Sie eine Synchronisation aus, die bewirkt, dass der Code von MobiLink mit einem Breakpoint ausgeführt wird.

## Ergebnisse

Ein Debugging für das Skript ist möglich.

## .NET-Synchronisationmethoden

In diesem Abschnitt werden die Methoden beschrieben, mit denen Sie allgemeine .NET-Synchronisationsaufgaben ausführen können.

## Zeilen-Uploads und -Downloads

Informationen über den Upload oder Download von Zeilen über .NET finden Sie unter „[Direkte Zeilenbehandlung](#)“ auf Seite 627.

## Gemeinsam genutzte Assemblys

Dieser Abschnitt beschreibt Optionen für das Laden von .NET-Assemblys sowie das Verfahren zum Laden gemeinsam genutzter Assemblys.

## Laden von .NET-Assemblys

Eine .NET-Assembly ist ein Paket von Typen, Metadaten und ausführbarem Code. In .NET-Anwendungen muss der gesamte Code in einer Assembly enthalten sein. Assembly-Dateien haben die Erweiterung *.dll* oder *.exe*.

Es gibt zwei Arten von Assemblys:

- **Private Assemblys** Eine private Assembly ist eine Datei im Dateisystem.
- **Gemeinsam genutzte Assemblys** Eine gemeinsam verwendete Assembly ist eine Assembly, die im globalen Assembly-Cache installiert ist.

Damit MobiLink eine Klasse laden und eine Methode dieser Klasse aufrufen kann, muss das Programm die Position der Assembly ermitteln, die die Klasse enthält. MobiLink braucht nur die Position der Assembly zu ermitteln, die direkt aufgerufen wird. Die Assembly kann dann ihrerseits jede beliebige andere erforderliche Assembly aufrufen.

Beispiel: MobiLink ruft "MeineAssembly" und "MeineAssembly" ruft "UtilityAssembly" und "NetworkingUtilsAssembly" auf. In dieser Situation muss MobiLink nur für den Aufruf von "MeineAssembly" konfiguriert werden.

MobiLink bietet zwei Möglichkeiten für das Laden von Assemblys:

- **-sl dnet verwenden ( -MLAutoLoadPath )** Diese Option kann nur für private Assemblys verwendet werden. Sie setzt den Pfad auf das Stammverzeichnis der Anwendung und lädt alle in diesem Verzeichnis enthaltenen Assemblys.

Wenn Sie die Option `-MLAutoLoadPath` verwenden, können Sie im Ereignisskript bei der Eingabe eines voll qualifizierten Methodennamens keine Domäne angeben.

Wenn Sie einen Pfad und ein Verzeichnis mit `-MLAutoLoadPath` festlegen, geht MobiLink wie folgt vor:

- Dieser Pfad wird als Basispfad der Anwendung festgesetzt.
- Alle Klassen in allen Dateien mit der Erweiterung `.dll` oder `.exe` in dem angegebenen Verzeichnis werden geladen.
- Eine Anwendungsdomäne wird erstellt, in die alle Benutzerklassen geladen werden, für die keine Domäne festgelegt wurde.

Assemblys im globalen Assembly-Cache können mit dieser Option nicht direkt aufgerufen werden. Diese gemeinsam genutzten Assemblys können mit `-MLDomConfigFile` aufgerufen werden.

- **-sl dnet verwenden ( -MLDomConfigFile )** Diese Option kann für private und gemeinsam genutzte Assemblys verwendet werden. Sie erfordert eine Konfigurationsdatei, die Domänen- und Assembly-Einstellungen enthält. Verwenden Sie diese Option, wenn Sie gemeinsam genutzte Assemblys haben und nicht alle Assemblys aus dem Anwendungsstammpfad laden möchten oder wenn Sie aus irgendeinem anderen Grund eine Konfigurationsdatei verwenden müssen.

Wenn diese Option verwendet wird, liest MobiLink die Einstellungen in der Konfigurationsdatei der angegebenen Domäne. Eine Domänen-Konfigurationsdatei enthält Konfigurationseinstellungen für eine oder mehrere .NET-Domänen. Wenn mehrere Domänen in der Datei angegeben sind, wird die erste von ihnen als Standarddomäne verwendet. (Die Standarddomäne wird verwendet, wenn für ein Skript keine Domäne festgelegt wurde.)

MobiLink versucht zunächst, eine Assembly als private Assembly zu laden. Schlägt dies fehl, wird sie aus dem globalen Assembly-Cache eingelesen. Private Assemblys müssen im Anwendungsstammverzeichnis untergebracht sein. Gemeinsam genutzte Assemblys werden aus dem globalen Assembly-Cache geladen.

Mit der Option `-MLDomConfigFile` können nur in der Domänen-Konfigurationsdatei angegebene Assemblys direkt aus Ereignisskripten aufgerufen werden.

### Beispiel einer Domänen-Konfigurationsdatei

Mit MobiLink wird ein Beispiel einer Domänen-Konfigurationsdatei namens `mlDomConfig.xml` installiert. Sie können Ihre eigene Datei völlig neu schreiben oder die Beispieldatei ändern, um sie an Ihre Anforderungen anzupassen. Die Beispieldatei befindet sich im SQL Anywhere-Pfad in

`MobiLink\setup\dnet\mlDomConfig.xml`

Der Inhalt der Beispiel-Domänen-Konfigurationsdatei `mlDomConfig.xml` lautet wie folgt:

```
<?xml version="1.0" encoding="utf-8"?>
<config xmlns="iAnywhere.MobiLink.mlDomConfig"
xsi:schemaLocation='iAnywhere.MobiLink.mlDomConfig mlDomConfig.xsd'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' >
  <domain>
```

```
<name>SampleDomain1</name>
<appBase>C:\scriptsDir</appBase>
<configFile></configFile>
<assembly name="Assembly1" />
<assembly name="Assembly2" />
</domain>

<domain>
  <name>SampleDomain2</name>
  <appBase>\Dom2assembly</appBase>
  <configFile>\Dom2assembly\AssemblyRedirects.config</configFile>
  <assembly name="Assembly3" />
  <assembly name="Assembly4" />
</domain>
</config>
```

Im Folgenden wird der Inhalt der Datei *mlDomConfig.xml* beschrieben:

- **name** Der Domänenname, der bei der Angabe der Domäne in einem Ereignisskript verwendet wird. Ein Ereignisskript mit dem Format "DomainName:Namespace.Class.Method" setzt voraus, dass die Domäne mit dem Namen DomainName in der Domänen-Konfigurationsdatei enthalten ist.

Sie müssen mindestens einen Domänennamen angeben.

- **appBase** Das Verzeichnis, das die Domäne als Anwendungsstammverzeichnis verwenden soll. Alle privaten Assemblys werden vom .NET-CLR basierend auf diesem Verzeichnis geladen. Sie müssen "appBase" angeben.
- **configFile** Die Konfigurationsdatei der .NET-Anwendung, die für die Domäne verwendet werden soll. Diese Angabe kann leer bleiben. Sie wird gewöhnlich verwendet, um das Bindungs- und Ladeverhalten der Standard-Assembly zu ändern. Weitere Hinweise zu Anwendungs-Konfigurationsdateien finden Sie in der .NET-Dokumentation.
- **assembly** Der Name einer Assembly, die MobiLink beim Auflösen von Typenreferenzen in Ereignisskripten laden und suchen soll. Sie müssen mindestens eine Assembly angeben. Wenn eine Assembly in mehreren Domänen verwendet wird, muss sie in jeder Domäne als Assembly angegeben werden. Wenn die Assembly privat ist, muss sie im Anwendungsstammverzeichnis für die Domäne angegeben werden.

Weitere Hinweise zur mlsrv16-Option -sl dnet finden Sie unter „[mlsrv16-Option -sl dnet](#)“ auf Seite 78.

## .NET-Synchronisationsbeispiel

Das folgende Beispiel ändert eine vorhandene Anwendung, um zu beschreiben, wie Sie die .NET-Synchronisationslogik verwenden können, um auf das Ereignis `authenticate_user` zu reagieren. Es erstellt für `authenticate_user` ein C#-Skript namens *AuthUser.cs*. Dieses Skript sucht das Kennwort des Benutzers in der Tabelle `user_pwd_table` und authentifiziert den Benutzer anhand dieses Kennworts.

1. Fügen Sie der Datenbank die Tabelle `user_pwd_table` hinzu. Führen Sie hierzu die folgende SQL-Anweisung in Interactive SQL aus:

```
CREATE TABLE user_pwd_table (
  user_name varchar(128) PRIMARY KEY NOT NULL,
```

```
        pwd          varchar(128)
    )
```

2. Fügen Sie der Tabelle anschließend einen Benutzer und ein Kennwort hinzu:

```
INSERT INTO user_pwd_table VALUES('user1', 'myPwd')
```

3. Erstellen Sie ein Verzeichnis für Ihre .NET-Assembly. Beispiel: *c:\mlexample*.
4. Erstellen Sie eine Datei mit dem Namen *AuthUser.cs* und folgendem Inhalt:

Siehe „[authenticate\\_user \(Verbindungsereignis\)](#)“ auf Seite 291.

```
using System;
using iAnywhere.MobiLink.Script;
namespace MLExample {

public class AuthClass {
    private DBConnection _conn;

    /// AuthClass constructor.
    public AuthClass(DBConnectionContext cc) {
        _conn = cc.GetConnection();
    }

    /// The DoAuthenticate method handles the 'authenticate_user'
    /// event.
    /// Note: This method does not handle password changes for
    /// advanced authorization status codes.
    public void DoAuthenticate(
        ref int authStatus,
        string user,
        string pwd,
        string newPwd)
    {
        DBCommand pwd_command = _conn.CreateCommand();
        pwd_command.CommandText = "select pwd from user_pwd_table"
            + " where user_name = ? ";
        pwd_command.Prepare();

        // Add a parameter for the user name.
        DBParameter user_param = new DBParameter();
        user_param.DbType = SQLType.SQL_CHAR;

        // Set the size for SQL_VARCHAR.
        user_param.Size = (uint) user.Length;
        user_param.Value = user;
        pwd_command.Parameters.Add(user_param);

        // Fetch the password for this user.
        DBRowReader rr = pwd_command.ExecuteReader();
        object[] pwd_row = rr.NextRow();

        if (pwd_row == null) {
            // User is unknown.
            authStatus = 4000;
        }
        else {
            if (((string) pwd_row[0]) == pwd) {
                // Password matched.
                authStatus = 1000;
            }
        }
    }
}
```

```
        else {  
            // Password did not match.  
            authStatus = 4000;  
        }  
    }  
    pwd_command.Close();  
    rr.Close();  
    return;  
}  
}
```

Die Methode `MLExample.AuthClass.DoAuthenticate` verarbeitet das Ereignis `authenticate_user`. Sie akzeptiert den Benutzernamen und das Kennwort und gibt einen Autorisierungsstatuscode zurück, der über den Erfolg bzw. Misserfolg der Prüfung Auskunft gibt.

5. Kompilieren Sie die Datei *AuthUser.cs*. Sie können hierzu die Befehlszeile oder Visual Studio verwenden.

Folgende Befehlszeile kompiliert z.B. *AuthUser.cs* und generiert eine Assembly namens *example.dll* in *c:\mlexample*.

```
csc /out:c:\mlexample\example.dll /target:library /reference:"%SQLANY16%\Assembly\V2\iAnywhere.MobiLink.Script.dll" AuthUser.cs
```

6. Registrieren Sie den .NET-Code für das Ereignis `authenticate_user`. Die Methode, die ausgeführt werden muss (`DoAuthenticate`), befindet sich im Namespace `MLExample` und der Klasse `AuthClass`. Führen Sie die folgenden SQL-Anweisungen aus:

```
CALL ml_add_dnet_connection_script('ex_version', 'authenticate_user',  
    'MLExample.AuthClass.DoAuthenticate');  
COMMIT
```

7. Anschließend starten Sie den MobiLink-Server mit der folgenden Option. Diese Option bewirkt, dass MobiLink alle Assemblys in *c:\mlexample* lädt:

```
-sl dnet (-MLAutoLoadPath=c:\mlexample)
```

Wenn nun ein Benutzer mit der Version `ex_version` synchronisiert, wird er mit dem Kennwort aus der Tabelle `user_pwd_table` authentifiziert.

## Referenz zur MobiLink-Server-API für .NET

In diesem Abschnitt werden die .NET-Schnittstellen und -Klassen für MobiLink beschrieben, sowie die ihnen zugeordneten Methoden, Eigenschaften und Konstruktoren. Zur Verwendung dieser Klassen referenzieren Sie die Assembly *iAnywhere.MobiLink.Script.dll*, die sich unter `%SQLANY16%\Assembly` befindet.

### Namespace

```
iAnywhere.MobiLink.Script
```

## DateTimeWithTimeZone-Klasse

Repräsentiert DateTime-Werte mit Zeitzonen-Offsets.

### Visual Basic-Syntax

```
Public NotInheritable Class DateTimeWithTimeZone
```

### C#-Syntax

```
public sealed class DateTimeWithTimeZone
```

### Mitglieder

Alle Mitglieder der DateTimeWithTimeZone-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">DateTimeWithTimeZone-Konstruktor</a>	Erstellt ein DateTimeWithTimeZone-Objekt mit den angegebenen Werten für Jahr, Monat, Tag, Stunde, Minute, Sekunde, Millisekunde, Zeitzonen-Stunde und Zeitzonen-Minute.
<a href="#">Parse-Methode</a>	Analysiert die angegebene Zeichenfolge syntaktisch und gibt ein neues DateTimeWithTimeZone-Objekt zurück.
<a href="#">ToString-Methode</a>	Ist äquivalent zu DateTime.ToString() mit angehängtem Zeitzonen-Offset.
<a href="#">DateTime-Eigenschaft</a>	Ruft den DateTime-Wert ab, der diesem ohne den Zeitzonen-Offset entspricht.
<a href="#">Day-Eigenschaft</a>	Ist äquivalent zu DateTime.Day.
<a href="#">Hour-Eigenschaft</a>	Ist äquivalent zu DateTime.Hour.
<a href="#">Millisecond-Eigenschaft</a>	Ist äquivalent zu DateTime.Millisecond.
<a href="#">Minute-Eigenschaft</a>	Ist äquivalent zu DateTime.Minute.
<a href="#">Month-Eigenschaft</a>	Ist äquivalent zu DateTime.Month.
<a href="#">Second-Eigenschaft</a>	Ist äquivalent zu DateTime.Second.
<a href="#">TimeZoneHour-Eigenschaft</a>	Ruft den Stundenteil des Zeitzonen-Offsets ab.
<a href="#">TimeZoneMinute-Eigenschaft</a>	Ruft den Minutenteil des Zeitzonen-Offsets ab.
<a href="#">Year-Eigenschaft</a>	Ist äquivalent zu DateTime.Year.

# DateTimeWithTimeZone-Konstruktor

Erstellt ein DateTimeWithTimeZone-Objekt mit den angegebenen Werten für Jahr, Monat, Tag, Stunde, Minute, Sekunde, Millisekunde, Zeitzonen-Stunde und Zeitzonen-Minute.

## Überladungsliste

Name	Beschreibung
<a href="#">DateTimeWithTimeZone(DateTime)-Konstruktor</a>	Erstellt ein DateTimeWithTimeZone-Objekt mit demselben Datum und derselben Uhrzeit wie der angegebene DateTime-Wert und Zeitzonen-Offsets von 0.
<a href="#">DateTimeWithTimeZone(DateTime, int, int)-Konstruktor</a>	Erstellt ein DateTimeWithTimeZone-Objekt mit demselben Datum und derselben Uhrzeit wie der angegebene DateTime-Wert und dem bereitgestellten Zeitzonen-Offset.
<a href="#">DateTimeWithTimeZone(int, int, int, int, int, int, int, int, int)-Konstruktor</a>	Erstellt ein DateTimeWithTimeZone-Objekt mit den angegebenen Werten für Jahr, Monat, Tag, Stunde, Minute, Sekunde, Millisekunde, Zeitzonen-Stunde und Zeitzonen-Minute.

## DateTimeWithTimeZone(DateTime)-Konstruktor

Erstellt ein DateTimeWithTimeZone-Objekt mit demselben Datum und derselben Uhrzeit wie der angegebene DateTime-Wert und Zeitzonen-Offsets von 0.

### Visual Basic-Syntax

```
Public Sub New(ByVal dt As Date)
```

### C#-Syntax

```
public DateTimeWithTimeZone(DateTime dt)
```

## DateTimeWithTimeZone(DateTime, int, int)-Konstruktor

Erstellt ein DateTimeWithTimeZone-Objekt mit demselben Datum und derselben Uhrzeit wie der angegebene DateTime-Wert und dem bereitgestellten Zeitzonen-Offset.

### Visual Basic-Syntax

```
Public Sub New(  
    ByVal dt As Date,  
    ByVal tz_hour As Integer,  
    ByVal tz_minute As Integer  
)
```

### C#-Syntax

```
public DateTimeWithTimeZone(DateTime dt, int tz_hour, int tz_minute)
```

**Parameter**

- **dt** Der Datums- und Uhrzeitteil.
- **tz\_hour** Der Stundenteil des Zeitzonen-Offsets. (-12 bis 14)
- **tz\_minute** Der Minutenteil des Zeitzonen-Offsets. (-59 bis 59). Dieser Wert kann nur negativ sein, wenn *tz\_hour* nicht positiv ist.

**Ausnahmebedingungen**

- **System.ArgumentOutOfRangeException** Wird ausgelöst, wenn *tz\_hour* oder *tz\_minute* außerhalb des zulässigen Bereichs liegt.

**DateTimeWithTimeZone(int, int, int, int, int, int, int, int, int)-Konstruktor**

Erstellt ein *DateTimeWithTimeZone*-Objekt mit den angegebenen Werten für Jahr, Monat, Tag, Stunde, Minute, Sekunde, Millisekunde, Zeitzonen-Stunde und Zeitzonen-Minute.

**Visual Basic-Syntax**

```
Public Sub New(  
    ByVal year As Integer,  
    ByVal month As Integer,  
    ByVal day As Integer,  
    ByVal hour As Integer,  
    ByVal minute As Integer,  
    ByVal second As Integer,  
    ByVal millisecond As Integer,  
    ByVal tz_hour As Integer,  
    ByVal tz_minute As Integer  
)
```

**C#-Syntax**

```
public DateTimeWithTimeZone(  
    int year,  
    int month,  
    int day,  
    int hour,  
    int minute,  
    int second,  
    int millisecond,  
    int tz_hour,  
    int tz_minute  
)
```

**Parameter**

- **year** Die Jahreszahl. (1 bis 9999)
- **month** Der Monat. (1 bis 12)
- **day** Der Tag. (1 bis Anzahl der Tage in *Monat* )

- **hour** Die Stunden. (0 bis 23)
- **minute** Die Minuten. (0 bis 59)
- **second** Die Sekunden. (0 bis 59)
- **millisecond** Die Millisekunden. (1 bis 999)
- **tz\_hour** Der Stundenteil des Zeitzonen-Offsets. (-12 bis 14)
- **tz\_minute** Der Minutenteil des Zeitzonen-Offsets. (-59 bis 59). Dieser Wert kann nur negativ sein, wenn *tz\_hour* nicht positiv ist.

### Ausnahmebedingungen

- **System.ArgumentOutOfRangeException** Wird ausgelöst, wenn ein Parameter außerhalb des zulässigen Bereichs liegt.

## Parse-Methode

Analysiert die angegebene Zeichenfolge syntaktisch und gibt ein neues `DateTimeWithTimeZone`-Objekt zurück.

### Visual Basic-Syntax

```
Public Shared Function Parse(  
    ByVal val As String  
) As DateTimeWithTimeZone
```

### C#-Syntax

```
public static DateTimeWithTimeZone Parse(string val)
```

### Parameter

- **val** Eine Zeichenfolge in der Form "JJJJ-MM TT HH:mm:ss.ffffff SHH:mm", wobei S für das Vorzeichen beim Stundenteil des Zeitzonen-Offsets steht und das zweite HH:mm der Zeitzonen-Offset ist. Der Bruchteil der Zeit kann weggelassen werden. Der Zeitzonen-Offset kann weggelassen werden. Wenn der Zeitzonen-Offset vorhanden ist, kann das Vorzeichen weggelassen werden.

### Ausnahmebedingungen

- **System.FormatException** Wird ausgelöst, wenn die angegebene Zeichenfolge nicht zum Format passt.

## ToString-Methode

Ist äquivalent zu `DateTime.ToString()` mit angehängtem Zeitzonen-Offset.

## Überladungsliste

Name	Beschreibung
<a href="#">ToString()-Methode</a>	Ist äquivalent zu <code>DateTime.ToString()</code> mit angehängtem Zeitzonen-Offset.
<a href="#">ToString(IFormatProvider)-Methode</a>	Ist äquivalent zu <code>DateTime.ToString()</code> mit angehängtem Zeitzonen-Offset.
<a href="#">ToString(string)-Methode</a>	Ist äquivalent zu <code>DateTime.ToString(format)</code> mit angehängtem Zeitzonen-Offset.
<a href="#">ToString(string, IFormatProvider)-Methode</a>	Ist äquivalent zu <code>DateTime.ToString(format, provider)</code> mit angehängtem Zeitzonen-Offset.

## ToString()-Methode

Ist äquivalent zu `DateTime.ToString()` mit angehängtem Zeitzonen-Offset.

### Visual Basic-Syntax

```
Public Overrides Function ToString() As String
```

### C#-Syntax

```
public override string ToString()
```

## ToString(IFormatProvider)-Methode

Ist äquivalent zu `DateTime.ToString()` mit angehängtem Zeitzonen-Offset.

### Visual Basic-Syntax

```
Public Function ToString(ByVal provider As IFormatProvider) As String
```

### C#-Syntax

```
public string ToString(IFormatProvider provider)
```

## ToString(string)-Methode

Ist äquivalent zu `DateTime.ToString(format)` mit angehängtem Zeitzonen-Offset.

### Visual Basic-Syntax

```
Public Function ToString(ByVal format As String) As String
```

### C#-Syntax

```
public string ToString(string format)
```

## **ToString(string, IFormatProvider)-Methode**

Ist äquivalent zu `DateTime.ToString(format, provider)` mit angehängtem Zeitzonen-Offset.

### **Visual Basic-Syntax**

```
Public Function ToString(  
    ByVal format As String,  
    ByVal provider As IFormatProvider  
) As String
```

### **C#-Syntax**

```
public string ToString(string format, IFormatProvider provider)
```

## **DateTime-Eigenschaft**

Ruft den `DateTime`-Wert ab, der diesem ohne den Zeitzonen-Offset entspricht.

### **Visual Basic-Syntax**

```
Public Property DateTime As Date
```

### **C#-Syntax**

```
public DateTime DateTime {get;set;}
```

## **Day-Eigenschaft**

Ist äquivalent zu `DateTime.Day`.

### **Visual Basic-Syntax**

```
Public ReadOnly Property Day As Integer
```

### **C#-Syntax**

```
public int Day {get;}
```

## **Hour-Eigenschaft**

Ist äquivalent zu `DateTime.Hour`.

### **Visual Basic-Syntax**

```
Public ReadOnly Property Hour As Integer
```

### **C#-Syntax**

```
public int Hour {get;}
```

## Millisecond-Eigenschaft

Ist äquivalent zu DateTime.Millisecond.

### Visual Basic-Syntax

```
Public ReadOnly Property Millisecond As Integer
```

### C#-Syntax

```
public int Millisecond {get;}
```

## Minute-Eigenschaft

Ist äquivalent zu DateTime.Minute.

### Visual Basic-Syntax

```
Public ReadOnly Property Minute As Integer
```

### C#-Syntax

```
public int Minute {get;}
```

## Month-Eigenschaft

Ist äquivalent zu DateTime.Month.

### Visual Basic-Syntax

```
Public ReadOnly Property Month As Integer
```

### C#-Syntax

```
public int Month {get;}
```

## Second-Eigenschaft

Ist äquivalent zu DateTime.Second.

### Visual Basic-Syntax

```
Public ReadOnly Property Second As Integer
```

### C#-Syntax

```
public int Second {get;}
```

## TimeZoneHour-Eigenschaft

Ruft den Stundenteil des Zeitzone-Offsets ab.

**Visual Basic-Syntax**

```
Public Property TimeZoneHour As Integer
```

**C#-Syntax**

```
public int TimeZoneHour {get;set;}
```

**TimeZoneMinute-Eigenschaft**

Ruft den Minutenteil des Zeitzonen-Offsets ab.

**Visual Basic-Syntax**

```
Public Property TimeZoneMinute As Integer
```

**C#-Syntax**

```
public int TimeZoneMinute {get;set;}
```

**Year-Eigenschaft**

Ist äquivalent zu DateTime.Year.

**Visual Basic-Syntax**

```
Public ReadOnly Property Year As Integer
```

**C#-Syntax**

```
public int Year {get;}
```

**DBCommand-Schnittstelle**

Repräsentiert eine SQL-Anweisung oder einen Datenbankbefehl.

**Visual Basic-Syntax**

```
Public Interface DBCommand
```

**C#-Syntax**

```
public interface DBCommand
```

**Mitglieder**

Alle Mitglieder der DBCommand-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">Close-Methode</a>	Schließt die aktuelle SQL-Anweisung oder den aktuellen Befehl.

Name	Beschreibung
<a href="#">ExecuteNonQuery-Methode</a>	Führt eine Anweisung aus, bei der es sich nicht um eine Abfrage handelt:
<a href="#">ExecuteReader-Methode</a>	Führt eine Abfrageanweisung aus und gibt die Ergebnismenge zurück.
<a href="#">Prepare-Methode</a>	Bereitet die in CommandText gespeicherte SQL-Anweisung für die Ausführung vor.
<a href="#">CommandText-Eigenschaft</a>	Die auszuführende SQL-Anweisung..
<a href="#">Parameters-Eigenschaft</a>	Ruft den Parameter DBParameterCollection für dieses DBCommand-Objekt ab.

### Bemerkungen

Bei DBCommand kann es sich um eine Aktualisierung oder eine Abfrage handeln.

### Beispiel

Beim folgenden C#-Code wird die DBCommand-Schnittstelle beispielsweise verwendet, um zwei Abfragen auszuführen:

```
DBCommand stmt = conn.CreateCommand();
stmt.CommandText = "SELECT t1a1, t1a2 FROM table1 ";

DBRowReader rs = stmt.ExecuteReader();
printResultSet( rs );
rs.Close();

stmt.CommandText = "SELECT t2a1 FROM table2 ";
rs = stmt.ExecuteReader();
printResultSet( rs );
rs.Close();
stmt.Close();
```

Beim folgenden C#-Code wird die DBCommand-Schnittstelle beispielsweise verwendet, um eine Aktualisierung mit Parametern auszuführen:

```
public void prepare_for_download(DateTime last_download,
    String ml_username)
{
    DBCommand cstmt = conn.CreateCommand();
    cstmt.CommandText = "CALL myProc( ?,?,? )";
    cstmt.Prepare();

    DBParameter param = new DBParameter();
    param.DbType      = SQLType.SQL_CHAR;
    param.Value       = "10000";
    cstmt.Parameters.Add(param);

    param              = new DBParameter();
    param.DbType       = SQLType.SQL_INTEGER;
    param.Value        = 20000;
    cstmt.Parameters.Add(param);
```

```
param                = new DBParameter();  
param.DbType         = SQLType.SQL_DECIMAL;  
param.Precision      = 5;  
param.Value          = new Decimal(30000);  
cstmt.Parameters.Add(param);  
  
// Execute update  
DBRowReader rset = cstmt.ExecuteNonQuery();  
cstmt.Close();  
}
```

## Close-Methode

Schließt die aktuelle SQL-Anweisung oder den aktuellen Befehl.

### Visual Basic-Syntax

```
Public Sub Close()
```

### C#-Syntax

```
public void Close()
```

## ExecuteNonQuery-Methode

Führt eine Anweisung aus, bei der es sich nicht um eine Abfrage handelt:

### Visual Basic-Syntax

```
Public Function ExecuteNonQuery() As Integer
```

### C#-Syntax

```
public int ExecuteNonQuery()
```

### Rückgabe

Die Anzahl der Zeilen in der Datenbank, die von der SQL-Anweisung betroffen sind.

## ExecuteReader-Methode

Führt eine Abfrageanweisung aus und gibt die Ergebnismenge zurück.

### Visual Basic-Syntax

```
Public Function ExecuteReader() As DBRowReader
```

### C#-Syntax

```
public DBRowReader ExecuteReader()
```

**Rückgabe**

Ein DBRowReader-Objekt für die Abfrage von Ergebnissen, die von der SQL-Anweisung zurückgegeben werden.

## Prepare-Methode

Bereitet die in CommandText gespeicherte SQL-Anweisung für die Ausführung vor.

**Visual Basic-Syntax**

```
Public Sub Prepare()
```

**C#-Syntax**

```
public void Prepare()
```

## CommandText-Eigenschaft

Die auszuführende SQL-Anweisung..

**Visual Basic-Syntax**

```
Public Property CommandText As String
```

**C#-Syntax**

```
public string CommandText {get;set;}
```

## Parameters-Eigenschaft

Ruft die Eigenschaft DBParameterCollection für dieses DBCommand-Objekt ab.

**Visual Basic-Syntax**

```
Public ReadOnly Property Parameters As DBParameterCollection
```

**C#-Syntax**

```
public DBParameterCollection Parameters {get;}
```

**Rückgabe**

Die angeforderte Parametersammlung.

**Siehe auch**

- [DBParameterCollection-Klasse \[MobiLink-Server .NET\] auf Seite 575](#)

## DBConnection-Schnittstelle

Repräsentiert eine MobiLink-ODBC-Verbindung.

### Visual Basic-Syntax

```
Public Interface DBConnection
```

### C#-Syntax

```
public interface DBConnection
```

### Mitglieder

Alle Mitglieder der DBConnection-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">Close-Methode</a>	Beendet die aktuelle Verbindung.
<a href="#">Commit-Methode</a>	Schreibt die aktuelle Transaktion fest.
<a href="#">CreateCommand-Methode</a>	Erstellt eine SQL-Anweisung oder einen Befehl für diese Verbindung.
<a href="#">Rollback-Methode</a>	Setzt die aktuelle Transaktion zurück.

### Bemerkungen

Diese Schnittstelle gestattet es der vom Benutzer geschriebenen Synchronisationslogik, auf eine von MobiLink erstellte ODBC-Verbindung zuzugreifen.

## Close-Methode

Beendet die aktuelle Verbindung.

### Visual Basic-Syntax

```
Public Sub Close()
```

### C#-Syntax

```
public void Close()
```

## Commit-Methode

Schreibt die aktuelle Transaktion fest.

### Visual Basic-Syntax

```
Public Sub Commit()
```

### C#-Syntax

```
public void Commit()
```

## CreateCommand-Methode

Erstellt eine SQL-Anweisung oder einen Befehl für diese Verbindung.

### Visual Basic-Syntax

```
Public Function CreateCommand() As DBCommand
```

### C#-Syntax

```
public DBCommand CreateCommand()
```

### Rückgabe

Das neu generierte DBCommand-Objekt.

## Rollback-Methode

Setzt die aktuelle Transaktion zurück.

### Visual Basic-Syntax

```
Public Sub Rollback()
```

### C#-Syntax

```
public void Rollback()
```

## DBConnectionContext-Schnittstelle

Ruft Informationen über die aktuelle Datenbankverbindung ab.

### Visual Basic-Syntax

```
Public Interface DBConnectionContext
```

### C#-Syntax

```
public interface DBConnectionContext
```

### Mitglieder

Alle Mitglieder der DBConnectionContext-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">getConnection-Methode</a>	Gibt die vorhandene Verbindung zur konsolidierten MobiLink-Datenbank zurück.
<a href="#">GetDownloadData-Methode</a>	Gibt das DownloadData-Objekt für die aktuelle Synchronisation zurück.

Name	Beschreibung
<a href="#">GetProperties-Methode</a>	Gibt eine Sammlung von Eigenschaften zurück, basierend auf der Skriptversion dieser Verbindung.
<a href="#">GetRemoteID-Methode</a>	Gibt die entfernte ID der Datenbank zurück, die zur Zeit über diese Verbindung synchronisiert wird.
<a href="#">GetServerContext-Methode</a>	Gibt den aktuellen Serverkontext zurück.
<a href="#">GetVersion-Methode</a>	Gibt die Versionszeichenfolge für diese Verbindung zurück.
<a href="#">NetworkData-Eigenschaft</a>	Gibt Informationen über die Netzwerkdatenströme für eine Synchronisation zurück.

### Bemerkungen

Diese Informationen werden an den Konstruktor von Klassen übergeben, die Skripten enthalten. Wenn Kontext für einen Hintergrund-Thread oder über die Verbindungsdauer hinaus benötigt wird, verwenden Sie ServerContext.

Weitere Hinweise zu Konstruktoren finden Sie unter [„Konstruktoren“ auf Seite 541](#).

#### Hinweis

Eine DBConnectionContext-Instanz sollte nicht außerhalb des Threads verwendet werden, der Ihren .NET-Code aufruft.

### Beispiel

Im folgenden Beispiel wird gezeigt, wie Sie eine DBConnectionContext-Instanz auf Klassenebene erstellen, die in Ihren Synchronisationsskripten verwendet werden kann. Die getConnection-Methode der DBConnectionContext-Schnittstelle ruft eine DBConnection-Instanz ab, die die aktuelle Verbindung zur konsolidierten MobiLink-Datenbank darstellt.

```
using iAnywhere.MobiLink.Script;
using System.Data;

public class OrderProcessor {
    DBConnectionContext _cc;

    public OrderProcessor( DBConnectionContext cc ) {
        _cc = cc;
    }

    // The method used for the handle_DownloadData event.
    public void HandleEvent() {
        DBConnection my_connection = _cc.GetConnection();
        // ...
    }

    // ...
}
```

## getConnection-Methode

Gibt die vorhandene Verbindung zur konsolidierten MobiLink-Datenbank zurück.

### Visual Basic-Syntax

```
Public Function GetConnection() As DBConnection
```

### C#-Syntax

```
public DBConnection GetConnection()
```

### Rückgabe

Die aktuelle Verbindung mit der konsolidierten Datenbank. Diese Verbindung gilt nur für die Dauer der zugrunde liegenden MobiLink-Verbindung.

### Bemerkungen

Die Verbindung ist dieselbe, die MobiLink zum Ausführen von SQL-Skripten verwendet. Sie darf nicht in einer Weise festgeschrieben, getrennt oder geändert werden, die die Nutzung der Verbindung durch den MobiLink-Server beeinträchtigen kann.

Verwenden Sie die Verbindung nicht, nachdem das Ereignis `end_connection` für die Verbindung aufgerufen wurde.

Verwenden Sie die `MakeConnection`-Methode, wenn eine Serververbindung mit vollständigem Zugriff erforderlich ist.

### Siehe auch

- [ServerContext.MakeConnection-Methode \[MobiLink-Server .NET\] auf Seite 607](#)

## GetDownloadData-Methode

Gibt das `DownloadData`-Objekt für die aktuelle Synchronisation zurück.

### Visual Basic-Syntax

```
Public Function GetDownloadData() As DownloadData
```

### C#-Syntax

```
public DownloadData GetDownloadData()
```

### Rückgabe

Das `DownloadData`-Objekt für die aktuelle Synchronisation; sonst der Wert `NULL`, wenn die Synchronisation ein reiner Upload ist.

### Bemerkungen

Verwenden Sie die `DownloadData`-Instanz, um den Download für direkte Zeilenbehandlung zu erstellen.

## Beispiel

Im folgenden Beispiel wird vorausgesetzt, dass eine DBConnectionContext-Instanz namens `_cc` vorhanden ist:

```
// The method used for the handle_DownloadData event.
public void HandleDownload() {
    // Get the DownloadData for the current synchronization.
    DownloadData my_dd = _cc.GetDownloadData();

    // Get an array of tables to set download operations.
    DownloadTableData[] download_tables = my_dd.GetDownloadTables();

    // Get the first table in the DownloadTableData array.
    DownloadTableData my_download_table = download_tables[0];

    // ...
}
```

## GetProperties-Methode

Gibt eine Sammlung von Eigenschaften zurück, basierend auf der Skriptversion dieser Verbindung.

### Visual Basic-Syntax

```
Public Function GetProperties() As NameValueCollection
```

### C#-Syntax

```
public NameValueCollection GetProperties()
```

### Rückgabe

Die Eigenschaften für die aktuelle Skriptversion.

### Bemerkungen

Die Eigenschaften werden in der Tabelle `ml_property` gespeichert. Weitere Hinweise finden Sie unter [„ml\\_add\\_property-Systemprozedur“ auf Seite 664](#).

## Beispiel

Das folgende Beispiel zeigt, wie Sie die Eigenschaften für DBConnectionContext ausgeben. Bei diesem Beispiel wird vorausgesetzt, dass eine DBConnectionContext-Instanz namens `_cc` vorhanden ist.

```
// The method used to output the connection properties.
public void OutputProperties() {
    // output the Properties for the current synchronization
    NameValueCollection properties = _cc.GetProperties();
    System.Console.WriteLine(properties.ToString());
}
```

## GetRemoteID-Methode

Gibt die entfernte ID der Datenbank zurück, die zur Zeit über diese Verbindung synchronisiert wird.

**Visual Basic-Syntax**

```
Public Function GetRemoteID() As String
```

**C#-Syntax**

```
public string GetRemoteID()
```

**Rückgabe**

Die entfernte ID.

**Bemerkungen**

Weitere Hinweise zu entfernten IDs finden Sie unter „[Entfernte IDs](#)“ [*MobiLink - Clientadministration*].

**Beispiel**

Das folgende Beispiel zeigt, wie Sie die entfernte ID für DBConnectionContext ausgeben.

```
// The method used to output the remote ID.  
public void OutputRemoteID() {  
    // output the Remote ID for the current synchronization  
    string remoteID = _cc.GetRemoteID();  
    System.Console.WriteLine(remoteID);  
}
```

## GetServerContext-Methode

Gibt den aktuellen Serverkontext zurück.

**Visual Basic-Syntax**

```
Public Function GetServerContext() As ServerContext
```

**C#-Syntax**

```
public ServerContext GetServerContext()
```

**Rückgabe**

Der Serverkontext für diesen MobiLink-Server.

**Bemerkungen**

Mit dieser Methode können Sie neue Verbindungen erstellen oder mit Boot-Klassen interagieren.

**Beispiel**

Das folgende Beispiel zeigt, wie Sie die ServerContext-Instanz für DBConnectionContext abrufen und den Server herunterfahren:

```
// A method that uses an instance of the ServerContext to shut down the  
server  
public void ShutDownServer() {  
    ServerContext context = _cc.GetServerContext();  
    context.Shutdown();  
}
```

## GetVersion-Methode

Gibt die Versionszeichenfolge für diese Verbindung zurück.

### Visual Basic-Syntax

```
Public Function GetVersion() As String
```

### C#-Syntax

```
public string GetVersion()
```

### Rückgabe

Der Skriptversionsname.

### Bemerkungen

Die Eigenschaften werden in der Tabelle ml\_property gespeichert. Weitere Hinweise finden Sie unter [„ml\\_add\\_property-Systemprozedur“ auf Seite 664](#).

### Beispiel

Das folgende Beispiel zeigt, wie Sie die Skriptversion abrufen und für Entscheidungen heranziehen:

```
public void MyEvent() {  
    // ...  
  
    string version = _cc.GetVersion();  
    switch( version ) {  
        case "My Version 1":  
            // ...  
            break;  
        case "My Version 2":  
            // ...  
            break;  
    }  
}
```

## NetworkData-Eigenschaft

Gibt Informationen über die Netzwerkdatenströme für eine Synchronisation zurück.

### Visual Basic-Syntax

```
Public ReadOnly Property NetworkData As NetworkData
```

### C#-Syntax

```
public NetworkData NetworkData {get;}
```

### Rückgabe

Informationen über die Netzwerkdatenströme, die für die Anforderung verwendet werden, oder NULL, wenn die Sammlung nicht aktiviert wurde.

## Bemerkungen

Diese Methode ist nützlich, wenn die Benutzerauthentifizierung bei einem anderen Server im Unternehmen erfolgt, der das clientseitige Zertifikat und HTTP-Header verwendet.

Zum Aktivieren einer Sammlung von Netzwerk-Datenströmen fügen Sie den -x Parametern `collect_network_data = 1` hinzu. Diese Option fügt für die Synchronisation zusätzlichen Speicher-Overhead zum Speichern der Daten hinzu.

## Siehe auch

- [NetworkData-Schnittstelle \[MobiLink-Server-API für .NET\] auf Seite 597](#)

## DBParameter-Klasse

Repräsentiert einen gebundenen ODBC-Parameter.

### Visual Basic-Syntax

```
Public Class DBParameter
```

### C#-Syntax

```
public class DBParameter
```

## Mitglieder

Alle Mitglieder der DBParameter-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">DbType-Eigenschaft</a>	Der SQLType-Wert dieses Parameters.
<a href="#">Direction-Eigenschaft</a>	Die Eingabe-/Ausgaberrichtung dieses Parameters.
<a href="#">IsNullable-Eigenschaft</a>	TRUE, wenn der Parameter NULL sein kann, sonst FALSE.
<a href="#">ParameterName-Eigenschaft</a>	Der Name dieses Parameters.
<a href="#">Precision-Eigenschaft</a>	Die dezimale Gesamtstellenzahl dieses Parameters.
<a href="#">Scale-Eigenschaft</a>	Die Dezimalstellen dieses Parameters.
<a href="#">Size-Eigenschaft</a>	Die Größe dieses Parameters, gemessen in Byte.
<a href="#">Value-Eigenschaft</a>	Der Wert dieses Parameters.
<a href="#">HasChanged-Feld</a>	Gibt die Information zurück, ob der Parameter seit seiner Erstellung geändert wurde.

## Bemerkungen

Diese Klasse ist erforderlich, um Befehle mit Parametern auszuführen. Alle Parameter müssen vorhanden sein, bevor der Befehl ausgeführt wird.

## Beispiel

Beim folgenden C#-Code wird die DBCommand-Schnittstelle beispielsweise verwendet, um eine Aktualisierung mit Parametern auszuführen:

```
using( DBCommand cstmt = conn.CreateCommand() ) {
    DBCommand cstmt = conn.CreateCommand();

    cstmt.CommandText = "call myProc( ?,?,?,? )";

    cstmt.Prepare();

    DBParameter param = new DBParameter();
    param.DbType      = SQLType.SQL_CHAR;
    param.Value       = "10000";
    cstmt.Parameters.Add( param );

    param              = new DBParameter();
    param.DbType       = SQLType.SQL_INTEGER;
    param.Value        = 20000;
    cstmt.Parameters.Add( param );

    param              = new DBParameter();
    param.DbType       = SQLType.SQL_DECIMAL;
    param.Precision    = 5;
    param.Value        = new Decimal( 30000 );
    cstmt.Parameters.Add( param );

    // Execute update
    DBRowReader rset = cstmt.ExecuteNonQuery();
    cstmt.Close();
}
```

## DbType-Eigenschaft

Der SQLType-Wert dieses Parameters.

## Visual Basic-Syntax

```
Public Property DbType As SQLType
```

## C#-Syntax

```
public SQLType DbType {get;set;}
```

## Bemerkungen

Der Standardwert lautet SQLType.SQL\_TYPE\_NULL.

## Direction-Eigenschaft

Die Eingabe-/Ausgeberichtung dieses Parameters.

**Visual Basic-Syntax**

```
Public Property Direction As ParameterDirection
```

**C#-Syntax**

```
public ParameterDirection Direction {get;set;}
```

**Bemerkungen**

Der Standardwert lautet ParameterDirection.Input.

## IsNullable-Eigenschaft

TRUE, wenn der Parameter NULL sein kann, sonst FALSE.

**Visual Basic-Syntax**

```
Public Property IsNullable As Boolean
```

**C#-Syntax**

```
public bool IsNullable {get;set;}
```

**Bemerkungen**

Der Standardwert lautet FALSE.

## ParameterName-Eigenschaft

Der Name dieses Parameters.

**Visual Basic-Syntax**

```
Public Property ParameterName As String
```

**C#-Syntax**

```
public string ParameterName {get;set;}
```

**Bemerkungen**

Der Standardwert ist NULL.

## Precision-Eigenschaft

Die dezimale Gesamtstellenzahl dieses Parameters.

**Visual Basic-Syntax**

```
Public Property Precision As UInteger
```

**C#-Syntax**

```
public uint Precision {get;set;}
```

### Bemerkungen

Diese Eigenschaft wird nur für die Parameter `SQLType.SQL_NUMERIC` und `SQLType.SQL_DECIMAL` verwendet.

Standardwert ist "0".

## Scale-Eigenschaft

Die Dezimalstellen dieses Parameters.

### Visual Basic-Syntax

```
Public Property Scale As Short
```

### C#-Syntax

```
public short Scale {get;set;}
```

### Bemerkungen

Diese Eigenschaft wird nur für die Parameter `SQLType.SQL_NUMERIC` und `SQLType.SQL_DECIMAL` verwendet.

Standardwert ist "0".

## Size-Eigenschaft

Die Größe dieses Parameters, gemessen in Byte.

### Visual Basic-Syntax

```
Public Property Size As UInteger
```

### C#-Syntax

```
public uint Size {get;set;}
```

### Bemerkungen

Der Standardwert wird von `DbType` abgeleitet.

## Value-Eigenschaft

Der Wert dieses Parameters.

### Visual Basic-Syntax

```
Public Property Value As Object
```

### C#-Syntax

```
public object Value {get;set;}
```

## Bemerkungen

Der Standardwert ist NULL.

## HasChanged-Feld

Gibt die Information zurück, ob der Parameter seit seiner Erstellung geändert wurde.

### Visual Basic-Syntax

```
Public HasChanged As Boolean
```

### C#-Syntax

```
public bool HasChanged;
```

## DBParameterCollection-Klasse

Sammlung von DBParameter-Objekten.

### Visual Basic-Syntax

```
Public Class DBParameterCollection
    Implements System.Data.IDataParameterCollection
    Implements System.Collections.IList
    Implements System.Collections.ICollection
    Implements System.Collections.IEnumerable
```

### C#-Syntax

```
public class DBParameterCollection :
    System.Data.IDataParameterCollection,
    System.Collections.IList,
    System.Collections.ICollection,
    System.Collections.IEnumerable
```

### Basisklassen

- [System.Data.IDataParameterCollection](#)
- [System.Collections.IList](#)
- [System.Collections.ICollection](#)
- [System.Collections.IEnumerable](#)

### Mitglieder

Alle Mitglieder der DBParameterCollection-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">DBParameterCollection-Konstruktor</a>	Erstellt eine leere Liste von DBParameter-Objekten.
<a href="#">Add-Methode</a>	Fügt der Sammlung den angegebenen Parameter hinzu.

Name	Beschreibung
<a href="#">Clear-Methode</a>	Entfernt alle Parameter aus der Sammlung.
<a href="#">Contains-Methode</a>	Überprüft, ob die Sammlung einen Parameter mit dem angegebenen Namen enthält.
<a href="#">CopyTo-Methode</a>	Kopiert den Inhalt der Sammlung in das angegebene Array, beginnend bei dem angegebenen Index.
<a href="#">GetEnumerator-Methode</a>	Gibt einen Enumerator für die Sammlung zurück.
<a href="#">IndexOf-Methode</a>	Gibt den Index des Parameters mit dem angegebenen Namen in der Sammlung zurück
<a href="#">Insert-Methode</a>	Fügt das angegebene DBParameter-Objekt am angegebenen Index in die Sammlung ein.
<a href="#">Remove-Methode</a>	Entfernt das angegebene DBParameter-Objekt aus der Sammlung.
<a href="#">RemoveAt-Methode</a>	Entfernt den Parameter mit dem angegebenen Namen aus der Sammlung.
<a href="#">Count-Eigenschaft</a>	Die Anzahl der Parameter in der Sammlung.
<a href="#">IsFixedSize-Eigenschaft</a>	Gibt FALSE zurück.
<a href="#">IsReadOnly-Eigenschaft</a>	Gibt FALSE zurück.
<a href="#">IsSynchronized-Eigenschaft</a>	Gibt FALSE zurück.
<a href="#">SyncRoot-Eigenschaft</a>	Dient zum Synchronisieren des Zugriffs auf das DBParameterCollection-Objekt.
<a href="#">this-Eigenschaft</a>	Ruft das DBParameter-Objekt am angegebenen Index in der Sammlung ab oder legt es fest.

### Bemerkungen

Ein DBParamterCollection-Objekt ist während der Erstellung durch DBCommand anfangs leer und muss mit geeigneten Parametern gefüllt werden, bevor der DBCommand-Befehl ausgeführt wird.

## DBParameterCollection-Konstruktor

Erstellt eine leere Liste von DBParameter-Objekten.

### Visual Basic-Syntax

```
Public Sub New()
```

**C#-Syntax**

```
public DBParameterCollection()
```

## Add-Methode

Fügt der Sammlung den angegebenen Parameter hinzu.

**Visual Basic-Syntax**

```
Public Function Add(ByVal value As Object) As Integer
```

**C#-Syntax**

```
public int Add(object value)
```

**Parameter**

- **value** Das DBParameter-Objekt, das der Sammlung hinzugefügt werden soll

**Rückgabe**

Gibt den Index des Parameters zurück, der der Sammlung hinzugefügt wurde.

**Siehe auch**

- [DBParameter-Klasse \[MobiLink-Server .NET\] auf Seite 571](#)

## Clear-Methode

Entfernt alle Parameter aus der Sammlung.

**Visual Basic-Syntax**

```
Public Sub Clear()
```

**C#-Syntax**

```
public void Clear()
```

## Contains-Methode

Überprüft, ob die Sammlung einen Parameter mit dem angegebenen Namen enthält.

**Überladungsliste**

Name	Beschreibung
<a href="#">Contains(Object)-Methode</a>	Gibt TRUE zurück, wenn die Sammlung das angegebene DBParameter-Objekt enthält.
<a href="#">Contains(String)-Methode</a>	Überprüft, ob die Sammlung einen Parameter mit dem angegebenen Namen enthält.

## Contains(Object)-Methode

Gibt TRUE zurück, wenn die Sammlung das angegebene DBParameter-Objekt enthält.

### Visual Basic-Syntax

```
Public Function Contains(ByVal value As Object) As Boolean
```

### C#-Syntax

```
public bool Contains(object value)
```

### Parameter

- **value** Das DBParameter-Objekt, auf das geprüft werden soll.

### Rückgabe

TRUE, wenn diese Sammlung das DBParameter-Objekt enthält, sonst FALSE.

### Siehe auch

- [DBParameter-Klasse \[MobiLink-Server .NET\] auf Seite 571](#)

## Contains(String)-Methode

Überprüft, ob die Sammlung einen Parameter mit dem angegebenen Namen enthält.

### Visual Basic-Syntax

```
Public Function Contains(ByVal parameterName As String) As Boolean
```

### C#-Syntax

```
public bool Contains(string parameterName)
```

### Parameter

- **parameterName** Der Name des zu suchenden Parameters.

### Rückgabe

TRUE, wenn diese Sammlung einen Parameter mit dem angegebenen Namen enthält, sonst FALSE.

## CopyTo-Methode

Kopiert den Inhalt der Sammlung in das angegebene Array, beginnend bei dem angegebenen Index.

### Visual Basic-Syntax

```
Public Sub CopyTo(ByVal array As Array, ByVal index As Integer)
```

### C#-Syntax

```
public void CopyTo(Array array, int index)
```

**Parameter**

- **array** Das Array, in das der Inhalt der Sammlung kopiert wird.
- **index** Der Index im Array, an dem der Inhalt der Sammlung in das Array kopiert werden soll.

## GetEnumerator-Methode

Gibt einen Enumerator für die Sammlung zurück.

**Visual Basic-Syntax**

```
Public Function GetEnumerator() As System.Collections.IEnumerator
```

**C#-Syntax**

```
public System.Collections.IEnumerator GetEnumerator()
```

**Rückgabe**

Ein Enumerator für die Sammlung.

## IndexOf-Methode

Gibt den Index des Parameters mit dem angegebenen Namen in der Sammlung zurück

**Überladungsliste**

Name	Beschreibung
<a href="#">IndexOf(Object)-Methode</a>	Gibt den Index des angegebenen DBParameter-Objekts in der Sammlung zurück.
<a href="#">IndexOf(String)-Methode</a>	Gibt den Index des Parameters mit dem angegebenen Namen in der Sammlung zurück

## IndexOf(Object)-Methode

Gibt den Index des angegebenen DBParameter-Objekts in der Sammlung zurück.

**Visual Basic-Syntax**

```
Public Function IndexOf(ByVal value As Object) As Integer
```

**C#-Syntax**

```
public int IndexOf(object value)
```

**Parameter**

- **value** Das DBParameter-Objekt, das gesucht werden soll

## Rückgabe

Der Index des DBParameter-Objekts in der Sammlung.

## Siehe auch

- [DBParameter-Klasse \[MobiLink-Server .NET\] auf Seite 571](#)

## IndexOf(String)-Methode

Gibt den Index des Parameters mit dem angegebenen Namen in der Sammlung zurück

### Visual Basic-Syntax

```
Public Function IndexOf(ByVal parameterName As String) As Integer
```

### C#-Syntax

```
public int IndexOf(string parameterName)
```

### Parameter

- **parameterName** Der Name des zu suchenden Parameters.

## Rückgabe

Der Index des Parameters oder -1, wenn kein Parameter mit dem angegebenen Namen vorhanden ist.

## Insert-Methode

Fügt das angegebene DBParameter-Objekt am angegebenen Index in die Sammlung ein.

### Visual Basic-Syntax

```
Public Sub Insert(ByVal index As Integer, ByVal value As Object)
```

### C#-Syntax

```
public void Insert(int index, object value)
```

### Parameter

- **value** Das DBParameter-Objekt, das eingefügt werden soll
- **Index** Der Index, an dem der Wert eingefügt werden soll.

## Siehe auch

- [DBParameter-Klasse \[MobiLink-Server .NET\] auf Seite 571](#)

## Remove-Methode

Entfernt das angegebene DBParameter-Objekt aus der Sammlung.

### Visual Basic-Syntax

```
Public Sub Remove(ByVal value As Object)
```

### C#-Syntax

```
public void Remove(object value)
```

### Parameter

- **value** Das DBParameter-Objekt, das entfernt werden soll.

### Siehe auch

- [DBParameter-Klasse \[MobiLink-Server .NET\] auf Seite 571](#)

## RemoveAt-Methode

Entfernt den Parameter mit dem angegebenen Namen aus der Sammlung.

### Überladungsliste

Name	Beschreibung
<a href="#">RemoveAt(Int)-Methode</a>	Entfernt das DBParameter-Objekt am angegebenen Index in der Sammlung.
<a href="#">RemoveAt(String)-Methode</a>	Entfernt den Parameter mit dem angegebenen Namen aus der Sammlung.

### RemoveAt(Int)-Methode

Entfernt das DBParameter-Objekt am angegebenen Index in der Sammlung.

### Visual Basic-Syntax

```
Public Sub RemoveAt(ByVal index As Integer)
```

### C#-Syntax

```
public void RemoveAt(int index)
```

### Parameter

- **index** Der Index des DBParameter-Objekts, das entfernt werden soll.

### Siehe auch

- [DBParameter-Klasse \[MobiLink-Server .NET\] auf Seite 571](#)

## RemoveAt(String)-Methode

Entfernt den Parameter mit dem angegebenen Namen aus der Sammlung.

### Visual Basic-Syntax

```
Public Sub RemoveAt(ByVal parameterName As String)
```

### C#-Syntax

```
public void RemoveAt(string parameterName)
```

### Parameter

- **parameterName** Der Name des zu entfernenden Parameters.

## Count-Eigenschaft

Die Anzahl der Parameter in der Sammlung.

### Visual Basic-Syntax

```
Public ReadOnly Property Count As Integer
```

### C#-Syntax

```
public int Count {get;}
```

## IsFixedSize-Eigenschaft

Gibt FALSE zurück.

### Visual Basic-Syntax

```
Public ReadOnly Property IsFixedSize As Boolean
```

### C#-Syntax

```
public bool IsFixedSize {get;}
```

## IsReadOnly-Eigenschaft

Gibt FALSE zurück.

### Visual Basic-Syntax

```
Public ReadOnly Property IsReadOnly As Boolean
```

### C#-Syntax

```
public bool IsReadOnly {get;}
```

## IsSynchronized-Eigenschaft

Gibt FALSE zurück.

**Visual Basic-Syntax**

```
Public ReadOnly Property IsSynchronized As Boolean
```

**C#-Syntax**

```
public bool IsSynchronized {get;}
```

**SyncRoot-Eigenschaft**

Dient zum Synchronisieren des Zugriffs auf das DBParameterCollection-Objekt.

**Visual Basic-Syntax**

```
Public ReadOnly Property SyncRoot As Object
```

**C#-Syntax**

```
public object SyncRoot {get;}
```

**this-Eigenschaft**

Ruft das DBParameter-Objekt mit dem angegebenen Namen in der Sammlung ab oder legt es fest.

**Überladungsliste**

Name	Beschreibung
<a href="#">this[Int]-Eigenschaft</a>	Ruft das DBParameter-Objekt am angegebenen Index in der Sammlung ab oder legt es fest.
<a href="#">this[String]-Eigenschaft</a>	Ruft das DBParameter-Objekt mit dem angegebenen Namen in der Sammlung ab oder legt es fest.

**this[Int]-Eigenschaft**

Ruft das DBParameter-Objekt am angegebenen Index in der Sammlung ab oder legt es fest.

**Visual Basic-Syntax**

```
Public Property Item(ByVal index As Integer) As Object
```

**C#-Syntax**

```
public object this[int index] {get;set;}
```

**Parameter**

- **index** Der Index des DBParameter-Objekts, das abgerufen oder festgelegt werden soll.

**Rückgabe**

Das DBParameter-Objekt mit dem angegebenen Index in der Sammlung.

**Siehe auch**

- [DBParameter-Klasse \[MobiLink-Server .NET\] auf Seite 571](#)

**this[String]-Eigenschaft**

Ruft das DBParameter-Objekt mit dem angegebenen Namen in der Sammlung ab oder legt es fest.

**Visual Basic-Syntax**

```
Public Property Item(ByVal parameterName As String) As Object
```

**C#-Syntax**

```
public object this[string parameterName] {get;set;}
```

**Parameter**

- **parameterName** Der Name des DBParameter-Objekts, das abgerufen oder festgelegt werden soll.

**Rückgabe**

Das DBParameter-Objekt mit dem angegebenen Namen in der Sammlung.

**Siehe auch**

- [DBParameter-Klasse \[MobiLink-Server .NET\] auf Seite 571](#)

# **DBRowReader-Schnittstelle**

Repräsentiert eine Gruppe von Zeilen, die aus einer Datenbank gelesen werden.

**Visual Basic-Syntax**

```
Public Interface DBRowReader
```

**C#-Syntax**

```
public interface DBRowReader
```

**Mitglieder**

Alle Mitglieder der DBRowReader-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">Close-Methode</a>	Bereinigt alle Ressourcen, die von diesem MLDBRowReader-Objekt verwendet werden.
<a href="#">NextRow-Methode</a>	Fragt die nächste Zeile in der Ergebnismenge ab und gibt sie zurück.
<a href="#">ColumnNames-Eigenschaft</a>	Ruft die Namen aller Spalten in der Ergebnismenge ab.

Name	Beschreibung
<a href="#">ColumnTypes-Eigenschaft</a>	Ruft die Typen aller Spalten in der Ergebnismenge ab.

### Bemerkungen

Die Ausführung der ExecuteReader-Methode erstellt eine DBRowReader-Schnittstelle.

### Siehe auch

- [DBCommand.ExecuteReader-Methode \[MobiLink-Server .NET\] auf Seite 562](#)

### Beispiel

Beim folgenden C#-Code wird eine Funktion aufgerufen, wobei die Zeilen in der Ergebnismenge durch die angegebene DBRowReader-Schnittstelle repräsentiert werden:

```

DBCommand stmt = conn.CreateCommand();
stmt.CommandText = "select intCol, strCol from table1 ";
DBRowReader rs = stmt.ExecuteReader();
object[] values = rset.NextRow();

while( values != null ) {
    handleRow( (int)values[0], (String)values[1] );
    values = rset.NextRow();
}
rset.Close();
stmt.Close();

```

## Close-Methode

Bereinigt alle Ressourcen, die von diesem MLDBRowReader-Objekt verwendet werden.

### Visual Basic-Syntax

```
Public Sub Close()
```

### C#-Syntax

```
public void Close()
```

### Bemerkungen

Dieses MLDBRowReader-Objekt kann nicht erneut verwendet werden, nachdem diese Methode aufgerufen wurde.

## NextRow-Methode

Fragt die nächste Zeile in der Ergebnismenge ab und gibt sie zurück.

### Visual Basic-Syntax

```
Public Function NextRow() As Object()
```

### C#-Syntax

```
public object[] NextRow()
```

### Rückgabe

Die nächste Zeile mit Werten in der Ergebnismenge oder der Wert NULL, wenn in dieser Ergebnismenge keine weiteren Zeilen vorhanden sind.

### Siehe auch

- [SQLType-Enumeration \[MobiLink-Server .NET\] auf Seite 624](#)

## ColumnNames-Eigenschaft

Ruft die Namen aller Spalten in der Ergebnismenge ab.

### Visual Basic-Syntax

```
Public ReadOnly Property ColumnNames As String()
```

### C#-Syntax

```
public string[] ColumnNames {get;}
```

### Bemerkungen

Der Wert ist ein Array von Zeichenfolgen, die den Spaltennamen in der Ergebnismenge entsprechen.

## ColumnTypes-Eigenschaft

Ruft die Typen aller Spalten in der Ergebnismenge ab.

### Visual Basic-Syntax

```
Public ReadOnly Property ColumnTypes As SQLType()
```

### C#-Syntax

```
public SQLType[] ColumnTypes {get;}
```

### Bemerkungen

Der Wert ist ein Array von SQLType-Typen, die den Spaltentypen in der Ergebnismenge entsprechen.

## DownloadData-Schnittstelle

Kapselt alle Datendownload-Vorgänge für die direkte Zeilenbehandlung.

### Visual Basic-Syntax

```
Public Interface DownloadData
```

**C#-Syntax**

```
public interface DownloadData
```

**Mitglieder**

Alle Mitglieder der DownloadData-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">GetDownloadTableByName-Methode</a>	Ruft die benannte Downloadtabelle für diese Synchronisation ab.
<a href="#">GetDownloadTables-Methode</a>	Ruft ein Array aller Downloadtabellen in dieser Synchronisation ab.

**Bemerkungen**

Verwenden Sie die GetDownloadData-Methode, um eine DownloadData-Instanz zu erhalten. Die Methoden GetDownloadTables und GetDownloadTableByName geben DownloadTableData-Instanzen zurück.

Diese Download-Daten stehen über DBConnectionContext zur Verfügung. Sie können nicht dazu verwendet werden, vor dem Ereignis begin\_sync oder in einer reinen Upload-Synchronisation auf die Download-Daten zuzugreifen.

Weitere Hinweise zur direkten Zeilenbehandlung finden Sie unter [„handle\\_DownloadData \(Verbindungsereignis\)“](#) auf Seite 371 und [„Direkte Zeilenbehandlung“](#) auf Seite 627.

**Siehe auch**

- [DownloadTableData-Schnittstelle \[MobiLink-Server .NET\]](#) auf Seite 588
- [DBConnectionContext.GetDownloadData-Methode \[MobiLink-Server .NET\]](#) auf Seite 567

**GetDownloadTableByName-Methode**

Ruft die benannte Downloadtabelle für diese Synchronisation ab.

**Visual Basic-Syntax**

```
Public Function GetDownloadTableByName(  
    ByVal table_name As String  
) As DownloadTableData
```

**C#-Syntax**

```
public DownloadTableData GetDownloadTableByName(string table_name)
```

**Parameter**

- **table\_name** Der Name der Tabelle, deren Daten Sie herunterladen möchten

## Rückgabe

Die Download-Daten für den angegebenen Tabellennamen oder der Wert NULL, wenn keine Tabelle gefunden wurde.

## GetDownloadTables-Methode

Ruft ein Array aller Downloadtabellen in dieser Synchronisation ab.

### Visual Basic-Syntax

```
Public Function GetDownloadTables() As DownloadTableData()
```

### C#-Syntax

```
public DownloadTableData[] GetDownloadTables()
```

## Rückgabe

Ein Array mit den Daten der Downloadtabelle. Die Reihenfolge der Tabellen im Array ist die gleiche wie die Reihenfolge für den entfernten Upload.

## Bemerkungen

Die in dieser Tabelle durchgeführten Vorgänge werden an die entfernte Datenbank gesendet.

## Beispiel

Im nachfolgenden Beispiel wird mit der Methode GetDownloadTables ein Array von DownloadTableData-Objekten für die aktuelle Synchronisation abgerufen. Bei diesem Beispiel wird vorausgesetzt, dass eine DBConnectionContext-Instanz namens \_cc vorhanden ist.

```
// The method used for the handle_DownloadData event.
public void HandleDownload() {
    // Get the DownloadData for the current synchronization.
    DownloadData my_dd = _cc.GetDownloadData();

    // Get an array of tables to set download operations.
    DownloadTableData[] download_tables = my_dd.GetDownloadTables();

    // Get the first table in the DownloadTableData array.
    DownloadTableData my_download_table = download_tables[0];

    // ...
}
```

## DownloadTableData-Schnittstelle

Kapselt Informationen für eine Downloadtabelle für eine Synchronisation.

### Visual Basic-Syntax

```
Public Interface DownloadTableData
```

**C#-Syntax**

```
public interface DownloadTableData
```

**Mitglieder**

Alle Mitglieder der DownloadTableData-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">GetDeleteCommand-Methode</a>	Ruft einen Befehl ab, mit dem der Benutzer den Download-Datenvorgängen Löschvorgänge hinzufügen kann.
<a href="#">GetLastDownloadTime-Methode</a>	Gibt die letzte Downloadzeit für diese Tabelle zurück.
<a href="#">GetName-Methode</a>	Ruft den Tabellennamen dieser Instanz ab.
<a href="#">GetSchemaTable-Methode</a>	Ruft ein DataTable-Objekt ab, das die Metadaten für diese Downloadtabelle beschreibt.
<a href="#">GetUpsertCommand-Methode</a>	Ruft einen Befehl ab, mit dem der Benutzer den Download-Datenvorgängen Einfüge- bzw. Aktualisierungsvorgänge hinzufügen kann.

**Bemerkungen**

Richten Sie mit dieser Schnittstelle die Datenvorgänge ein, die auf eine Synchronisationsclient-Site geladen werden.

**Beispiel**

Angenommen, Sie haben die folgende Tabelle:

```
CREATE TABLE remoteOrders (
    pk INT NOT NULL,
    coll VARCHAR(200),
    PRIMARY KEY (pk)
);
```

Im folgenden Beispiel wird mit der Methode GetDownloadTableByName eine DownloadTableData-Instanz zurückgegeben, die die Tabelle remoteOrders darstellt:

```
// The method used for the handle_DownloadData event
public void HandleDownload() {
    // _cc is a DBConnectionContext instance.

    // Get the DownloadData for the current synchronization.
    DownloadData my_dd = _cc.GetDownloadData();

    // Get the DownloadTableData for the remoteOrders table.
    DownloadTableData td = my_dd.GetDownloadTableByName("remoteOrders");

    // User defined-methods to set download operations.
    SetDownloadUpserts(td);
    SetDownloadDeletes(td);
}
```

```

    } // ...
}

```

In diesen Beispiel verwendet die Methode `SetDownloadInserts` die Methode `GetUpsertCommand`, um einen Befehl für die Zeilen abzurufen, die Sie einfügen oder aktualisieren möchten. `IDbCommand` enthält die Parameter, die auf die Werte gesetzt werden, die in die entfernte Datenbank eingefügt werden sollen.

```

void SetDownloadInserts(DownloadTableData td) {
    IDbCommand upsert_cmd = td.GetUpsertCommand();
    IDataParameterCollection parameters = upsert_cmd.Parameters;

    // The following method calls are the same as the following SQL
    statement:
    // INSERT INTO remoteOrders(pk, coll) values(2300, "truck");
    ((IDataParameter) (parameters[0])).Value = (Int32) 2300;
    ((IDataParameter) (parameters[1])).Value = (String) "truck";

    if (upsert_cmd.ExecuteNonQuery() > 0) {
        // Insert was not filtered.
    }
    else {
        // Insert was filtered because it was uploaded
        // in the same synchronization.
    }
}

```

Die folgende Methode ruft mit der Methode `DownloadTableData.GetDeleteCommand` einen Befehl für Zeilen ab, die Sie löschen möchten.

```

void SetDownloadDeletes(DownloadTableData td) {
    IDbCommand delete_cmd = t2_download_dd.GetDeleteCommand();

    // The following method calls are the same as the following SQL
    statement:
    // DELETE FROM remoteOrders where pk = 2300;
    IDataParameterCollection parameters = delete_cmd.Parameters;
    ((IDataParameter) (parameters[0])).Value = (Int32) 2300;
    delete_cmd.ExecuteNonQuery();
}

```

## GetDeleteCommand-Methode

Ruft einen Befehl ab, mit dem der Benutzer den Download-Datenvorgängen Löschvorgänge hinzufügen kann.

### Visual Basic-Syntax

```
Public Function GetDeleteCommand() As IDbCommand
```

### C#-Syntax

```
public IDbCommand GetDeleteCommand()
```

### Rückgabe

Ein Befehl für Löschvorgänge im Download.

## Bemerkungen

Der zurückgegebene Befehl enthält eine Anzahl von Parametern, die mit der Anzahl der Primärschlüsselspalten in dieser Tabelle übereinstimmt. Legen Sie die Spaltenwerte für die Primärschlüsselspalten fest und schließen Sie die Anweisung, die mit der ExecuteNonQuery-Methode für den Löschvorgang ausgeführt wird, in den Download mit ein. Wenn Sie die ExecuteNonQuery-Methode für den Befehl ausführen, wird der Wert 0 zurückgegeben, wenn der Löschvorgang gefiltert wurde. Wenn die Einfügung nicht gefiltert wurde, wird der Wert 1 zurückgegeben.

Um eine Zeile zu löschen, müssen Sie alle Primärschlüsselwerte für Download-Löschvorgänge festlegen. Um die entfernte Tabelle zu kürzen, setzen Sie alle Primärschlüsselspalten auf NULL.

## Siehe auch

- [DownloadTableData-Schnittstelle \[MobiLink-Server .NET\] auf Seite 588](#)

## GetLastDownloadTime-Methode

Gibt die letzte Downloadzeit für diese Tabelle zurück.

### Visual Basic-Syntax

```
Public Function GetLastDownloadTime() As Date
```

### C#-Syntax

```
public DateTime GetLastDownloadTime()
```

### Rückgabe

Die letzte Downloadzeit für diese Tabelle.

## Bemerkungen

Dies ist dieselbe letzte Downloadzeit, die an mehrere der tabellenweisen Download-Ereignisse übergeben wurde.

Die letzte Downloadzeit dient zur Generierung der Download-Daten der Tabelle für eine bestimmte Synchronisation.

## GetName-Methode

Ruft den Tabellennamen dieser Instanz ab.

### Visual Basic-Syntax

```
Public Function GetName() As String
```

### C#-Syntax

```
public string GetName()
```

## Rückgabe

Der Tabellename dieser Instanz.

## Bemerkungen

Dies ist eine Dienstprogrammfunktion. Auf den Tabellennamen kann auch über das Schema für diese Instanz zugegriffen werden.

## GetSchemaTable-Methode

Ruft ein DataTable-Objekt ab, das die Metadaten für diese Downloadtabelle beschreibt.

### Visual Basic-Syntax

```
Public Function GetSchemaTable() As DataTable
```

### C#-Syntax

```
public DataTable GetSchemaTable()
```

## Rückgabe

Ein DataTable-Objekt, das die Spalten-Metadaten beschreibt.

## Bemerkungen

Sie müssen die Clientoption zum Senden von Spaltennamen angeben, wenn Sie möchten, dass das DataTable-Objekt Spaltennamen-Informationen enthält. SendColumnNames ist standardmäßig angegeben.

## GetUpsertCommand-Methode

Ruft einen Befehl ab, mit dem der Benutzer den Download-Datenvorgängen Einfüge- bzw. Aktualisierungsvorgänge hinzufügen kann.

### Visual Basic-Syntax

```
Public Function GetUpsertCommand() As IDbCommand
```

### C#-Syntax

```
public IDbCommand GetUpsertCommand()
```

## Rückgabe

Ein Befehl für Einfügungen/Aktualisierungen für den Download.

## Bemerkungen

Der zurückgegebene Befehl enthält eine Anzahl von Parametern, die mit der Anzahl der Spalten in dieser Tabelle übereinstimmt. Legen Sie die Spaltenwerte für die Einfügung fest und schließen Sie die Anweisung, die mit der ExecuteNonQuery-Methode für die Einfügung bzw. Aktualisierung ausgeführt wird, in den Download mit ein. Wenn Sie die ExecuteNonQuery-Methode für den Befehl ausführen, wird

der Wert 0 zurückgegeben, wenn der Einfügevorgang gefiltert wurde. Wenn die Einfügung nicht gefiltert wurde, wird der Wert 1 zurückgegeben.

Sie können diesem Befehl keine Parameter hinzufügen und keine daraus entfernen. Sie können lediglich die Werte festlegen.

#### Siehe auch

- [DownloadTableData-Schnittstelle \[MobiLink-Server .NET\] auf Seite 588](#)

## FatalException-Klasse

Signalisiert MobiLink, dass intern ein schwerwiegender serverseitiger Fehler aufgetreten ist und die Anwendung sofort geschlossen werden sollte.

#### Visual Basic-Syntax

```
Public Class FatalException Inherits System.ApplicationException
```

#### C#-Syntax

```
public class FatalException : System.ApplicationException
```

#### Basisklassen

- [System.ApplicationException](#)

#### Mitglieder

Alle Mitglieder der FatalException-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">FatalException-Konstruktor</a>	Erstellt ein neues FatalException-Objekt mit der Standardmeldung.

## FatalException-Konstruktor

Erstellt ein neues FatalException-Objekt mit der Standardmeldung.

#### Überladungsliste

Name	Beschreibung
<a href="#">FatalException()-Konstruktor</a>	Erstellt ein neues FatalException-Objekt mit der Standardmeldung.
<a href="#">FatalException(string)-Konstruktor</a>	Erstellt ein neues FatalException-Objekt mit der angegebenen Meldung.

Name	Beschreibung
<a href="#">FatalException(string, Exception)-Konstruktor</a>	Erstellt ein neues FatalException-Objekt mit der angegebenen Meldung sowie der angegebenen inneren Ausnahmebedingung, durch die diese Ausnahmebedingung verursacht wurde.

## FatalException()-Konstruktor

Erstellt ein neues FatalException-Objekt mit der Standardmeldung.

### Visual Basic-Syntax

```
Public Sub New()
```

### C#-Syntax

```
public FatalException()
```

## FatalException(string)-Konstruktor

Erstellt ein neues FatalException-Objekt mit der angegebenen Meldung.

### Visual Basic-Syntax

```
Public Sub New(ByVal message As String)
```

### C#-Syntax

```
public FatalException(string message)
```

### Parameter

- **message** Die Meldung für dieses FatalException-Objekt.

## FatalException(string, Exception)-Konstruktor

Erstellt ein neues FatalException-Objekt mit der angegebenen Meldung sowie der angegebenen inneren Ausnahmebedingung, durch die diese Ausnahmebedingung verursacht wurde.

### Visual Basic-Syntax

```
Public Sub New(ByVal message As String, ByVal ie As Exception)
```

### C#-Syntax

```
public FatalException(string message, Exception ie)
```

### Parameter

- **message** Die Meldung für dieses FatalException-Objekt.
- **ie** Die Ausnahmebedingung, aufgrund derer dieses FatalException-Objekt erstellt wurde.

## LogMessage-Klasse

Enthält Informationen über eine Meldung, die in das Log geschrieben wurde.

### Visual Basic-Syntax

```
Public Class LogMessage
```

### C#-Syntax

```
public class LogMessage
```

### Mitglieder

Alle Mitglieder der LogMessage-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">LogMessage-Konstruktor</a>	Erstellt ein LogMessage-Objekt mit den angegebenen Attributen.
<a href="#">MessageType-Enumeration</a>	Enumeriert alle möglichen LogMessage-Typen.
<a href="#">Text-Eigenschaft</a>	Der Haupttext der Meldung.
<a href="#">Type-Eigenschaft</a>	Der Typ der Logmeldung, die diese Instanz repräsentiert.
<a href="#">User-Eigenschaft</a>	Der Benutzer, für den diese Meldung protokolliert wird.

### Bemerkungen

Eine Instanz dieser Klasse wird an ein LogCallback übergeben.

### Siehe auch

- [LogCallback-Delegat \[MobiLink-Server .NET\] auf Seite 623](#)

## LogMessage-Konstruktor

Erstellt ein LogMessage-Objekt mit den angegebenen Attributen.

### Visual Basic-Syntax

```
Public Sub New(  
    ByVal type As MessageType,  
    ByVal user As String,  
    ByVal text As String  
)
```

### C#-Syntax

```
public LogMessage(MessageType type, string user, string text)
```

## MessageType-Enumeration

Enumeriert alle möglichen LogMessage-Typen.

### Visual Basic-Syntax

```
Public Enum MessageType
```

### C#-Syntax

```
public enum MessageType
```

### Mitglieder

Mitgliedsname	Beschreibung
ERROR	Die Logmeldung ist ein Fehler.
WARNING	Die Logmeldung ist eine Warnung.
INFO	Die Logmeldung ist eine Information.

## Text-Eigenschaft

Der Haupttext der Meldung.

### Visual Basic-Syntax

```
Public ReadOnly Property Text As String
```

### C#-Syntax

```
public string Text {get;}
```

## Type-Eigenschaft

Der Typ der Logmeldung, die diese Instanz repräsentiert.

### Visual Basic-Syntax

```
Public ReadOnly Property Type As MessageType
```

### C#-Syntax

```
public MessageType Type {get;}
```

## User-Eigenschaft

Der Benutzer, für den diese Meldung protokolliert wird.

### Visual Basic-Syntax

```
Public ReadOnly Property User As String
```

**C#-Syntax**

```
public string User {get;}
```

**Bemerkungen**

Diese Eigenschaft kann NULL sein.

## NetworkData-Schnittstelle

Enthält Informationen über die Netzwerkdatenströme für eine Synchronisation.

**Visual Basic-Syntax**

```
Public Interface NetworkData
```

**C#-Syntax**

```
public interface NetworkData
```

**Mitglieder**

Alle Mitglieder der NetworkData-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">GetHTTPHeaderValue-Methode</a>	Gibt den Wert des letzten vom Server empfangenen Headers mit dem angegebenen Namen zurück.
<a href="#">GetHTTPHeaderValues-Methode</a>	Gibt alle vom Server empfangenen Header-Werte zurück, die mit dem angegebenen Namen verknüpft sind.
<a href="#">ClientCertificates-Eigenschaft</a>	Gibt ein X509Certificate2Collection-Objekt zurück, das alle vom Client gesendeten Zertifikate enthält.
<a href="#">HTTPHeaderHeaders-Eigenschaft</a>	Gibt ein Wörterbuch zurück, in dem Header-Namen einer Liste von Header-Werten zugeordnet sind.
<a href="#">IsEndToEndEncrypted-Eigenschaft</a>	Ermittelt, ob diese Synchronisation eine Punkt-zu-Punkt-Verschlüsselung hat.
<a href="#">IsHTTP-Eigenschaft</a>	Ermittelt, ob die Synchronisation HTTP oder HTTPS verwendet.
<a href="#">IsTLS-Eigenschaft</a>	Ermittelt, ob diese Synchronisation TLS verwendet.

**Bemerkungen**

Diese Schnittstelle ist nützlich, wenn die Benutzerauthentifizierung bei einem anderen Server im Unternehmen erfolgt, der das clientseitige Zertifikat und HTTP-Header verwendet.

Zum Aktivieren einer Sammlung von Netzwerk-Datenströmen fügen Sie den -x Parametern `collect_network_data = 1` hinzu. Diese Option fügt für die Synchronisation zusätzlichen Speicher-

Overhead zum Speichern der Daten hinzu. Wenn Sie TLS oder HTTPS mit clientseitigen Zertifikaten verwenden, fügen Sie `trusted_certificates=<Zertifikatdatei>` hinzu, damit der Server den Client während des TLS-Handshakes auffordert, ein Zertifikat zu senden. Dies ist jedoch mit Zeitaufwand und Netzwerkkosten verbunden.

Sie erhalten ein `NetworkData`-Objekt durch Aufruf der `NetworkData`-Methode der `DBConnectionContext`-Schnittstelle. Wenn Sie HTTP oder HTTPS verwenden, enthält es die Header-Daten für die letzte HTTP-Anforderung, die vom Server vor dem Aufruf der Authentifizierungsskripten empfangen wurden.

## Beispiel

Das folgende Beispiel zeigt, wie ein `NetworkData`-Objekt vom `DBConnectionContext`-Objekt abgerufen und die Daten ausgegeben werden.

```
using iAnywhere.MobiLink.Script;
using System.Collections.Generic;
using System.Security.Cryptography.X509Certificates;

public class OrderProcessor {
    DBConnectionContext _cc;

    public OrderProcessor( DBConnectionContext cc ) {
        _cc = cc;
    }

    public void AuthUser() {
        NetworkData nd = _cc.NetworkData;
        if( nd != null ) {
            if( nd.IsHTTP ) {
                PrintLn( "http" );
                string user_agent = nd.GetHTTPHeaderValue( "user-agent" );
                PrintLn( "  user-agent: " + user_agent.Substring( 0,
user_agent.IndexOf( '/' ) ) );
            } else {
                PrintLn( "no http" );
            }
            if( nd.IsTLS ) {
                PrintLn( "tls" );
                X509Certificate2Collection certs = nd.ClientCertificates;
                if( certs != null ) {
                    PrintLn( "  client-side cert:" );
                    int n = 1;
                    foreach( X509Certificate2 x509 in certs ) {
                        PrintLn( "    cert " + n++ );
                        PrintLn( "      Subject: " + x509.SubjectName.Name );
                        PrintLn( "      Issuer: " + x509.IssuerName.Name );
                    }
                } else {
                    PrintLn( "  no client cert" );
                }
            } else {
                PrintLn( "no tls" );
            }
            if( nd.IsEndToEndEncrypted ) {
                PrintLn( "e2ee" );
            } else {
                PrintLn( "no e2ee" );
            }
        } else {
```

```

        }
    }
    PrintLn( "NULL networkdata" );
}

```

## GetHTTPHeaderValue-Methode

Gibt den Wert des letzten vom Server empfangenen Headers mit dem angegebenen Namen zurück.

### Visual Basic-Syntax

```
Public Function GetHTTPHeaderValue(ByVal name As String) As String
```

### C#-Syntax

```
public string GetHTTPHeaderValue(string name)
```

### Parameter

- **name** Der Headername, für den der Wert zurückgegeben werden soll.

### Rückgabe

Der letzte Header-Wert, der dem angegebenen Header-Namen zugeordnet ist.

### Siehe auch

- [NetworkData.GetHTTPHeaderValues-Methode \[MobiLink-Server-API für .NET\] auf Seite 599](#)
- [NetworkData.HTTPHeaders-Eigenschaft \[MobiLink-Server-API für .NET\] auf Seite 600](#)

## GetHTTPHeaderValues-Methode

Gibt alle vom Server empfangenen Header-Werte zurück, die mit dem angegebenen Namen verknüpft sind.

### Visual Basic-Syntax

```
Public Function GetHTTPHeaderValues(
    ByVal name As String
) As IList< string >
```

### C#-Syntax

```
public IList< string > GetHTTPHeaderValues(string name)
```

### Parameter

- **name** Der Headername, für den die Werte zurückgegeben werden sollen.

### Rückgabe

Die Header-Werte, die dem angegebenen Header-Namen zugeordnet sind.

#### Siehe auch

- [NetworkData.GetHTTPHeaderValue-Methode \[MobiLink-Server-API für .NET\] auf Seite 599](#)
- [NetworkData.HTTPHeaders-Eigenschaft \[MobiLink-Server-API für .NET\] auf Seite 600](#)

## ClientCertificates-Eigenschaft

Gibt ein X509Certificate2Collection-Objekt zurück, das alle vom Client gesendeten Zertifikate enthält.

#### Visual Basic-Syntax

```
Public ReadOnly Property ClientCertificates As  
X509Certificate2Collection
```

#### C#-Syntax

```
public X509Certificate2Collection ClientCertificates {get;}
```

#### Rückgabe

Ein X509Certificate2Collection-Objekt, das die X.509-Zertifikate enthält, die den Client identifizieren, bzw. NULL, wenn keine solchen Zertifikate angegeben wurden.

#### Bemerkungen

Diese Funktion gibt nur dann einen anderen Wert als NULL zurück, wenn der isTLS()-Parameter TRUE ist, der Client mit dem identity-Datenstromparameter ein Zertifikat liefert und die trusted\_certificates-Option auf dem Server gesetzt ist. Ein CertPath-Wert, der nicht NULL ist, enthält die Zertifikate, sortiert vom selbstsignierten Zertifikat zum Peer-Zertifikat.

#### Siehe auch

- [trusted\\_certificates-Beispiel für .NET auf Seite 102](#)

## HTTPHeaders-Eigenschaft

Gibt ein Wörterbuch zurück, in dem Header-Namen einer Liste von Header-Werten zugeordnet sind.

#### Visual Basic-Syntax

```
Public ReadOnly Property HTTPHeaders As IDictionary< string, IList<  
string > >
```

#### C#-Syntax

```
public IDictionary< string, IList< string > > HTTPHeaders {get;}
```

#### Rückgabe

Ein Wörterbuch der Paare aus Headernamen und Werten.

#### Siehe auch

- [NetworkData.GetHTTPHeaderValue-Methode \[MobiLink-Server-API für .NET\] auf Seite 599](#)
- [NetworkData.GetHTTPHeaderValues-Methode \[MobiLink-Server-API für .NET\] auf Seite 599](#)

## IsEndToEndEncrypted-Eigenschaft

Ermittelt, ob diese Synchronisation eine Punkt-zu-Punkt-Verschlüsselung hat.

### Visual Basic-Syntax

```
Public ReadOnly Property IsEndToEndEncrypted As Boolean
```

### C#-Syntax

```
public bool IsEndToEndEncrypted {get;}
```

## IsHTTP-Eigenschaft

Ermittelt, ob die Synchronisation HTTP oder HTTPS verwendet.

### Visual Basic-Syntax

```
Public ReadOnly Property IsHTTP As Boolean
```

### C#-Syntax

```
public bool IsHTTP {get;}
```

### Rückgabe

TRUE, wenn diese Synchronisation HTTP oder HTTPS verwendet, sonst FALSE.

## IsTLS-Eigenschaft

Ermittelt, ob diese Synchronisation TLS verwendet.

### Visual Basic-Syntax

```
Public ReadOnly Property IsTLS As Boolean
```

### C#-Syntax

```
public bool IsTLS {get;}
```

### Rückgabe

TRUE, wenn diese Synchronisation TLS verwendet, sonst FALSE.

## ScriptExecutionException-Klasse

Signalisiert, dass in einem Benutzerskript ein Fehler aufgetreten ist.

### Visual Basic-Syntax

```
Public Class ScriptExecutionException  
    Inherits System.ApplicationException
```

**C#-Syntax**

```
public class ScriptExecutionException : System.ApplicationException
```

**Basisklassen**

- [System.ApplicationException](#)

**Abgeleitete Klassen**

- [ServerException-Klasse \[MobiLink-Server .NET\] auf Seite 608](#)
- [SynchronizationException-Klasse \[MobiLink-Server .NET\] auf Seite 612](#)

**Mitglieder**

Alle Mitglieder der ScriptExecutionException-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">ScriptExecutionException-Konstruktor</a>	Erstellt ein neues ScriptExecutionException-Objekt mit der Standardmeldung.

**Bemerkungen**

Das Auftreten dieser Ausnahmebedingung oder einer beliebigen daraus abgeleiteten Ausnahmebedingung, außer SynchronizationException, veranlasst den MobiLink-Server herunterzufahren.

**ScriptExecutionException-Konstruktor**

Erstellt ein neues ScriptExecutionException-Objekt mit der Standardmeldung.

**Überladungsliste**

Name	Beschreibung
<a href="#">ScriptExecutionException()-Konstruktor</a>	Erstellt ein neues ScriptExecutionException-Objekt mit der Standardmeldung.
<a href="#">ScriptExecutionException(string)-Konstruktor</a>	Erstellt ein neues ScriptExecutionException-Objekt mit der angegebenen Meldung.
<a href="#">ScriptExecutionException(string, Exception)-Konstruktor</a>	Erstellt ein neues ScriptExecutionException-Objekt mit der angegebenen Meldung sowie der angegebenen inneren Ausnahmebedingung, durch die diese Ausnahmebedingung verursacht wurde.

**ScriptExecutionException()-Konstruktor**

Erstellt ein neues ScriptExecutionException-Objekt mit der Standardmeldung.

**Visual Basic-Syntax**

```
Public Sub New()
```

**C#-Syntax**

```
public ScriptExecutionException()
```

**ScriptExecutionException(string)-Konstruktor**

Erstellt ein neues ScriptExecutionException-Objekt mit der angegebenen Meldung.

**Visual Basic-Syntax**

```
Public Sub New(ByVal message As String)
```

**C#-Syntax**

```
public ScriptExecutionException(string message)
```

**Parameter**

- **message** Die Meldung für dieses ScriptExecutionException-Objekt.

**ScriptExecutionException(string, Exception)-Konstruktor**

Erstellt ein neues ScriptExecutionException-Objekt mit der angegebenen Meldung sowie der angegebenen inneren Ausnahmebedingung, durch die diese Ausnahmebedingung verursacht wurde.

**Visual Basic-Syntax**

```
Public Sub New(ByVal message As String, ByVal ie As Exception)
```

**C#-Syntax**

```
public ScriptExecutionException(string message, Exception ie)
```

**Parameter**

- **message** Die Meldung für dieses ScriptExecutionException-Objekt.
- **ie** Die Ausnahmebedingung, aufgrund derer dieses ScriptExecutionException-Objekt erstellt wurde.

## ServerContext-Schnittstelle

Instanziert den Kontext, der für die Dauer des MobiLink-Servers vorhanden ist.

**Visual Basic-Syntax**

```
Public Interface ServerContext
```

**C#-Syntax**

```
public interface ServerContext
```

## Mitglieder

Alle Mitglieder der ServerContext-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">getProperties-Methode</a>	Gibt eine Sammlung der Eigenschaften für die angegebene Komponente und Eigenschaftengruppe zurück.
<a href="#">getPropertiesByVersion-Methode</a>	Gibt eine Sammlung der Eigenschaften für die angegebene Skriptversion zurück.
<a href="#">getPropertySetNames-Methode</a>	Gibt eine Sammlung der Eigenschaftengruppennamen für die angegebene Komponente zurück.
<a href="#">GetStartClassInstances-Methode</a>	Gibt alle Startklassen zurück, die in diese Domäne geladen wurden.
<a href="#">makeConnection-Methode</a>	Erstellt eine neue Datenbankverbindung.
<a href="#">Shutdown-Methode</a>	Bewirkt, dass der MobiLink-Server ein sanftes Herunterfahren ausführt.
<a href="#">ErrorListener-Ereignis</a>	Wird ausgelöst, wenn der MobiLink-Server einen Fehler ausgibt.
<a href="#">InfoListener-Ereignis</a>	Wird ausgelöst, wenn der MobiLink-Server Informationen ausgibt.
<a href="#">ShutdownListener-Ereignis</a>	Wird ausgelöst, wenn der MobiLink-Server heruntergefahren wird.
<a href="#">WarningListener-Ereignis</a>	Wird ausgelöst, wenn der MobiLink-Server eine Warnung ausgibt.

## Bemerkungen

Dieser Kontext kann wie statische Daten gespeichert und in einem Hintergrund-Thread benutzt werden. Er ist für die Dauer der von MobiLink aufgerufenen .NET CLR gültig.

Verwenden Sie die GetServerContext-Methode für den Zugriff auf eine ServerContext-Instanz. Sie wird an den Konstruktor von Boot-Klassen übergeben.

## Siehe auch

- [DBConnectionContext.GetServerContext-Methode \[MobiLink-Server .NET\] auf Seite 569](#)

## getProperties-Methode

Gibt eine Sammlung der Eigenschaften für die angegebene Komponente und Eigenschaftengruppe zurück.

**Visual Basic-Syntax**

```
Public Function getProperties(  
    ByVal component As String,  
    ByVal set As String  
) As NameValueCollection
```

**C#-Syntax**

```
public NameValueCollection getProperties(string component, string set)
```

**Parameter**

- **component** Referenziert die Spalte *component\_name* der Tabelle *ml\_property*.
- **set** Referenziert die Spalte *property\_set\_name* der Tabelle *ml\_property*.

**Rückgabe**

Die Eigenschaften für die angegebene Komponente bzw. Eigenschaftengruppe.

**Bemerkungen**

Die Eigenschaften werden in der Tabelle *ml\_property* gespeichert. Weitere Hinweise finden Sie unter [„ml\\_add\\_property-Systemprozedur“ auf Seite 664](#).

## getPropertiesByVersion-Methode

Gibt eine Sammlung der Eigenschaften für die angegebene Skriptversion zurück.

**Visual Basic-Syntax**

```
Public Function getPropertiesByVersion(  
    ByVal script_version As String  
) As NameValueCollection
```

**C#-Syntax**

```
public NameValueCollection getPropertiesByVersion(string script_version)
```

**Parameter**

- **script\_version** Die Skriptversion, für die zugehörige Eigenschaften zurückgegeben werden sollen.

**Rückgabe**

Die Eigenschaften für die angegebene Skriptversion.

**Bemerkungen**

Die Eigenschaften werden in der Tabelle *ml\_property* gespeichert. Weitere Hinweise finden Sie unter [„ml\\_add\\_property-Systemprozedur“ auf Seite 664](#).

## getPropertySetNames-Methode

Gibt eine Sammlung der Eigenschaftengruppennamen für die angegebene Komponente zurück.

### Visual Basic-Syntax

```
Public Function getPropertySetNames(  
    ByVal component As String  
) As StringCollection
```

### C#-Syntax

```
public StringCollection getPropertySetNames(string component)
```

### Parameter

- **component** Referenziert die Spalte component\_name der Tabelle ml\_property.

### Rückgabe

Die Sammlung der Eigenschaftengruppenamen für eine bestimmte Komponente.

### Bemerkungen

Die Eigenschaften werden in der Tabelle ml\_property gespeichert. Weitere Hinweise finden Sie unter [„ml\\_add\\_property-Systemprozedur“ auf Seite 664](#).

## GetStartClassInstances-Methode

Gibt alle Startklassen zurück, die in diese Domäne geladen wurden.

### Visual Basic-Syntax

```
Public Function GetStartClassInstances() As Object()
```

### C#-Syntax

```
public object[] GetStartClassInstances()
```

### Rückgabe

Ruft ein Array aller Startklassen ab, die beim Serverstart erstellt wurden. Die Array-Länge beträgt 0, wenn keine Startklassen vorhanden sind.

### Bemerkungen

Weitere Hinweise zu benutzerdefinierten Startklassen finden Sie unter [„Benutzerdefinierte Startklassen“ auf Seite 542](#).

### Beispiel

Das folgende Beispiel zeigt, wie Sie eine Startklasse finden:

```
void FindStartClass( ServerContext sc, string name )  
{  
    object[] startClasses = sc.GetStartClassInstances();  
  
    foreach( object obj in startClasses ) {  
        if( obj is MyClass ) {  
            // Execute some code.....  
        }  
    }  
}
```

```
}  
}
```

## makeConnection-Methode

Erstellt eine neue Datenbankverbindung.

### Visual Basic-Syntax

```
Public Function MakeConnection() As DBConnection
```

### C#-Syntax

```
public DBConnection MakeConnection()
```

### Rückgabe

Eine neue Verbindung.

## Shutdown-Methode

Bewirkt, dass der MobiLink-Server ein sanftes Herunterfahren ausführt.

### Visual Basic-Syntax

```
Public Sub Shutdown()
```

### C#-Syntax

```
public void Shutdown()
```

## ErrorListener-Ereignis

Wird ausgelöst, wenn der MobiLink-Server einen Fehler ausgibt.

### Visual Basic-Syntax

```
Public Event ErrorListener As LogCallback
```

### C#-Syntax

```
public event LogCallback ErrorListener;
```

## InfoListener-Ereignis

Wird ausgelöst, wenn der MobiLink-Server Informationen ausgibt.

### Visual Basic-Syntax

```
Public Event InfoListener As LogCallback
```

**C#-Syntax**

```
public event LogCallback InfoListener;
```

**ShutdownListener-Ereignis**

Wird ausgelöst, wenn der MobiLink-Server heruntergefahren wird.

**Visual Basic-Syntax**

```
Public Event ShutdownListener As ShutdownCallback
```

**C#-Syntax**

```
public event ShutdownCallback ShutdownListener;
```

**WarningListener-Ereignis**

Wird ausgelöst, wenn der MobiLink-Server eine Warnung ausgibt.

**Visual Basic-Syntax**

```
Public Event WarningListener As LogCallback
```

**C#-Syntax**

```
public event LogCallback WarningListener;
```

**ServerException-Klasse**

Sendet ein Signal an MobiLink, wenn auf dem Server ein Fehler aufgetreten ist, um anzuzeigen, dass er sofort heruntergefahren werden muss.

**Visual Basic-Syntax**

```
Public Class ServerException Inherits ScriptExecutionException
```

**C#-Syntax**

```
public class ServerException : ScriptExecutionException
```

**Basisklassen**

- [ScriptExecutionException-Klasse \[MobiLink-Server .NET\] auf Seite 601](#)

**Mitglieder**

Alle Mitglieder der ServerException-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">ScriptExecutionException-Konstruktor</a>	Erstellt ein neues ScriptExecutionException-Objekt mit der Standardmeldung.

Name	Beschreibung
<a href="#">ServerException-Konstruktor</a>	Erstellt ein neues ServerException-Objekt.

## ServerException-Konstruktor

Erstellt ein neues ServerException-Objekt.

### Überladungsliste

Name	Beschreibung
<a href="#">ServerException()-Konstruktor</a>	Erstellt ein neues ServerException-Objekt mit der Standardmeldung.
<a href="#">ServerException(string)-Konstruktor</a>	Erstellt ein neues ServerException-Objekt mit der angegebenen Meldung.
<a href="#">ServerException(string, Exception)-Konstruktor</a>	Erstellt ein neues ServerException-Objekt mit der angegebenen Meldung sowie der angegebenen inneren Ausnahmebedingung, durch die diese Ausnahmebedingung verursacht wurde.

### ServerException()-Konstruktor

Erstellt ein neues ServerException-Objekt mit der Standardmeldung.

#### Visual Basic-Syntax

```
Public Sub New()
```

#### C#-Syntax

```
public ServerException()
```

### ServerException(string)-Konstruktor

Erstellt ein neues ServerException-Objekt mit der angegebenen Meldung.

#### Visual Basic-Syntax

```
Public Sub New(ByVal message As String)
```

#### C#-Syntax

```
public ServerException(string message)
```

#### Parameter

- **message** Die Meldung für diese ServerException.

## ServerException(string, Exception)-Konstruktor

Erstellt ein neues ServerException-Objekt mit der angegebenen Meldung sowie der angegebenen inneren Ausnahmebedingung, durch die diese Ausnahmebedingung verursacht wurde.

### Visual Basic-Syntax

```
Public Sub New(ByVal message As String, ByVal ie As Exception)
```

### C#-Syntax

```
public ServerException(string message, Exception ie)
```

### Parameter

- **message** Die Meldung für diese ServerException.
- **ie** Die Ausnahme, die diese ServerException verursacht hat.

## SpatialUtilities-Klasse

Repräsentiert eine Sammlung von statischen Methoden für die Arbeit mit räumlichen Werten.

### Visual Basic-Syntax

```
Public NotInheritable Class SpatialUtilities
```

### C#-Syntax

```
public sealed class SpatialUtilities
```

### Mitglieder

Alle Mitglieder der SpatialUtilities-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">CreateSpatialValue-Methode</a>	Gibt ein neues Byte-Array zurück, das einen für den Download formatierten räumlichen Wert enthält: Die ersten vier Byte enthalten die angegebene SRID in Little-Endian und der Rest besteht aus den räumlichen Daten, die in dem angegebenen Byte-Array übergeben werden.
<a href="#">GetBytes-Methode</a>	Gibt ein neues Byte-Array mit denselben räumlichen Daten wie das angegebene Byte-Array zurück, wobei jedoch die SRID entfernt wurde.
<a href="#">GetSRID-Methode</a>	Gibt die SRID für den angegebenen räumlichen Wert zurück.
<a href="#">SetSRID-Methode</a>	Speichert die angegebene SRID in den ersten vier Byte des angegebenen Byte-Arrays.

## CreateSpatialValue-Methode

Gibt ein neues Byte-Array zurück, das einen für den Download formatierten räumlichen Wert enthält: Die ersten vier Byte enthalten die angegebene SRID in Little-Endian und der Rest besteht aus den räumlichen Daten, die in dem angegebenen Byte-Array übergeben werden.

### Visual Basic-Syntax

```
Public Shared Function CreateSpatialValue(  
    ByVal srid As Integer,  
    ByVal spatial_value As Byte()  
) As Byte()
```

### C#-Syntax

```
public static byte[] CreateSpatialValue(int srid, byte[] spatial_value)
```

### Parameter

- **srid** Die SRID.
- **spatial\_value** Die räumlichen Daten.

### Rückgabe

Der für den Download formatierte räumliche Wert.

## GetBytes-Methode

Gibt ein neues Byte-Array mit denselben räumlichen Daten wie das angegebene Byte-Array zurück, wobei jedoch die SRID entfernt wurde.

### Visual Basic-Syntax

```
Public Shared Function GetBytes(ByVal spatial_value As Byte()) As Byte()
```

### C#-Syntax

```
public static byte[] GetBytes(byte[] spatial_value)
```

### Parameter

- **spatial\_value** Ein räumlicher Wert, dessen SRID entfernt werden muss

### Rückgabe

Das neue Byte-Array.

## GetSRID-Methode

Gibt die SRID für den angegebenen räumlichen Wert zurück.

### Visual Basic-Syntax

```
Public Shared Function GetSRID(ByVal spatial_value As Byte()) As Integer
```

## C#-Syntax

```
public static int GetSRID(byte[] spatial_value)
```

### Parameter

- **spatial\_value** Der hochgeladene Wert. Die ersten vier Byte müssen die angegebene SRID in Little-Endian enthalten.

### Rückgabe

Die SRID.

## SetSRID-Methode

Speichert die angegebene SRID in den ersten vier Byte des angegebenen Byte-Arrays.

### Visual Basic-Syntax

```
Public Shared Sub SetSRID(  
    ByVal spatial_value As Byte(),  
    ByVal srid As Integer  
)
```

## C#-Syntax

```
public static void SetSRID(byte[] spatial_value, int srid)
```

### Parameter

- **spatial\_value** Das Array, in dem die SRID gespeichert werden soll.
- **srid** Die zu speichernde SRID.

## SynchronizationException-Klasse

Zeigt an, dass eine Ausnahmebedingung eingetreten ist und die aktuelle Synchronisation zurückgesetzt und neu gestartet werden sollte.

### Visual Basic-Syntax

```
Public Class SynchronizationException Inherits ScriptExecutionException
```

## C#-Syntax

```
public class SynchronizationException : ScriptExecutionException
```

### Basisklassen

- [ScriptExecutionException-Klasse \[MobiLink-Server .NET\] auf Seite 601](#)

### Mitglieder

Alle Mitglieder der SynchronizationException-Klasse, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">ScriptExecutionException-Konstruktor</a>	Erstellt ein neues ScriptExecutionException-Objekt mit der Standardmeldung.
<a href="#">SynchronizationException-Konstruktor</a>	Erstellt ein neues SynchronizationException-Objekt.

## SynchronizationException-Konstruktor

Erstellt ein neues SynchronizationException-Objekt.

### Überladungsliste

Name	Beschreibung
<a href="#">SynchronizationException()-Konstruktor</a>	Erstellt ein neues SynchronizationException-Objekt mit der Standardmeldung.
<a href="#">SynchronizationException(string)-Konstruktor</a>	Erstellt ein neues SynchronizationException-Objekt mit der angegebenen Meldung.
<a href="#">SynchronizationException(string, Exception)-Konstruktor</a>	Erstellt ein neues SynchronizationException-Objekt mit der angegebenen Meldung sowie der angegebenen inneren Ausnahmebedingung, durch die diese Ausnahmebedingung verursacht wurde.

## SynchronizationException()-Konstruktor

Erstellt ein neues SynchronizationException-Objekt mit der Standardmeldung.

### Visual Basic-Syntax

```
Public Sub New()
```

### C#-Syntax

```
public SynchronizationException()
```

## SynchronizationException(string)-Konstruktor

Erstellt ein neues SynchronizationException-Objekt mit der angegebenen Meldung.

### Visual Basic-Syntax

```
Public Sub New(ByVal message As String)
```

### C#-Syntax

```
public SynchronizationException(string message)
```

**Parameter**

- **message** Die Meldung für dieses SynchronizationException-Objekt.

**SynchronizationException(string, Exception)-Konstruktor**

Erstellt ein neues SynchronizationException-Objekt mit der angegebenen Meldung sowie der angegebenen inneren Ausnahmebedingung, durch die diese Ausnahmebedingung verursacht wurde.

**Visual Basic-Syntax**

```
Public Sub New(ByVal message As String, ByVal ie As Exception)
```

**C#-Syntax**

```
public SynchronizationException(string message, Exception ie)
```

**Parameter**

- **message** Die Meldung für dieses SynchronizationException-Objekt.
- **ie** Die Ausnahmebedingung, aufgrund derer dieses SynchronizationException-Objekt erstellt wurde.

**UpdateDataReader-Schnittstelle**

Enthält die Aktualisierungsvorgänge für eine Uploadtransaktion für eine Tabelle.

**Visual Basic-Syntax**

```
Public Interface UpdateDataReader Implements System.Data.IDataReader
```

**C#-Syntax**

```
public interface UpdateDataReader : System.Data.IDataReader
```

**Basisklassen**

- [System.Data.IDataReader](#)

**Mitglieder**

Alle Mitglieder der UpdateDataReader-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">Close-Methode</a> (geerbt aus System.Data.IDataReader)	Schließt das <a href="#">System.Data.IDataReader</a> -Objekt.
<a href="#">GetSchemaTable-Methode</a> (geerbt aus System.Data.IDataReader)	Gibt ein <a href="#">System.Data.DataTable</a> -Objekt zurück, das die Spaltenmetadaten des <a href="#">System.Data.IDataReader</a> -Objekts beschreibt

Name	Beschreibung
<a href="#">NextResult-Methode</a> (geerbt aus System.Data.IDataReader)	Rückt das DataReader-Objekt zum nächsten Ergebnis vor, wenn Ergebnisse von Batch-SQL-Anweisungen gelesen werden.
<a href="#">Read-Methode</a> (geerbt aus System.Data.IDataReader)	Rückt das <a href="#">System.Data.IDataReader</a> -Objekt zum nächsten Datensatz vor.
<a href="#">SetNewRowValues-Methode</a>	Legt den Modus dieses DataReader-Objekts so fest, dass neue Spaltenwerte zurückgegeben werden (die Post-Update-Zeile).
<a href="#">SetOldRowValues-Methode</a>	Legt den Modus dieses DataReader-Objekts so fest, dass alte Spaltenwerte zurückgegeben werden (die Pre-Update-Zeile).
<a href="#">Depth-Eigenschaft</a> (geerbt aus System.Data.IDataReader)	Ruft einen Wert ab, der die Tiefe der Verschachtelung für die aktuelle Zeile anzeigt.
<a href="#">IsClosed-Eigenschaft</a> (geerbt aus System.Data.IDataReader)	Ruft einen Wert ab, der angibt, ob das DataReader-Objekt geschlossen ist.
<a href="#">RecordsAffected-Eigenschaft</a> (geerbt aus System.Data.IDataReader)	Ruft die Anzahl der durch eine SQL-Anweisung geänderten, eingefügten oder gelöschten Zeilen ab.

### Bemerkungen

Neue und alte Zeilen lassen sich aufrufen, indem Sie den Modus des DataReader-Objekts auf "alt" oder "neu" setzen. Andernfalls kann die Schnittstelle als normales DataReader-Objekt benutzt werden.

## SetNewRowValues-Methode

Legt den Modus dieses DataReader-Objekts so fest, dass neue Spaltenwerte zurückgegeben werden (die Post-Update-Zeile).

### Visual Basic-Syntax

```
Public Sub SetNewRowValues()
```

### C#-Syntax

```
public void SetNewRowValues()
```

### Bemerkungen

Dies ist der Standardmodus.

## SetOldRowValues-Methode

Legt den Modus dieses DataReader-Objekts so fest, dass alte Spaltenwerte zurückgegeben werden (die Pre-Update-Zeile).

**Visual Basic-Syntax**

```
Public Sub SetOldRowValues()
```

**C#-Syntax**

```
public void SetOldRowValues()
```

# UploadData-Schnittstelle

Kapselt Upload-Vorgänge für die direkte Zeilenbehandlung.

**Visual Basic-Syntax**

```
Public Interface UploadData
```

**C#-Syntax**

```
public interface UploadData
```

**Mitglieder**

Alle Mitglieder der UploadData-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">GetUploadedTableByName-Methode</a>	Ruft die Daten der angegebenen Uploadtabelle in dieser Uploadtransaktion ab.
<a href="#">GetUploadedTables-Methode</a>	Ruft ein Array mit den Daten aller Updatetabellen in dieser Uploadtransaktion ab.

**Bemerkungen**

Eine Uploadtransaktion enthält eine Sammlung von Tabellen mit Zeilenvorgängen. Eine UploadData-Instanz, die eine einzelne Uploadtransaktion darstellt, wird an das Ereignis `handle_UploadData` übergeben.

**Hinweis**

Sie müssen Uploadvorgänge zur direkten Zeilenbehandlung in der Methode verarbeiten, die für das Ereignis `handle_UploadData` registriert ist. UploadData wird nach jedem Aufruf der registrierten Methode gelöscht. Erstellen Sie keine neue Instanz von UploadData für nachfolgende Ereignisse.

Mit den Methoden `UploadData.GetUploadedTables` oder `UploadData.GetUploadedTableByName` rufen Sie `UploadedTableData`-Instanzen ab.

Eine Synchronisation enthält ein UploadData-Objekt, außer wenn die entfernte Datenbank den transaktionalen oder inkrementellen Upload verwendet.

**Beispiel**

Siehe [„handle\\_UploadData \(Verbindungsereignis\)“](#) auf Seite 383.

## GetUploadedTableByName-Methode

Ruft die Daten der angegebenen Uploadtabelle in dieser Uploadtransaktion ab.

### Visual Basic-Syntax

```
Public Function GetUploadedTableByName(  
    ByVal table_name As String  
) As UploadedTableData
```

### C#-Syntax

```
public UploadedTableData GetUploadedTableByName(string table_name)
```

### Parameter

- **table\_name** Der Name der Tabelle, für die die hochgeladenen Daten bestimmt sind

### Rückgabe

Die Daten für den angegebenen Uploadtabellennamen oder der Wert NULL, wenn keine Tabelle gefunden wurde.

### Beispiel

Unter der Annahme, dass Sie eine Methode namens HandleUpload für das Synchronisationsereignis handle\_UploadData verwenden, wird im folgenden Beispiel mit der Methode GetUploadedTableByName eine UploadedTableData-Instanz für die Tabelle remoteOrders zurückgegeben:

```
// The method used for the handle_UploadData event.  
public void HandleUpload(UploadData ut) {  
    UploadedTableData uploaded_tl =  
    ut.GetUploadedTableByName("remoteOrders");  
    // ...  
}
```

## GetUploadedTables-Methode

Ruft ein Array mit Daten aller Uploadtabellen in dieser Uploadtransaktion ab.

### Visual Basic-Syntax

```
Public Function GetUploadedTables() As UploadedTableData()
```

### C#-Syntax

```
public UploadedTableData[] GetUploadedTables()
```

### Rückgabe

Ein Array mit den Daten der Uploadtabelle. Die Reihenfolge der Tabellen im Array ist die gleiche wie die Upload-Reihenfolge für den Client.

### Bemerkungen

Die Reihenfolge der Tabellen im Array entspricht der Reihenfolge, die MobiLink für die SQL-Zeilenbehandlung benutzt. Diese Reihenfolge ist optimal für die Vermeidung von Verletzungen der

referenziellen Integrität. Verwenden Sie diese Tabellenreihenfolge, wenn Ihre Datenquelle eine relationale Datenbank ist.

**Siehe auch**

- [UploadedTableData-Schnittstelle \[MobiLink-Server .NET\] auf Seite 618](#)

**Beispiel**

Unter der Annahme, dass Sie eine Methode namens HandleUpload für das Synchronisationsereignis handle\_UploadData verwenden, wird im folgenden Beispiel mit der Methode GetUploadedTables eine UploadedTableData-Instanz für die aktuelle Upload-Transaktion zurückgegeben:

```
// The method used for the handle_UploadData event.
public void HandleUpload(UploadData ud) {
    UploadedTableData[] tables = ud.GetUploadedTables();
    //...
}
```

# UploadedTableData-Schnittstelle

Kapselt Informationen für eine Uploadtabelle für eine Synchronisation.

**Visual Basic-Syntax**

```
Public Interface UploadedTableData
```

**C#-Syntax**

```
public interface UploadedTableData
```

**Mitglieder**

Alle Mitglieder der UploadedTableData-Schnittstelle, einschließlich aller geerbten Mitglieder.

Name	Beschreibung
<a href="#">GetDeletes-Methode</a>	Ruft ein DataReader-Objekt mit den Löschvorgängen für diese Uploadtabel- lendaten ab.
<a href="#">GetInserts-Methode</a>	Ruft ein DataReader-Objekt mit den Einfügevorgängen für diese Uploadda- bellendaten ab.
<a href="#">GetName-Methode</a>	Ruft den Tabellennamen dieser Instanz ab.
<a href="#">GetSchemaTable-Methode</a>	Ruft ein DataTable-Objekt ab, das die Metadaten für diese Uploadtabelle beschreibt.
<a href="#">GetUpdates-Methode</a>	Ruft ein DataReader-Objekt mit den Aktualisierungsvorgängen für diese Uploadtabellendaten ab.

## Bemerkungen

Die Einfüge-, Aktualisierungs- und Löschvorgänge können alle über das Standard-IDataReader-Objekt für ADO.NET aufgerufen werden. Die Tabellenmetadaten lassen sich über den Aufruf von GetSchemaTable oder die DataReader-Objekte zum Einfügen und Löschen aufrufen. Der DataReader zum Löschen bezieht nur die Primärschlüsselspalten der Tabelle mit ein.

## GetDeletes-Methode

Ruft ein DataReader-Objekt mit den Löschvorgängen für diese Uploadtabellendaten ab.

### Visual Basic-Syntax

```
Public Function GetDeletes() As IDataReader
```

### C#-Syntax

```
public IDataReader GetDeletes()
```

### Rückgabe

Ein DataReader mit Primärschlüsselspalten für gelöschte Zeilen.

## Bemerkungen

Jeder Löschvorgang wird durch die Primärschlüsselwerte dargestellt, die zur eindeutigen Darstellung einer Zeile in dieser Instanzentabelle erforderlich sind.

### Hinweis

Der Index und die Reihenfolge der Spalten entsprechen dem Array der DataTable.PrimaryKey-Eigenschaft für das Schema dieser Tabelle.

## Beispiel

Angenommen, der entfernte Client enthält eine Tabelle namens sparse\_pk. Im folgenden Beispiel wird die GetDeletes-Methode verwendet, um ein DataReader-Objekt für gelöschte Zeilen zu erhalten. In diesem Fall enthält das DataReader-Objekt für Löschvorgänge zwei Primärschlüsselspalten. Notieren Sie den Index jeder Primärschlüsselspalte.

```
CREATE TABLE sparse_pk (
    pcol1 INT NOT NULL,
    col2 VARCHAR(200),
    pcol3 INT NOT NULL,
    PRIMARY KEY (pcol1, pcol3)
);

using iAnywhere.MobiLink.Script;
using System;
using System.IO;
using System.Data;
using System.Text;
...

// The method used for the handle_UploadData event.
public void HandleUpload(UploadData ut) {
```

```

        // Get an UploadedTableData for the sparse_pk table.
        UploadedTableData sparse_pk_table =
        ut.GetUploadedTableByName("sparse_pk");

        // Get deletes uploaded by the MobiLink client.
        using( IDataReader data_reader = sparse_pk_table.GetDeletes() ) {
            while (data_reader.Read()) {
                StringBuilder row_str = new StringBuilder("( ");
                row_str.Append(data_reader.GetString(0)); // pcol1
                row_str.Append(", ");
                row_str.Append(data_reader.GetString(1)); // pcol3
                row_str.Append(")");
                writer.WriteLine(row_str);
            }
        }
    }
}

```

## GetInserts-Methode

Ruft ein DataReader-Objekt mit den Einfügevorgängen für diese Uploadtabellendaten ab.

### Visual Basic-Syntax

```
Public Function GetInserts() As IDataReader
```

### C#-Syntax

```
public IDataReader GetInserts()
```

### Rückgabe

Ein DataReader mit den Einfügevorgängen für diese Tabellendaten.

### Bemerkungen

Jede Einfügung wird durch eine Zeile in der Ergebnismenge dargestellt.

### Beispiel

```

CREATE TABLE sparse_pk (
    pcol1 INT NOT NULL,
    col2 VARCHAR(200),
    pcol3 INT NOT NULL,
    PRIMARY KEY (pcol1, pcol3)
);

using iAnywhere.MobiLink.Script;
using System;
using System.IO;
using System.Data;
using System.Text;
...

// The method used for the handle_UploadData event.
public void HandleUpload(UploadData ut) {

    // Get an UploadedTableData for the sparse_pk table.
    UploadedTableData sparse_pk_table =
    ut.GetUploadedTableByName("sparse_pk");
}

```

```

// Get inserts uploaded by the MobiLink client.
using( IDataReader data_reader = sparse_pk_table.GetInserts() ) {

    while (data_reader.Read()) {
        StringBuilder row_str = new StringBuilder("( ");
        row_str.Append(data_reader.GetString(0)); // pcol1
        row_str.Append(", ");
        if (data_reader.IsDBNull(1)) {
            row_str.Append("<NULL>");
        }
        else {
            row_str.Append(data_reader.GetString(1)); // col2
        }
        row_str.Append(", ");
        row_str.Append(data_reader.GetString(2)); // pcol3
        row_str.Append(")");
        writer.WriteLine(row_str);
    }
}

```

## GetName-Methode

Ruft den Tabellennamen dieser Instanz ab.

### Visual Basic-Syntax

```
Public Function GetName() As String
```

### C#-Syntax

```
public string GetName()
```

### Rückgabe

Der Tabellennamen dieser Instanz.

### Bemerkungen

Dies ist eine Dienstprogrammfunktion. Auf den Tabellennamen kann auch über das Schema für diese Instanz zugegriffen werden.

## GetSchemaTable-Methode

Ruft ein DataTable-Objekt ab, das die Metadaten für diese Uploadtabelle beschreibt.

### Visual Basic-Syntax

```
Public Function GetSchemaTable() As DataTable
```

### C#-Syntax

```
public DataTable GetSchemaTable()
```

### Rückgabe

Ein DataTable-Objekt, das die Spalten-Metadaten beschreibt.

## Bemerkungen

Wenn das DataTable-Objekt Spaltennamen-Informationen enthalten soll, müssen Sie die Clientoption zum Senden von Spaltennamen angeben (das Standardverhalten).

## GetUpdates-Methode

Ruft ein DataReader-Objekt mit den Aktualisierungsvorgängen für diese Uploadtabellendaten ab.

### Visual Basic-Syntax

```
Public Function GetUpdates() As UpdateDataReader
```

### C#-Syntax

```
public UpdateDataReader GetUpdates()
```

### Rückgabe

Ein DataReader-Objekt mit den Aktualisierungsvorgängen für diese Tabellendaten

## Bemerkungen

Jede Zeile in der Ergebnismenge stellt jeweils eine Aktualisierung dar. Der Modus der Ergebnismenge kann zwischen neuen und alten Spaltenwerten gewechselt werden.

## Beispiel

Das folgende Beispiel zeigt, wie Sie die GetUpdates-Methode verwenden:

```
CREATE TABLE sparse_pk (  
    pcoll INT NOT NULL,  
    col2 VARCHAR(200),  
    pcoll3 INT NOT NULL,  
    PRIMARY KEY (pcoll, pcoll3)  
);  
  
using iAnywhere.MobiLink.Script;  
using System;  
using System.IO;  
using System.Data;  
using System.Text;  
...  
  
// The method used for the handle_UploadData event.  
public void HandleUpload(UploadData ut) {  
  
    // Get an UploadedTableData for the sparse_pk table.  
    UploadedTableData sparse_pk_table =  
    ut.GetUploadedTableByName("sparse_pk");  
  
    // Get updates uploaded by the MobiLink client.  
    using( UpdateDataReader data_reader = sparse_pk_table.GetInserts() ) {  
  
        while (data_reader.Read()) {  
            data_reader.SetNewRowValues();  
            StringBuilder row_str = new StringBuilder("New values ( ");  
            row_str.Append(data_reader.GetString(0)); // pcoll  
            row_str.Append(", ");  
            if (data_reader.IsDBNull(1)) {
```

```

        row_str.Append("<NULL>");
    }
    else {
        row_str.Append(data_reader.GetString(1)); // col2
    }
    row_str.Append(", ");
    row_str.Append(data_reader.GetString(2)); // pcol3
    row_str.Append(" ");
    data_reader.SetOldRowValues();
    row_str.Append(" Old Values (");
    row_str.Append(data_reader.GetString(0)); // pcol1
    row_str.Append(", ");
    if (data_reader.IsDBNull(1)) {
        row_str.Append("<NULL>");
    }
    else {
        row_str.Append(data_reader.GetString(1)); // col2
    }
    row_str.Append(", ");
    row_str.Append(data_reader.GetString(2)); // pcol3
    row_str.Append(" ");
    writer.WriteLine(row_str);
}
}
}

```

## LogCallback-Delegat

Wird aufgerufen, wenn der MobiLink-Server eine Meldung ausgibt.

### Visual Basic-Syntax

```

Public Delegate Sub LogCallback(
    ByVal sc As ServerContext,
    ByVal message As LogMessage
)

```

### C#-Syntax

```

public delegate void LogCallback(ServerContext sc, LogMessage message);

```

### Bemerkungen

Implementierungen dieses Delegats können mit den ServerContext- Ereignissen registriert werden und werden anschließend aufgerufen, wenn der MobiLink-Server eine Meldung ausgibt.

### Siehe auch

- [ServerContext.ErrorListener-Ereignis \[MobiLink-Server .NET\] auf Seite 607](#)
- [ServerContext.InfoListener-Ereignis \[MobiLink-Server .NET\] auf Seite 607](#)
- [ServerContext.WarningListener-Ereignis \[MobiLink-Server .NET\] auf Seite 608](#)

## ShutdownCallback-Delegat

Wird aufgerufen, wenn der MobiLink-Server heruntergefahren wird.

**Visual Basic-Syntax**

```
Public Delegate Sub ShutdownCallback(ByVal sc As ServerContext)
```

**C#-Syntax**

```
public delegate void ShutdownCallback(ServerContext sc);
```

**Bemerkungen**

Implementierungen dieses Delegats können mit dem Ereignis ShutdownListener registriert werden und werden anschließend aufgerufen, wenn der MobiLink-Server heruntergefahren wird.

**Siehe auch**

- [ServerContext.ShutdownListener-Ereignis \[MobiLink-Server .NET\] auf Seite 608](#)

# SQLType-Enumeration

Enumeriert alle möglichen ODBC-Datentypen.

**Visual Basic-Syntax**

```
Public Enum SQLType
```

**C#-Syntax**

```
public enum SQLType
```

**Mitglieder**

Mitgliedsname	Beschreibung
SQL_TYPE_NULL	Null-Datentyp
SQL_UNKNOWN_TYPE	Unbekannter Datentyp.
SQL_CHAR	UTF-8-Zeichen-Array einer festgelegten Größe. Hat den .NET-Typ String.
SQL_NUMERIC	Nummerischer Wert einer festgelegten Größe und Gesamtstellenzahl. Hat den .NET-Typ Decimal.
SQL_DECIMAL	Dezimalzahl einer festgelegten Größe und Gesamtstellenzahl. Hat den .NET-Typ Decimal.
SQL_INTEGER	32-Bit-Ganzzahl. Hat den .NET-Typ Int32.

<b>Mitgliedsname</b>	<b>Beschreibung</b>
SQL_SMALLINT	16-Bit-Ganzzahl. Hat den .NET-Typ Int16
SQL_FLOAT	Gleitkommazahl mit einer vom ODBC-Treiber festgelegten Gesamtstellenzahl. Hat den .NET-Typ Double.
SQL_REAL	Gleitkommazahl mit einfacher Präzision. Hat den .NET-Typ Single.
SQL_DOUBLE	Gleitkommazahl mit doppelter Präzision. Hat den .NET-Typ Double.
SQL_DATE	Ein Datumswert. Hat den .NET-Typ DateTime.
SQL_DATETIME	Ein Datums- und Uhrzeitwert Hat den .NET-Typ DateTime.
SQL_TIME	Eine Uhrzeit. Hat den .NET-Typ DateTime
SQL_INTERVAL	Ein Zeitintervall. Hat den .NET-Typ TimeSpan
SQL_TIMESTAMP	Ein Zeitstempel. Hat den .NET-Typ DateTime.
SQL_VARCHAR	Eine mit einem Nullwert abgeschlossene UTF-8-Zeichenfolge mit einer vom Benutzer festgelegten maximalen Länge. Hat den .NET-Typ String.
SQL_TYPE_DATE	Ein Datumswert. Hat den .NET-Typ DateTime.

<b>Mitgliedsname</b>	<b>Beschreibung</b>
SQL_TYPE_TIME	Eine Uhrzeit. Hat den .NET-Typ DateTime.
SQL_TYPE_TIMESTAMP	Ein Zeitstempel. Hat den .NET-Typ DateTime
SQL_DEFAULT	Ein Standardtyp. Hat keinen Typ.
SQL_ARD_TYPE	Ein ARD-Objekt. Hat keinen Typ.
SQL_BIT	Ein einzelnes Bit. Hat den .NET-Typ Boolean.
SQL_TINYINT	8-Bit-Ganzzahl. Hat den .NET-Typ SByte.
SQL_BIGINT	64-Bit-Ganzzahl. Hat den .NET-Typ Int64.
SQL_LONGVARBINARY	Binärdaten variabler Länge mit einer vom Treiber abhängigen maximalen Länge. Hat den .NET-Typ byte[].
SQL_VARBINARY	Binärdaten variabler Länge mit einer vom Benutzer festgelegten maximalen Länge. Hat den .NET-Typ byte[].
SQL_BINARY	Binärdaten mit fester Länge. Hat den .NET-Typ byte[].
SQL_LONGVARCHAR	Eine mit einem Nullwert abgeschlossene UTF-8-Zeichenfolge mit einer vom Treiber abhängigen maximalen Länge. Hat den .NET-Typ String.

Mitgliedsname	Beschreibung
SQL_GUID	Eine GUID.  Hat den .NET-Typ Guid.
SQL_WCHAR	Unicode-Zeichen-Array fester Größe.  Hat den .NET-Typ String.
SQL_WVARCHAR	Eine mit einem Nullwert abgeschlossene Unicode-Zeichenfolge mit einer benutzerdefinierten maximalen Länge. Hat den .NET-Typ String.
SQL_WLONGVARCHAR	Eine mit einem Nullwert abgeschlossene Unicode-Zeichenfolge mit einer vom Treiber abhängigen maximalen Länge. Hat den .NET-Typ String.
SQL_SS_TIMESTAMPOFFSET	Zeitstempel mit Zeitzonen-Offset. Hat den .NET-Typ <code>Anywhere.MobiLink.Script.DateTimeWithTimeZone</code> . Dies kann nur für Microsoft SQL Server- und Oracle-Datenbanken verwendet werden.

### Bemerkungen

Jeder SQLType-Wert entspricht einem .NET-Typ.

## Direkte Zeilenbehandlung

### Hinweis

Die direkte Zeilenbehandlung ist eine fortgeschrittene MobiLink-Funktion. Um sie verwenden zu können, sind umfassende Kenntnisse über die Erstellung einer MobiLink-Anwendung und die Verwendung der MobiLink-APIs erforderlich. Siehe:

- [MobiLink - Erste Orientierung](#)
- [MobiLink - Serveradministration auf Seite 1](#)
- [MobiLink - Clientadministration](#)

MobiLink unterstützt zwei Methoden zur Bearbeitung von Zeilen: SQL und direkt. Sie können beide separat oder gemeinsam verwenden.

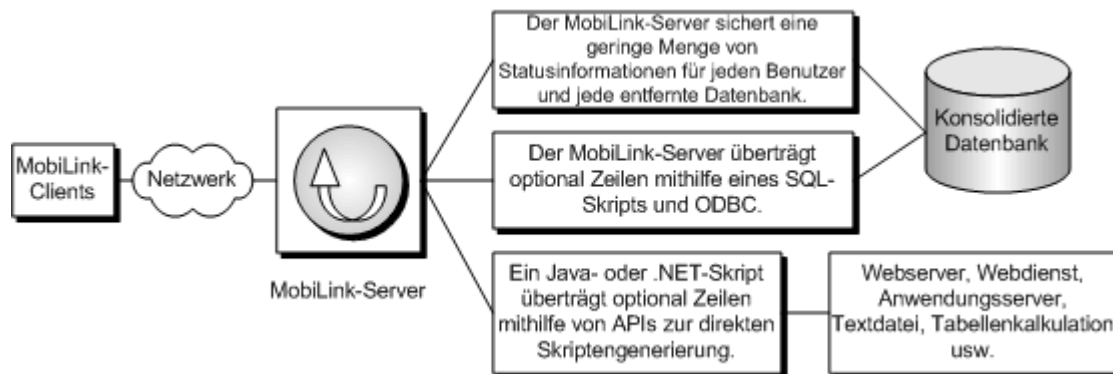
- **Zeilenbehandlung über SQL** Sie ermöglicht die Synchronisation von entfernten Daten mit einer unterstützten konsolidierten Datenbank. SQL-basierte Ereignisse bieten eine stabile Schnittstelle für die Konfliktlösung und andere Synchronisationsaufgaben. Sie können SQL entweder direkt verwenden oder unter Verwendung der MobiLink-Server-APIs für Java und .NET zurückgeben.

- **Direkte Zeilenbehandlung** Sie ermöglicht die Synchronisation entfernter Daten mit jeder zentralen Datenquelle. Die direkte Zeilenbehandlung ermöglicht den Zugriff auf unbearbeitete synchronisierte Daten mittels spezieller MobiLink-Ereignisse und der MobiLink-Server-API für Java und .NET.

Tatsächlich können Sie alle Datenquellen synchronisieren, darunter Anwendungen, Webserver, Webservices, Anwendungsserver, Textdateien, Tabellenkalkulationen, nicht-relationale Datenbanken oder RDBMS, die nicht als konsolidierte Datenbanken verwendet werden können. Zum Speichern Ihrer MobiLink-Systemtabellen benötigen Sie weiterhin eine konsolidierte Datenbank. Viele Implementierungen der direkten Zeilenbehandlung synchronisieren sowohl die konsolidierte Datenbank als auch andere Datenquellen.

Für die direkte Zeilenbehandlung sind Erfahrungen bei der Erstellung von konsolidierten MobiLink-Datenbanken, dem Hinzufügen von Synchronisationsskripten und der Erstellung von entfernten MobiLink-Benutzern erforderlich.

Das folgende Diagramm veranschaulicht die grundlegende MobiLink-Architektur:



## Komponenten der direkten Zeilenbehandlung

Zur Implementierung der direkten Zeilenbehandlung können Sie zwei Synchronisationsereignisse zusammen mit verschiedenen Schnittstellen und Methoden in der MobiLink-Server-API für Java und .NET verwenden.

### Direkte Synchronisationsereignisse

Die direkte Zeilenbehandlung ermöglicht den direkten Zugriff auf den Upload- und den Download-Datenstrom. Dazu schreiben Sie Java- oder .NET-Methoden für die Synchronisationsereignisse `handle_UploadData` und `handle_DownloadData`.

- **handle\_UploadData** Akzeptiert einen einzelnen UploadData-Parameter, der Vorgänge umfasst, die ein MobiLink-Client für eine einzelne Uploadtransaktion hochgeladen hat. Siehe:
  - „Direkte Uploads“ auf Seite 631
  - „handle\_UploadData (Verbindungsereignis)“ auf Seite 383

- **handle\_DownloadData** Ermöglicht die Festlegung von Download-Vorgängen unter Verwendung der DownloadData-Schnittstelle. Siehe:
  - „Direkte Downloads“ auf Seite 637
  - „handle\_DownloadData (Verbindungsereignis)“ auf Seite 371

## Komponenten der MobiLink-Server-API für die direkte Zeilenbehandlung

Für die Java-API:

- [DBConnectionContext.getDownloadData-Methode \[MobiLink-Server Java\]](#)
- [DownloadData-Schnittstelle \[MobiLink-Server Java\]](#)
- [DownloadTableData-Schnittstelle \[MobiLink-Server Java\]](#)
- [UpdateResultSet-Schnittstelle \[MobiLink-Server Java\]](#)
- [UploadData-Schnittstelle \[MobiLink-Server Java\]](#)
- [UploadedTableData-Schnittstelle \[MobiLink-Server Java\]](#)

Für die .NET-API:

- [DBConnectionContext.GetDownloadData-Methode \[MobiLink-Server .NET\]](#)
- [DownloadData-Schnittstelle \[MobiLink-Server .NET\]](#)
- [DownloadTableData-Schnittstelle \[MobiLink-Server .NET\]](#)
- [UpdateDataReader-Schnittstelle \[MobiLink-Server .NET\]](#)
- [UploadedTableData-Schnittstelle \[MobiLink-Server .NET\]](#)
- [UploadData-Schnittstelle \[MobiLink-Server .NET\]](#)

## Setup für die direkte Zeilenbehandlung

Für die direkte Zeilenbehandlung sind Erfahrungen bei der Erstellung von konsolidierten MobiLink-Datenbanken, dem Hinzufügen von Synchronisationsskripten und der Erstellung von entfernten MobiLink-Benutzern erforderlich.

Im Folgenden finden Sie einen Überblick darüber, wie Sie eine Synchronisation mit einer Datenquelle durchführen, bei der es sich nicht um eine konsolidierte Datenbank handelt.

1. Richten Sie, falls nicht bereits vorhanden, eine konsolidierte Datenbank ein.

Unabhängig davon, ob Sie mit einer konsolidierten Datenbank synchronisieren, muss eine solche Datenbank zum Speichern der MobiLink-Systemtabellen vorhanden sein.

Siehe „[MobiLink - konsolidierte Datenbanken](#)“ auf Seite 1.

2. Wenn Sie Uploads verarbeiten möchten, schreiben Sie mithilfe der UploadData-Schnittstelle eine öffentliche Methode und registrieren sie diese für das handle\_UploadData-Verbindungsereignis.

Siehe „[Direkte Uploads](#)“ auf Seite 631.

3. Wenn Sie Downloads verarbeiten möchten, schreiben Sie mithilfe der DownloadData-Schnittstelle eine öffentliche Methode und registrieren sie diese für das handle\_DownloadData-Verbindungsereignis (oder ein anderes Ereignis).

Siehe „Direkte Downloads“ auf Seite 637.

### Weitere Ressourcen für die ersten Schritte

- „Praktische Einführung: Verwenden der direkten Zeilenbehandlung“ [*MobiLink - Erste Orientierung*]
- <http://www.sybase.com/detail?id=1058600#319> (Sie benötigen ein Login für Sybase.com, um diese Seite anzuzeigen.)
- „Java-Synchronisationslogik einrichten“
- „Synchronisationsskripten in .NET implementieren“

Sie können Fragen im SQL Anywhere-Forum veröffentlichen: <http://sqlanywhere-forum.sybase.com>.

## Tipps für die Entwicklung mit direkter Zeilenbehandlung

### Eindeutige Primärschlüssel

Für die MobiLink-Synchronisation mit direkter Zeilenbehandlung muss die Datenquelle eindeutige Primärschlüssel besitzen, die nicht aktualisiert werden. In einer nicht relationalen Datenquelle, z.B. einer Tabellenkalkulation oder Textdatei, muss eine Spalte eindeutige, unveränderliche Werte enthalten, die die Zeile identifizieren.

Siehe „Eindeutige Primärschlüssel“ auf Seite 117.

### Spaltennamen

Die Spaltennamen von Tabellen werden immer vom Client gesendet und können für die direkte Zeilenbehandlung verwendet werden. Alternativ können Sie mithilfe von Spaltenindizes auf Zeileninformationen zugreifen, basierend auf der aus der entfernten Datenbank gesendeten Spaltenreihenfolge.

### Letzte Downloadzeit für Downloads verwenden

Sofern möglich, sollten Sie Ihre Anwendung mit direkter Zeilenbehandlung wie eine zeitstempelbasierte SQL-Anwendung einrichten und die Spalte `last_modified` und darauf basierende Download-Daten verwalten. Bei dieser Methode werden unvorhergesehene Probleme vermieden, die bei anderen Download-Methoden auftreten können.

Siehe „Implementieren zeitstempelbasierter Downloads“ auf Seite 107.

### Transaktionsverwaltung für Uploads

Es ist nicht möglich, Transaktionen in konsolidierten MobiLink-Datenbanken festzuschreiben. Sie können jedoch Transaktionen in der Datenquelle festschreiben, die die direkte Zeilenbehandlung verwendet. Beachten Sie bei der Einstellung der Transaktionsverwaltung folgende Tipps:

- **Upload vor MobiLink-Festschreibungen festschreiben** Bei der Anwendung eines Uploads schreibt MobiLink die Änderungen am Ende des `end_upload`-Ereignisses fest. Sie müssen sicherstellen, dass alle Upload-Änderungen, die beibehalten werden sollen, vor dem Ende des `end_upload`-Skripts festgeschrieben werden. Andernfalls kann bei einem Fehler oder Ausfall ein Zustand erreicht werden, in dem die Anwendung davon ausgeht, dass der Upload angewendet wurde, aber MobiLink die Daten nicht übernommen hat. Dies könnte zu einem Datenverlust führen.

- **Redundante Uploads verarbeiten** Falls ein Fehler oder Ausfall auftritt, nachdem die Anwendung und bevor der MobiLink-Server eine hochgeladene Zeile festschreibt, kommt es möglicherweise zu einem inkonsistenten Zustand des MobiLink-Servers und der Datenquelle. Sie können dieses Problem beheben, indem Sie redundante Uploads zulassen und eine Logik einsetzen, die sicherstellt, dass der redundante Upload richtig angewendet wird. Vor allem muss sichergestellt werden, dass der Upload nicht erneut angewendet wird, wenn die Anwendung ihn ein zweites Mal sendet.

## Fehler behandeln

Für die Behandlung von Fehlern muss die Verwendung einer geeigneten Transaktionsverwaltung sichergestellt werden, wie oben beschrieben. Außerdem muss der Java- oder .NET-Code, der Zeilen verarbeitet, alle auftretenden Ausnahmebedingungen an den MobiLink-Server senden. Falls ein Fehler auftritt, bevor der MobiLink-Server oder Ihre Anwendung Änderungen festgeschrieben hat, wird die Transaktion von MobiLink zurückgesetzt und somit ein konsistenter Zustand der Anwendung bewahrt.

## Klasseninstanz

MobiLink erstellt für die direkte Zeilenbehandlung für jede Datenbankverbindung eine Klasseninstanz. Die Klasseninstanz wird am Ende einer Synchronisation nicht zerstört: Sie wird erst zerstört, wenn die Datenbankverbindung geschlossen wird. Variablen auf Klassenebene behalten solange ihre Werte aus früheren Synchronisationen bei.

# Direkte Uploads

Direkte Uploads verarbeiten

1. Registrieren Sie eine Java-Methode für das Verbindungsereignis `handle_UploadData`. Siehe [„handle\\_UploadData \(Verbindungsereignis\)“ auf Seite 383](#).
2. Erstellen Sie eine Methode für das Verbindungsereignis `handle_UploadData`. Dieses Ereignis akzeptiert nur einen `UploadData`-Parameter. Siehe:
  - [UploadData-Schnittstelle \[MobiLink-Server Java\] auf Seite 528](#)
  - [UploadData-Schnittstelle \[MobiLink-Server .NET\] auf Seite 616](#)

Das Ereignis `handle_UploadData` wird bei jeder Synchronisation normalerweise einmal aufgerufen. Bei SQL Anywhere-Clients, die Uploads auf der Transaktionsstufe verwenden, kann jedoch mehr als nur ein Upload pro Synchronisation durchgeführt werden. In einem solchen Fall wird `handle_UploadData` einmal pro Transaktion aufgerufen.

Weitere Hinweise über `dbmlsync`-Uploads auf der Transaktionsstufe finden Sie unter [„dbmlsync-Option -tu“ \[MobiLink - Clientadministration\]](#).

Allgemeine Informationen zum Schreiben von Java- oder .NET-Synchronisationsskripten finden Sie unter:

- [„Schreiben eines Synchronisationsskripts in Java“ auf Seite 463](#)
- [„Synchronisationsskripten in .NET“ auf Seite 536](#)

Informationen über die Registrierung von Ereignissen auf der Verbindungsstufe finden Sie unter:

- „[ml\\_add\\_java\\_connection\\_script-Systemprozedur](#)“ auf Seite 655
- „[ml\\_add\\_dnet\\_connection\\_script-Systemprozedur](#)“ auf Seite 653

### Klassen für direkte Uploads

Die MobiLink-Server-APIs für Java .NET stellen folgende Klassen für die Verarbeitung direkter Uploads bereit:

- **UploadData** Umfasst eine einzelne Uploadtransaktion. Eine Uploadtransaktion enthält eine Sammlung von Tabellen mit Zeilenvorgängen. Siehe:
  - [UploadData-Schnittstelle \[MobiLink-Server Java\] auf Seite 528](#)
  - [UploadData-Schnittstelle \[MobiLink-Server .NET\] auf Seite 616](#)
- **UploadedTableData** Umfasst die Einfüge-, Aktualisierungs- und Löschvorgänge einer Tabelle, die von einem MobiLink-Client hochgeladen wurden. Für Java geben die UploadedTableData-Methoden eine Instanz von UpdateResultSet zurück. Für .NET geben die UploadedTableData-Methoden eine Instanz einer UpdateDataReader-Schnittstelle zurück. Durchsuchen Sie die Ergebnismenge IDataReader, um die hochgeladenen Zeilenvorgänge zu verarbeiten. Siehe:
  - [UploadedTableData-Schnittstelle \[MobiLink-Server Java\] auf Seite 530](#)
  - [UploadedTableData-Schnittstelle \[MobiLink-Server .NET\] auf Seite 618](#)
- **UpdateResultSet** Für Java stellt diese Klasse eine Aktualisierungsergebnismenge dar, die von der Methode UploadedTableData getUpdates zurückgegeben wurde. Sie erweitert java.sql.ResultSet so, dass es spezielle Methoden zum Abrufen neuer und alter Versionen einer aktualisierten Zeile umfasst.

Siehe [UpdateResultSet-Schnittstelle \[MobiLink-Server Java\] auf Seite 526](#).

Für .NET stellt die Schnittstelle UpdateDataReader eine Gruppe von Zeilen dar, die von der Methode UploadedTableData GetUpdates geliefert wurden. Sie erweitert IDataReader so, dass es spezielle Methoden zum Abrufen neuer und alter Versionen einer aktualisierten Zeile umfasst.

Siehe [UpdateDataReader-Schnittstelle \[MobiLink-Server .NET\] auf Seite 614](#).

### Beispiel

Siehe „[handle\\_UploadData \(Verbindungsereignis\)](#)“ auf Seite 383.

## Konflikte bei direkten Uploads

Sendet ein MobiLink-Client eine aktualisierte Zeile an den MobiLink-Server, so enthält diese Zeile nicht nur die aktualisierten Werte (die Post-Image-Zeile), sondern auch eine Kopie der Werte der alten Zeile (Pre-Image-Zeile oder alte Zeile) aus der letzten Synchronisation mit dem MobiLink-Server. Wenn die Pre-Image-Zeile nicht mit den aktuellen Werten in Ihrer zentralen Datenquelle übereinstimmt, wird ein Konflikt festgestellt.

## SQL-basierte Konfliktlösung

Bei SQL-basierten Uploads ist die konsolidierte MobiLink-Datenbank Ihre zentrale Datenquelle, und MobiLink stellt spezielle Ereignisse für die Konflikterkennung und -lösung bereit.

Siehe „Konfliktverarbeitung“ auf Seite 124.

## Konfliktlösung mit der direkten Zeilenbehandlung

Bei direkten Uploads können Sie zur Konflikterkennung über Programme auf neue und alte Zeilen zugreifen.

UpdateResultSet (von der Methode UploadedTableData.getUpdates geliefert) erweitert die Standardergebnismengen von Java oder .NET so, dass diese spezielle Methoden zum Bearbeiten von Konflikten umfassen. setNewRowValues stellt UpdateResultSet so ein, das neue aktualisierte Werte von einem entfernten Client zurückgegeben werden (Standardmodus). setOldRowValues stellt UpdateResultSet so ein, dass alte Zeilenwerte zurückgegeben werden.

## Konflikte mit der direkten Zeilenbehandlung erkennen

Bei der Verwendung der UpdateResultSet-Methode .setOldRowValues erhalten Sie die Werte einer Zeile in der entfernten Datenbank vor ihrer Änderung. Sie vergleichen die zurückgegebenen Zeilenwerte mit den vorhandenen Zeilenwerten in Ihrer Datenquelle. Wenn die verglichenen Zeilen nicht gleich sind, liegt ein Konflikt vor.

## Konflikte mit der direkten Zeilenbehandlung lösen

Wenn Sie einen Konflikt während eines Uploads erkannt haben, können Sie benutzerdefinierte Geschäftslogik verwenden, um ihn aufzulösen. Die Lösung wird von Ihrem Java- oder .NET-Code verarbeitet.

## Beispiel

Angenommen, Sie protokollieren den Lagerbestand in einem XML-Dokument und möchten das Dokument als zentrale Datenquelle verwenden. Benutzer1 verwendet eine Ihrer entfernten Datenbanken namens Remote1. Benutzer2 verwendet eine andere entfernte Datenbank namens Remote2.

Ihr XML-Dokument, Benutzer1 und Benutzer2 starten jeweils mit einem Lager, das zehn Artikel umfasst. Benutzer1 verkauft drei Artikel und aktualisiert den Lagerwert von Remote1 auf sieben Artikel. Benutzer2 verkauft vier Artikel und aktualisiert das Lager von Remote2 auf sechs Artikel. Wenn Remote1 synchronisiert, wird die zentrale Datenbank auf sieben Artikel aktualisiert. Wenn Remote2 synchronisiert, wird ein Konflikt festgestellt, da der Wert des Lagers nicht mehr zehn Artikel ist. Um diesen Konflikt programmatisch zu lösen, benötigen Sie drei Zeilenwerte:

- Den aktuellen Wert in der zentralen Datenquelle
- Den neuen Zeilenwert, den Remote2 ausgelesen hat
- Den alten Zeilenwert, den Remote2 bei der letzten Synchronisation erhalten hat

In diesem Fall würde die Geschäftslogik folgende Berechnung verwenden, um den neuen Lagerwert zu ermitteln und den Konflikt zu lösen:

```
current data source - (old remote - new remote)
-> 7 - (10-6) = 3
```

Die nachfolgenden Prozeduren für Java und .NET zeigen, wie Sie diesen Konflikt für direkte Uploads lösen können. Die gezeigte Tabelle dient dabei als Beispiel:

```
CREATE TABLE remoteOrders
(
    pk integer primary key not null,
    inventory integer not null
);
```

## Java

1. Registrieren Sie eine Java-Methode für das handle\_UploadData-Verbindungsereignis.

Siehe „[handle\\_UploadData \(Verbindungsereignis\)](#)“ auf Seite 383.

Die folgende gespeicherte Prozedur registriert beispielsweise eine Java-Methode namens HandleUpload für das Verbindungsereignis handle\_UploadData, wenn die Skriptversion ver1 synchronisiert wird. Sie führen diese gespeicherte Prozedur für Ihre konsolidierte MobiLink-Datenbank aus.

```
call ml_add_java_connection_script( 'ver1',
    'handle_UploadData',
    'OrderProcessor.HandleUpload' )
```

Weitere Hinweise zum Registrieren von Methoden für Synchronisationsereignisse finden Sie unter:

- „[Hinzufügungen und Löschungen von Skripten](#)“ auf Seite 258
- „[ml\\_add\\_java\\_connection\\_script-Systemprozedur](#)“ auf Seite 655

2. Rufen Sie UpdateResultSet für eine Tabelle im Upload ab.

Die Methode OrderProcessor.HandleUpload gibt UpdateResultSet für die remoteOrders-Tabelle zurück.

```
// method for handle_UploadData event
public void HandleUpload( UploadData u_data )
{
    // Get UploadedTableData for the remoteOrders table.
    UploadedTableData u_table =
    u_data.getUploadedTableByName( "remoteOrders" );

    // Get an UpdateResultSet for the remoteOrders table.
    UpdateResultSet update_rs = u_table.getUpdates();

    // (Continued...)
```

3. Rufen Sie für jede Aktualisierung die aktuellen Werte in Ihrer zentralen Datenquelle ab.

In diesem Beispiel gibt die Methode UpdateResultSet getInt einen Ganzzahlwert für die Primärschlüsselspalte (die erste Spalte) zurück. Sie können die Methode getMyCentralData implementieren und anschließend mit ihr Daten aus Ihrer zentralen Datenquelle abrufen.

```

while( update_rs.next() )
{
    // Get central data source values.

    // Get the primary key value.
    int pk_value = update_rs.getInt(1);

    // Get central data source values.
    int central_value = getMyCentralData(pk_value);

    // (Continued...)

```

4. Rufen Sie für jede Aktualisierung die alten und die neuen Werte ab, die der MobiLink-Client hochgeladen hat.

Im Beispiel werden UpdateResultSet setOldRowValues und UpdateResultSet setNewRowValues für alte bzw. neue Werte verwendet.

```

// Set mode for old row values.
update_rs.setOldRowValues();

// Get the _old_ stored value on the remote.
int old_value = update_rs.getInt(2);

// Set mode for new row values.
update_rs.setNewRowValues();

// Get the _new_ updated value on the remote.
int new_value = update_rs.getInt(2);

// (Continued...)

```

5. Überprüfen Sie jede Aktualisierung auf Konflikte.

Ein Konflikt liegt vor, wenn der alte Zeilenwert nicht mit dem aktuellen Wert in der zentralen Datenquelle übereinstimmt. Um den Konflikt zu lösen, wird unter Verwendung der Geschäftslogik ein gelöster Wert berechnet. Tritt kein Konflikt auf, wird die zentrale Datenquelle mit dem neuen entfernten Wert aktualisiert. Sie können die Methode setMyCentralData implementieren und anschließend mit ihr die Aktualisierung ausführen.

```

// Check if there is a conflict.

if(old_value == central_value)
{
    // No conflict.
    setMyCentralData(pk_value, new_value);
}
else
{
    // Handle the conflict.
    int inventory = old_value - new_value;
    int resolved_value = central_value - inventory;

    setMyCentralData(pk_value, resolved_value);
}
}

```

**.NET**

1. Registrieren Sie eine Methode für das Verbindungsereignis `handle_UploadData`.

Die folgende gespeicherte Prozedur registriert beispielsweise eine .NET-Methode namens `HandleUpload` für das Verbindungsereignis `handle_UploadData`, wenn die Skriptversion `ver1` synchronisiert wird. Sie führen diese gespeicherte Prozedur für Ihre konsolidierte MobiLink-Datenbank aus.

```
call ml_add_dnet_connection_script( 'ver1',  
    'handle_UploadData',  
    'MyScripts.OrderProcessor.HandleUpload' )
```

Weitere Hinweise zum Registrieren von Methoden für Synchronisationsereignisse finden Sie unter:

- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)
- [„ml\\_add\\_dnet\\_connection\\_script-Systemprozedur“ auf Seite 653](#)

2. Rufen Sie `UpdateDataReader` für eine Tabelle im Upload ab.

Die Methode `MyScripts.OrderProcessor.HandleUpload` ruft ein `UpdateResultSet` für die Tabelle `remoteOrders` ab:

```
// method for handle_UploadData event  
public void HandleUpload( UploadData u_data )  
{  
  
    // Get UploadedTableData for the remoteOrders table.  
    UploadedTableData u_table =  
    u_data.GetUploadedTableByName("remoteOrders");  
  
    // Get an UpdateDataReader for the remoteOrders table.  
    UpdateDataReader update_dr = u_table.GetUpdates();  
  
    // (Continued...)
```

3. Rufen Sie für jede Aktualisierung die aktuellen Werte in Ihrer zentralen Datenquelle ab.

In diesem Beispiel gibt die Methode `UpdateDataReader` `GetInt32` einen Ganzzahlwert für die Primärschlüsselspalte (die erste Spalte) zurück. Sie können die Methode `getMyCentralData` implementieren und anschließend mit ihr Daten aus Ihrer zentralen Datenquelle abrufen.

```
while( update_dr.Read() )  
{  
    // Get central data source values.  
  
    // Get the primary key value.  
    int pk_value = update_dr.GetInt32(0);  
  
    // Get central data source values.  
    int central_value = getMyCentralData(pk_value);  
  
    // (Continued...)
```

4. Rufen Sie für jede Aktualisierung die alten und die neuen Werte ab, die der MobiLink-Client hochgeladen hat.

Im Beispiel werden `UpdateResultSet` `setOldRowValues` und `UpdateResultSet` `setNewRowValues` für alte bzw. neue Werte verwendet.

```
// Set mode for old row values.
update_dr.SetOldRowValues();

// Get an _old_ value.
int old_value = update_dr.GetInt32(1);

// Set mode for new row values.
update_dr.SetNewRowValues();

// Get the _new_ updated value.
int new_value = update_dr.GetInt32(1);

// (Continued...)
```

## 5. Überprüfen Sie jede Aktualisierung auf Konflikte.

Ein Konflikt liegt vor, wenn der alte Zeilenwert nicht mit dem aktuellen Wert in der zentralen Datenquelle übereinstimmt. Um den Konflikt zu lösen, wird unter Verwendung der Geschäftslogik ein gelöster Wert berechnet. Tritt kein Konflikt auf, wird die zentrale Datenquelle mit dem neuen entfernten Wert aktualisiert. Sie können die Methode `setMyCentralData` implementieren und anschließend mit ihr die Aktualisierung ausführen.

```
// Check if there is a conflict.

if(old_value == central_value)
{
    // No conflict.
    setMyCentralData(pk_value, new_value);
}
else
{
    // Handle the conflict.
    int inventory = old_value - new_value;
    int resolved_value = central_value - inventory;

    setMyCentralData(pk_value, resolved_value);
}
}
```

## Direkte Downloads

Direkte Downloads verarbeiten

1. Registrieren Sie eine Java- oder .NET-Methode für das Verbindungsereignis `handle_DownloadData`.  
Siehe [„handle\\_DownloadData \(Verbindungsereignis\)“ auf Seite 371](#).

2. Erstellen Sie eine Methode für das Verbindungsereignis `handle_DownloadData`. In diesem Ereignis verwenden Sie eine Instanz von `DBConnectionContext`, um eine `DownloadData`-Instanz für die aktuelle Synchronisation abzurufen. Siehe:

- [DBConnectionContext-Schnittstelle \[MobiLink-Server Java\] auf Seite 477](#)
- [DownloadData-Schnittstelle \[MobiLink-Server Java\] auf Seite 483](#)
- [DBConnectionContext-Schnittstelle \[MobiLink-Server .NET\] auf Seite 565](#)
- [DownloadData-Schnittstelle \[MobiLink-Server .NET\] auf Seite 586](#)

Sie können den gesamten direkten Download im Synchronisationsereignis `handle_DownloadData` erstellen. Alternativ können Sie andere Synchronisationsereignisse verwenden, um direkte Download-Vorgänge einzurichten. Sie müssen jedoch ein Skript `handle_DownloadData` erstellen, sogar dann, wenn die Methode nichts tut. Falls Sie den direkten Download in einem anderen Ereignis als `handle_DownloadData` verarbeiten, kann das Ereignis nicht vor `begin_synchronization` und nicht nach `end_download` stattfinden.

Informationen über die Reihenfolge von Ereignissen finden Sie unter „[Komplettes MobiLink-Ereignismodell](#)“ auf Seite 276.

### Klassen für direkte Downloads

Die MobiLink-Server-APIs für Java und .NET stellen folgende Klassen für das Erstellen direkter Downloads bereit:

- **DownloadData** Verpackt Download-Tabellen mit Vorgängen, die während der Synchronisation an einen entfernten Client gesendet werden. Siehe:
  - [DownloadData-Schnittstelle \[MobiLink-Server Java\] auf Seite 483](#)
  - [DownloadData-Schnittstelle \[MobiLink-Server .NET\] auf Seite 586](#)
- **DownloadTableData** Umfasst die Upsert-Vorgänge (Aktualisierung und Einfügung) und die Löschvorgänge für den Download auf einen MobiLink-Client.

Für Java liefern die Methoden `DownloadTableData` eine Instanz eines JDBC `PreparedStatement`. In Java fügen Sie dem Download eine Zeile hinzu, indem Sie die Spaltenwerte der vorbereiteten Anweisung einrichten und anschließend die vorbereitete Anweisung ausführen.

Für .NET liefern die Methoden `DownloadTableData` eine Instanz eines .NET `IDbCommand`. In .NET fügen Sie dem Download eine Zeile hinzu, indem Sie die Spaltenwerte der Anweisung einrichten und anschließend die vorbereitete Anweisung ausführen.

Siehe:

- [DownloadTableData-Schnittstelle \[MobiLink-Server Java\] auf Seite 485](#)
- [DownloadTableData-Schnittstelle \[MobiLink-Server .NET\] auf Seite 588](#)

### Beispiel

Siehe „[handle\\_DownloadData \(Verbindungsereignis\)](#)“ auf Seite 371.

---

# MobiLink-Referenz

Dieser Abschnitt enthält MobiLink-Referenzmaterial.

## MobiLink Replay-C++-Callback-Funktionen

In diesem Abschnitt wird die öffentliche Struktur der Datei *mlreplaycallbacks.cpp* beschrieben, die vom Dienstprogramm *mlgenreplayapi* generiert werden kann. Er enthält eine komplette Liste der Callback-Funktionen, die Sie entwickeln können, um die Daten anzupassen, die während einer Wiedergabesitzung mit dem *mlreplay*-Dienstprogramm auf den MobiLink-Server hochgeladen werden.

### Hinweis

Wenn keine Callback-Funktionen verwendet werden, müssen Sie warten, bis alle simulierten Clients bereit sind, bevor diese mit der Wiedergabe beginnen können. Simulierte Clients können die Wiedergabe nicht ausführen, wenn ein simulierter Client nicht erstellt oder initialisiert werden kann.

### Siehe auch

- „Dienstprogramm MobiLink Replay (*mlreplay*)“ auf Seite 702

## Callback-Funktion

### "CreateAndInitMLReplayUploadTransaction"

Wird zum Erstellen und Initialisieren einer Uploadtransaktion verwendet. Aufrufe erfolgen einmal anfänglich und danach einmal pro Uploadtransaktion, pro Synchronisation und pro simuliertem Client sowie pro Wiederholung für alle Wiederholungen größer als 1.

### Syntax

```
_MLREPLAY_EXPORT bool _MLREPLAY_CDECL  
CreateAndInitMLReplayUploadTransaction(  
    IMLReplayUploadTransaction ** uploadTrans,  
    const IMLReplayAPICallbacks * mlrAPICallbacks  
)
```

### Parameter

- **uploadTrans** Eine Implementierung von *IMLReplayUploadTransaction*, die *mlreplay* zum Füllen der Wiedergabesitzung mit benutzerdefinierten Daten verwendet.
- **mlrAPICallbacks** Callback-Funktionen zum Bereitstellen von Informationen aus *mlreplay*.

### Rückgabe

TRUE bei Erfolg; gibt FALSE zurück bei einem Fehler, der zum Abbruch der Wiedergabesitzung führt.

### Bemerkungen

Diese Callback-Funktion sollte nicht geändert werden.

## Callback-Funktion "DelayCreationOfSimulatedClient"

Kann für die Koordination verwendet werden, wenn jeder simulierte Client auf Basis der angegebenen simulierten Clientnummer und der Anzahl der simulierten Clients erstellt wird.

### Syntax

```
_MLREPLAY_EXPORT bool _MLREPLAY_CDECL DelayCreationOfSimulatedClient(  
    asa_uint32 simulatedClientNum,  
    const IMLReplayAPICallbacks * mlrAPICallbacks  
)
```

### Parameter

- **simulatedClientNum** Die simulierte Clientnummer (Ordinalzahl 1), mit der dieser simulierte Client von anderen simulierten Clients in derselben mlreplay-Instanz unterschieden wird.
- **mlrAPICallbacks** Callback-Funktionen zum Bereitstellen von Informationen aus mlreplay, die verwendet werden können, um das Wiederverhalten anzupassen.

### Rückgabe

TRUE, wenn der angegebene simulierte Client erstellt werden soll, oder FALSE, wenn der angegebene simulierte Client nicht erstellt wird.

### Bemerkungen

Simulierter Client *X* wird unter den folgenden Bedingungen erstellt:

- DelayCreationOfSimulatedClient wurde für simulierte Clients 1, ..., *X* - 1 zurückgegeben.
- DelayCreationOfSimulatedClient( *X*, *mlrAPICallbacks* ) gibt TRUE zurück.

Der simulierte Client wird nicht erstellt, wenn diese Callback-Funktion FALSE zurückgibt und noch immer weitere simulierte Clients erstellt werden. Diese Callback-Funktion wird einmal pro simuliertem Client pro mlreplay-Instanz aufgerufen.

## Callback-Funktion "DelayDestructionOfSimulatedClient"

Kann verwendet werden, um zu koordinieren, wann die einzelnen simulierten Clients vernichtet werden, und wird einmal pro simuliertem Client pro mlreplay-Instanz aufgerufen.

### Syntax

```
_MLREPLAY_EXPORT bool _MLREPLAY_CDECL DelayDestructionOfSimulatedClient(  
    asa_uint32 simulatedClientNum,  
    const IMLReplayAPICallbacks * mlrAPICallbacks  
)
```

### Parameter

- **simulatedClientNum** Die simulierte Clientnummer (Ordinalzahl 1), mit der dieser simulierte Client von anderen simulierten Clients in derselben mlreplay-Instanz unterschieden wird.

- **mlrAPICallbacks** Callback-Funktionen zum Bereitstellen von Informationen aus mlreplay, die verwendet werden können, um das Wiedergabeverhalten anzupassen.

### Bemerkungen

Diese Callback-Funktion kann gleichzeitig aufgerufen werden.

Der simulierte Client *X* wird erst vernichtet, wenn DelayDestructionOfSimulatedClient( , *mlrAPICallbacks* ) einen Wert zurückgibt.

## Callback-Funktion "DelayStartOfReplay"

Kann verwendet werden, um zu koordinieren, wann die Wiedergabe beginnt, und wird einmal pro Wiederholung, pro simuliertem Client und pro mlreplay-Instanz aufgerufen.

### Syntax

```
bool DelayStartOfReplay(  
    asa_uint32 repetitionNum  
    uint32 simulatedClientNum,  
    const IMLReplayAPICallbacks * mlrAPICallbacks  
)
```

### Parameter

- **repetitionNum** Die aktuelle Wiederholungsnummer.
- **simulatedClientNum** Die simulierte Clientnummer (Ordinalzahl 1), mit der dieser simulierte Client von anderen simulierten Clients in derselben mlreplay-Instanz unterschieden wird.
- **mlrAPICallbacks** Callback-Funktionen zum Bereitstellen von Informationen aus mlreplay, die verwendet werden können, um das Wiedergabeverhalten anzupassen.

### Rückgabe

TRUE, um die Wiedergabe zu starten, oder FALSE, um die Generierung zu überspringen.

### Bemerkungen

Diese Callback-Funktion kann gleichzeitig aufgerufen werden.

Wenn für Wiederholung *X* FALSE zurückgegeben wird, wird die Wiederholung übersprungen.

## Callback-Funktion "DestroyMLReplayUploadTransaction"

Wird zum Dekonstruieren einer Uploadtransaktion verwendet. Aufrufe erfolgen einmal anfänglich nach dem ersten Aufruf von CreateAndInitMLReplayUploadTransaction sowie danach einmal pro Uploadtransaktion, pro Synchronisation, pro simuliertem Client und pro Wiederholung.

## Syntax

```
_MLREPLAY_EXPORT void _MLREPLAY_CDECL DestroyMLReplayUploadTransaction(
    IMLReplayUploadTransaction * uploadTrans
)
```

## Parameter

- **uploadTrans** Eine Implementierung von IMLReplayUploadTransaction, die das Dienstprogramm mlreplay verwendet, um die Wiedergabesitzung mit benutzerdefinierten Daten zu füllen.

## Bemerkungen

Diese Callback-Funktion sollte nicht geändert werden.

# Callback-Funktion "FiniIdentifySimulatedClient"

Wird verwendet, um den Speicher zu bereinigen, der durch den Aufruf von IdentifySimulatedClient für den entsprechenden simulierten Client genutzt wurde, wird einmal pro simuliertem Client aufgerufen.

## Syntax

```
_MLREPLAY_EXPORT void _MLREPLAY_CDECL FiniIdentifySimulatedClient(
    asa_uint32 simulatedClientNum,
    char * remoteID,
    char * username,
    char * password,
    char * scriptVersion,
    char ** authenticationParameters,
    asa_uint16 numAuthenticationParameters,
    char * ldt,
    const IMLReplayAPICallbacks * mlrAPICallbacks
)
```

## Parameter

- **simulatedClientNum** Die simulierte Clientnummer (Ordinalzahl 1), mit der dieser simulierte Client von anderen simulierten Clients in derselben mlreplay-Instanz unterschieden wird.
- **remoteID** Die entfernte ID, die vom IdentifySimulatedClient-Aufruf für den entsprechenden simulierten Client angegeben wird.
- **username** Der Benutzername, der vom IdentifySimulatedClient-Aufruf für den entsprechenden simulierten Client angegeben wird.
- **password** Das Kennwort, das vom IdentifySimulatedClient-Aufruf für den entsprechenden simulierten Client angegeben wird.
- **scriptVersion** Die Skriptversion, die vom IdentifySimulatedClient-Aufruf für den entsprechenden simulierten Client angegeben wurde.
- **authenticationParameters** Die Authentifizierungsparameter, die vom IdentifySimulatedClient-Aufruf für den entsprechenden simulierten Client angegeben werden.

- **numAuthenticationParameters** Die Anzahl der Authentifizierungsparameter, die vom IdentifySimulatedClient-Aufruf für den entsprechenden simulierten Client angegeben werden.
- **Idt** Die Zeit des letzten Downloads, die vom IdentifySimulatedClient-Aufruf für den entsprechenden simulierten Client angegeben wird.
- **mlrAPICallbacks** Callback-Funktionen zum Bereitstellen von Informationen aus mlreplay, die verwendet werden können, um das Wiedergabeverhalten anzupassen.

### Bemerkungen

Wenn sie generiert werden, enthält diese Callback-Funktion auskommentierten Code, den Sie bei der Verwendung von generischen Benutzernamen, Kennwörtern und entfernten IDs für eine Wiedergabesitzung implementieren können. Der Code gibt den genutzten Speicher frei.

## Callback-Funktion "GetDownloadApplyTime"

Wird verwendet, um langsame Geräte zu simulieren. Aufrufe erfolgen einmal pro Download, pro simuliertem Client und pro Wiederholung.

### Syntax

```
_MLREPLAY_EXPORT asa_uint32 _MLREPLAY_CDECL GetDownloadApplyTime(  
    asa_uint32 repetitionNum,  
    asa_uint32 simulatedClientNum,  
    asa_uint32 recordedSyncNum,  
    asa_uint32 recordedDownloadApplyTime,  
    const IMLReplayAPICallbacks * mlrAPICallbacks  
)
```

### Parameter

- **repetitionNum** Die aktuelle Wiederholungsnummer.
- **simulatedClientNum** Die simulierte Clientnummer (Ordinalzahl 1), mit der dieser simulierte Client von anderen simulierten Clients in derselben mlreplay-Instanz unterschieden wird.
- **syncNum** Die Synchronisationsnummer (Ordinalzahl 1) innerhalb des aufgezeichneten Protokolls.
- **recordedDownloadApplyTime** Die aufgezeichnete Zeit der Downloadübernahme (in Millisekunden).
- **mlrAPICallbacks** Callback-Funktionen zum Bereitstellen von Informationen aus mlreplay, die verwendet werden können, um das Wiedergabeverhalten anzupassen.

### Rückgabe

Die Anzahl an Millisekunden, die für das Übernehmen des Downloads benötigt werden würde.

### Bemerkungen

Diese Callback-Funktion kann gleichzeitig aufgerufen werden.

Das Dienstprogramm `mlreplay` liefert meist eine gute Schätzung der Zeit, die bei Synchronisationen außerhalb einer beständigen Verbindung für die Downloadübernahme benötigt wird. Für Synchronisationen, die in einer dauerhaften Verbindung erfolgen, kann `mlreplay` die für die Downloadübernahme benötigte Zeit nicht mit Genauigkeit schätzen, es sei denn, es werden Downloadbestätigungen verwendet.

## GetMLReplayAPIVersion-Callback-Funktion

Wird verwendet, um die Version der Replay-API zurückzugeben.

### Syntax

```
_MLREPLAY_EXPORT asa_uint32 _MLREPLAY_CDECL GetMLReplayAPIVersion( void )
```

### Bemerkungen

Diese Callback-Funktion wird einmal pro `mlreplay`-Instanz aufgerufen und sollte nicht geändert werden.

## Callback-Funktion "GetUploadTransaction"

Wird verwendet, um die Zeilen anzupassen, die während der Wiedergabesitzung auf den MobiLink-Server hochgeladen werden. Aufrufe erfolgen einmal pro Uploadtransaktion, pro Synchronisation, pro simuliertem Client und pro Wiederholung.

### Hinweis

Das `mlreplay`-Dienstprogramm versucht, die Informationen zum Zeitablauf anzupassen, basierend auf der durch `GetUploadTransaction` angegebenen Größe des neuen Uploads und dem Upload in der aufgezeichneten Protokolldatei. Falls allerdings das Timing wichtig ist, erhalten Sie die besten Ergebnisse, wenn die Gesamtgröße der mit `GetUploadTransaction` hinzugefügten Zeilen etwa mit der Größe der Zeilen übereinstimmt, die bei der ursprünglichen aufgezeichneten Synchronisation hochgeladen wurden. Die einfachste Möglichkeit, um zu gewährleisten, dass die Größe ungefähr gleich ist, besteht darin, eine Synchronisation aufzuzeichnen, bei der dieselbe Anzahl an Zeilen mit ähnlichen Daten hochgeladen wird.

### Syntax

```
_MLREPLAY_EXPORT bool _MLREPLAY_CDECL GetUploadTransaction(  
    asa_uint32 repetitionNum,  
    asa_uint32 simulatedClientNum,  
    asa_uint32 recordedSyncNum,  
    asa_uint32 uploadTransNum,  
    asa_uint32 numUploadedTrans,  
    IMLReplayUploadTransaction * uploadTrans,  
    const IMLReplayAPICallbacks * mlrAPICallbacks  
)
```

### Parameter

- **repetitionNum** Die aktuelle Wiederholungsnummer.

- **simulatedClientNum** Die simulierte Clientnummer (Ordinalzahl 1), mit der dieser simulierte Client von anderen simulierten Clients in derselben mlreplay-Instanz unterschieden wird.
- **syncNum** Die Synchronisationsnummer (Ordinalzahl 1) innerhalb des aufgezeichneten Protokolls.
- **transNum** Die Transaktionsnummer (Ordinalzahl 1) innerhalb der angegebenen Synchronisation.
- **numUploadedTrans** Die Gesamtzahl der Upload-Transaktionen in der angegebenen Synchronisation.
- **uploadTrans** Ein Ausgabeparameter, der mit den Upload-Vorgängen für die aktuelle Transaktion eingestellt werden muss.
- **mlrAPICallbacks** Callback-Funktionen zum Bereitstellen von Informationen aus mlreplay, die verwendet werden können, um das Wiedergabeverhalten anzupassen.

### Rückgabe

Gibt TRUE bei Erfolg und FALSE bei Fehler zurück. Wenn GetUploadTransaction vor der ersten Wiederholung fehlschlägt, wird die Wiedergabesitzung abgebrochen. Wenn GetUploadTransaction vor einer anderen Wiederholung als der ersten fehlschlägt, wird nur der betreffende simulierte Client beendet.

### Bemerkungen

Diese Callback-Funktion kann mehrere Male aufgerufen werden, basierend auf der Anzahl der Synchronisations- und Uploadtransaktionen, die erscheinen, wenn die aufgezeichnete Protokolldatei wiedergegeben wird.

Diese Callback-Funktion kann gleichzeitig aufgerufen werden, aber gleichzeitige Aufrufe haben keinen Zeiger auf dasselbe uploadTrans-Objekt.

## Callback-Funktion "GlobalFini"

Wird verwendet, um globale Variablen zu bereinigen, die von den anderen Callback-Funktionen benutzt wurden, und einmal pro mlreplay-Instanz aufgerufen.

### Syntax

```
_MLREPLAY_EXPORT void _MLREPLAY_CDECL GlobalFini(  
    const IMLReplayAPICallbacks * mlrAPICallbacks  
)
```

### Parameter

- **mlrAPICallbacks** Callback-Funktionen zum Bereitstellen von Informationen aus mlreplay, die verwendet werden können, um das Wiedergabeverhalten anzupassen.

## Callback-Funktion "GlobalInit"

Wird verwendet, um globale Variablen zu initialisieren, die von den anderen Callback-Funktionen benutzt wurden, und einmal pro mlreplay-Instanz aufgerufen.

## Syntax

```
_MLREPLAY_EXPORT bool _MLREPLAY_CDECL GlobalInit(  
    const IMLReplayAPICallbacks * mlrAPICallbacks  
)
```

## Parameter

- **mlrAPICallbacks** Callback-Funktionen zum Bereitstellen von Informationen aus mlreplay, die verwendet werden können, um das Wiedergabeverhalten anzupassen.

## Rückgabe

TRUE bei Erfolg, FALSE bei Fehler. Letzteres führt zum Abbruch der Wiedergabesitzung.

# Callback-Funktion "IdentifySimulatedClient"

Wird verwendet, um Informationen zum simulierten Client anzugeben, und einmal pro simuliertem Client aufgerufen.

## Syntax

```
_MLREPLAY_EXPORT bool _MLREPLAY_CDECL IdentifySimulatedClient(  
    asa_uint32 simulatedClientNum,  
    char ** remoteID,  
    char ** username,  
    char ** password,  
    char ** scriptVersion,  
    char *** authenticationParameters,  
    asa_uint16 * numAuthenticationParameters,  
    char ** ldt,  
    const IMLReplayAPICallbacks * mlrAPICallbacks  
)
```

## Parameter

- **simulatedClientNum** Die simulierte Clientnummer (Ordinalzahl 1), mit der dieser simulierte Client von anderen simulierten Clients in derselben mlreplay-Instanz unterschieden wird.
- **remoteID** Ein Ausgabeparameter, der auf die entfernte ID des simulierten Clients eingestellt werden und ein unter allen mlreplay-Instanzen eindeutiger Wert sein muss.
- **username** Ein Ausgabeparameter, der auf den MobiLink-Benutzernamen für diesen simulierten Client eingestellt werden muss.
- **password** Ein Ausgabeparameter, der auf das Kennwort für den MobiLink Benutzer eingestellt werden muss.
- **scriptVersion** Ein Ausgabeparameter, der auf die vom MobiLink-Benutzer zu verwendende Skriptversion gesetzt werden muss.
- **authenticationParameters** Ein Ausgabeparameter, der auf ein Array von Authentifizierungsparametern für diesen simulierten Client eingestellt werden muss.

- **numAuthenticationParameters** Ein Ausgabeparameter, der auf die Anzahl der in authenticationParameters zurückgegebenen Authentifizierungsparameter eingestellt wurde.
- **ldt** Ein Ausgabeparameter, der auf die Zeit des letzten Downloads für den Benutzer eingestellt wurde. Die LDT muss das Format JJJJ-MM TT hh:mm:ss.SSS haben.
- **mlrAPICallbacks** Callback-Funktionen zum Bereitstellen von Informationen aus mlreplay, die verwendet werden können, um das Wiedergabeverhalten anzupassen.

### Rückgabe

TRUE bei Erfolg, FALSE bei Fehler. Letzteres führt zum Abbruch der Wiedergabesitzung.

### Bemerkungen

Wenn username, password, authenticationParameters, scriptVersion oder ldt NULL ist, verwendet mlreplay die entsprechenden Werte aus dem erfassten Protokoll. Wenn remoteID NULL ist, ersetzt mlreplay die entfernte ID durch eine GUID für den simulierten Client.

Wenn sie generiert wurde, enthält diese Callback-Funktion auskommentierten Code, den Sie bei der Verwendung von generischen Benutzernamen, Kennwörtern und entfernten IDs für eine Wiedergabesitzung implementieren können. Der Code erstellt Benutzername, Kennwort und entfernte ID für **user\_simulierte Clientnummer**, **pwd\_simulierte Clientnummer** bzw. **rid\_simulierte Clientnummer**.

## Callback-Funktion "ReportEndOfReplay"

Wird verwendet, um beliebige erforderliche Aktionen durchzuführen, wenn ein simulierter Client die Wiedergabe abgeschlossen hat. Aufrufe erfolgen einmal pro simuliertem Client und pro Wiederholung.

### Syntax

```
_MLREPLAY_EXPORT bool _MLREPLAY_CDECL ReportEndOfReplay(  
    asa_uint32 repetitionNum,  
    asa_uint32 simulatedClientNum,  
    bool success,  
    const IMLReplayAPICallbacks * mlrAPICallbacks  
)
```

### Parameter

- **repetitionNum** Die aktuelle Wiederholungsnummer.
- **simulatedClientNum** Die simulierte Clientnummer (Ordinalzahl 1), mit der dieser simulierte Client von anderen simulierten Clients in derselben mlreplay-Instanz unterschieden wird.
- **success** TRUE, wenn der simulierte Client die Wiedergabe erfolgreich abgeschlossen hat, sonst FALSE.
- **mlrAPICallbacks** Callback-Funktionen zum Bereitstellen von Informationen aus mlreplay, die verwendet werden können, um das Wiedergabeverhalten anzupassen.

## Rückgabe

TRUE bei Erfolg, FALSE bei Fehler.

## Bemerkungen

Mithilfe dieser Callback-Funktion können Sie sicherstellen, dass die Daten in Wiederholung *X* korrekt hochgeladen wurden.

Diese Callback-Funktion bestimmt das Ergebnis der angegebenen Wiedergabe. Wenn der Rückgabewert FALSE Erfolg bedeutet, hat diese Callback-Funktion keine Auswirkungen auf den Erfolg der Wiedergabe. Wenn der Rückgabewert TRUE Erfolg bedeutet, aber diese Callback-Funktion FALSE zurückgibt, behandelt mlreplay den simulierten Client, als ob er für die angegebene Wiederholung fehlgeschlagen wäre. Diese Callback-Funktion kann gleichzeitig aufgerufen werden.

# Systemprozeduren des MobiLink-Servers

MobiLink verfügt über folgende gespeicherte Prozeduren, die Sie bei der Erstellung Ihrer Anwendungen unterstützen.

## Systemprozeduren zum Hinzufügen oder Löschen von Skripten

Um Synchronisationsskripten verwenden zu können, müssen Sie sie in der konsolidierten Datenbank in die MobiLink-Systemtabellen einfügen. Die folgenden Systemprozeduren fügen der konsolidierten Datenbank Synchronisationsskripten hinzu oder löschen diese aus ihr.

- „ml\_add\_connection\_script-Systemprozedur“
- „ml\_add\_table\_script-Systemprozedur“
- „ml\_add\_dnet\_connection\_script-Systemprozedur“
- „ml\_add\_dnet\_table\_script-Systemprozedur“
- „ml\_add\_java\_connection\_script-Systemprozedur“
- „ml\_add\_java\_table\_script-Systemprozedur“

Wenn Sie die API für Java des MobiLink-Servers oder .NET verwenden, registrieren Sie mit diesen gespeicherten Prozeduren eine Methode als das Skript für ein Ereignis, sodass die Methode ausgeführt wird, wenn das Ereignis stattfindet. Sie können damit auch die Registrierung Ihrer Methoden aufheben.

Wenn Sie ein Skript mithilfe einer Systemprozedur hinzufügen, ist das Skript eine Zeichenfolge. Alle Zeichenfolgen innerhalb des Skripts müssen entwertet werden. In SQL Anywhere muss jedes Anführungszeichen (') doppelt angegeben werden, um die Zeichenfolge nicht abubrechen.

Es ist nicht möglich, Skripten mit einer Länge von mehr als 255 Byte mithilfe von Systemprozeduren in Adaptive Server Enterprise 11.5 oder früher einzufügen. Für längere Skripten verwenden Sie stattdessen Sybase Central oder fügen sie direkt ein.

IBM DB2 LUW vor Version 6 unterstützt nur Spaltennamen und andere Bezeichner mit maximal 18 Zeichen. Längere Namen werden gekürzt. So wird beispielsweise ml\_add\_connection\_script zu ml\_add\_connection\_ gekürzt.

## Systemprozeduren für die Verwaltung von entfernten Aufgaben

Die folgenden gespeicherten Prozeduren können für die Verwaltung von entfernten Aufgaben verwendet werden:

- „ml\_ra\_add\_agent\_id-Systemprozedur“
- „ml\_ra\_assign\_task-Systemprozedur“
- „ml\_ra\_cancel\_notification-Systemprozedur“
- „ml\_ra\_cancel\_task\_instance-Systemprozedur“
- „ml\_ra\_clone\_agent\_properties-Systemprozedur“
- „ml\_ra\_delete\_agent\_id-Systemprozedur“
- „ml\_ra\_delete\_events\_before-Systemprozedur“
- „ml\_ra\_delete\_remote\_id-Systemprozedur“
- „ml\_ra\_delete\_task-Systemprozedur“
- „ml\_ra\_get\_agent\_events-Systemprozedur“
- „ml\_ra\_get\_agent\_ids-Systemprozedur“
- „ml\_ra\_get\_agent\_properties-Systemprozedur“
- „ml\_ra\_get\_latest\_event\_id-Systemprozedur“
- „ml\_ra\_get\_orphan\_taskdbs-Systemprozedur“
- „ml\_ra\_reassign\_taskdb-Systemprozedur“
- „ml\_ra\_get\_remote\_ids-Systemprozedur“
- „ml\_ra\_get\_task\_results-Systemprozedur“
- „ml\_ra\_get\_task\_status-Systemprozedur“
- „ml\_ra\_manage\_remote\_db-Systemprozedur“
- „ml\_ra\_notify\_agent\_sync-Systemprozedur“
- „ml\_ra\_set\_agent\_property-Systemprozedur“
- „ml\_ra\_unmanage\_remote\_db-Systemprozedur“

## LDAP-Systemprozeduren

Die folgenden gespeicherten Prozeduren können verwendet werden, um die LDAP-Authentifizierung einzurichten und zu verwalten:

- „ml\_add\_certificates\_file-Systemprozedur“ auf Seite 650
- „ml\_add\_ldap\_server-Systemprozedur“ auf Seite 658
- „ml\_add\_user\_auth\_policy-Systemprozedur“ auf Seite 670

## Systemprozeduren für das Synchronisationsmodell

Die folgenden gespeicherten Prozeduren können verwendet werden, um Schema-Upgrades zu verwalten:

- „ml\_model\_drop-Systemprozedur“ auf Seite 675
- „ml\_model\_check\_all\_schema-Systemprozedur“ auf Seite 676
- „ml\_model\_check\_version\_schema-Systemprozedur“ auf Seite 677

## Andere Systemprozeduren

- „ml\_add\_property-Systemprozedur“
- „ml\_delete\_sync\_state\_before-Systemprozedur“
- „ml\_reset\_sync\_state-Systemprozedur“

# ml\_add\_certificates\_file-Systemprozedur

Richten Sie vertrauenswürdige Zertifikate ein, wenn Sie TLS mit LDAP-Authentifizierung verwenden.

**Syntax**

```
ml_add_certificates_file (  
  'file_name',  
)
```

**Parameter**

Syntax	Beschreibung
file_name	VARCHAR(1024). Der Name der Datei mit den vertrauenswürdigen Zertifikaten.

**Bemerkungen**

Diese Prozedur füllt die ml\_trusted\_certificates\_file-Tabelle mit Daten zur angegebenen Datei für das vertrauenswürdige Zertifikat.

Vorhandene Einträge in der ml\_trusted\_certificates\_file-Tabelle werden gelöscht, bevor der Name der neuen Datei für das vertrauenswürdige Zertifikat eingefügt wird, weil für eine Serverfarm nur eine einzelne Datei für das vertrauenswürdige Zertifikat benötigt wird.

**Siehe auch**

- „ml\_add\_ldap\_server-Systemprozedur“ auf Seite 658
- „ml\_add\_user\_auth\_policy-Systemprozedur“ auf Seite 670

# ml\_add\_column-Systemprozedur (nicht mehr empfohlen)

Registriert Informationen über Spalten in entfernten Datenbanken für die Verwendung durch benannte Spaltenparameter.

**Syntax**

```
ml_add_column (  
  'version',  
  'table',  
  'column',  
  'type'  
)
```

**Parameter**

Syntax	Beschreibung
version	VARCHAR(128). Versionsname
table	VARCHAR(128). Der Tabellename.

Syntax	Beschreibung
column	VARCHAR(128). Der Spaltenname.
type	VARCHAR(128). Reserviert für spätere Verwendung. Auf Null setzen.

### Bemerkungen

Diese Prozedur füllt die MobiLink-Systemtabelle ml\_column mit Informationen über die Spalten in der entfernten Datenbank. Diese Informationen werden von benannten Zeilenparametern verwendet.

#### Vorsicht

Aufrufe von ml\_add\_column müssen in derselben Reihenfolge ausgeführt werden, in der die Spalten in der entfernten Datenbanktabelle enthalten sind. Andernfalls sind die resultierenden Daten möglicherweise falsch.

Sie müssen diese Systemprozedur ausführen, wenn Ihre Synchronisationsclients **keine** Spaltennamen senden. Standardmäßig **senden** Clients ab Version 12 Spaltennamen, sodass ml\_add\_column in den meisten Deployments nicht erforderlich ist. Die ml\_add\_column-Namen überschreiben immer Namen vom Client.

Sie können alle Einträge für den Tabellennamen in einer bestimmten Skriptversion löschen, indem Sie den Spaltennamen auf NULL setzen.

### Siehe auch

- „Skriptparameter“ auf Seite 236

### Beispiele

Der Aufruf der folgenden gespeicherten Prozedur füllt die MobiLink-Systemtabelle ml\_column für Spalte col1 in MyTable für die Skriptversion Version1. Dieser Aufruf ermöglicht Ihnen die Verwendung des benannten Zeilenparameter r.col1 und o.col1 in Tabellenskripten für MyTable1 in der Skriptversion Version1, wenn der Synchronisationsclient keine Spaltennamen sendet. (Bei Clients vor Version 12 ist dies standardmäßig der Fall.)

```
CALL ml_add_column( 'Version1', 'MyTable1', 'col1', NULL )
```

Der Aufruf der folgenden gespeicherten Prozedur löscht in Skriptversion Version1 alle Einträge für MyTable1 aus der MobiLink-Systemtabelle ml\_column:

```
CALL ml_add_column( 'Version1', 'MyTable1', NULL, NULL )
```

## ml\_add\_connection\_script-Systemprozedur

Mit dieser Systemprozedur können Sie der konsolidierten Datenbank SQL-Verbindungsskripten hinzufügen.

**Syntax**

```
ml_add_connection_script (  
  'version',  
  'event',  
  'script'  
)
```

**Parameter**

Syntax	Beschreibung
version	VARCHAR(128). Versionsname
event	VARCHAR(128). Der Ereignisname
script	TEXT. Skriptinhalt. Für Adaptive Server Enterprise hat dieser Parameter den Datentyp VARCHAR(16384). Für IBM DB2 LUW hat dieser Parameter den Datentyp VARCHAR(4000). Bei SAP HANA und Oracle ist dieser Parameter vom Typ CLOB.

**Bemerkungen**

Um ein Verbindungsskript zu löschen, setzen Sie den Skriptinhaltparameter auf NULL.

Wenn Sie ein Skript hinzufügen, wird das Skript in die Tabelle ml\_script eingefügt, und die entsprechenden Referenzen werden definiert, um dem Skript das festgelegte Ereignis und die betreffende Skriptenversion zuzuordnen. Wenn der Versionsname neu ist, wird er automatisch in die Tabelle ml\_version eingefügt.

**Siehe auch**

- [Systemprozeduren zum Hinzufügen oder Löschen von Skripten auf Seite 648](#)
- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)
- [„ml\\_add\\_table\\_script-Systemprozedur“ auf Seite 668](#)
- [„ml\\_add\\_dnet\\_connection\\_script-Systemprozedur“ auf Seite 653](#)
- [„ml\\_add\\_dnet\\_table\\_script-Systemprozedur“ auf Seite 654](#)
- [„ml\\_add\\_java\\_connection\\_script-Systemprozedur“ auf Seite 655](#)
- [„ml\\_add\\_java\\_table\\_script-Systemprozedur“ auf Seite 656](#)

**Beispiel**

Folgende Anweisung fügt ein Verbindungsskript, das dem Ereignis begin\_synchronization zugeordnet ist, der Skriptversion custdb in einer konsolidierten SQL Anywhere-Datenbank hinzu. Das Skript selbst ist eine einzelne Anweisung, mit der die Variable @EmployeeID belegt wird.

```
call ml_add_connection_script( 'custdb',  
  'begin_synchronization',  
  'set @EmployeeID = {ml s.username}' )
```

## ml\_add\_dnet\_connection\_script-Systemprozedur

Mit dieser Systemprozedur registrieren Sie eine .NET-Methode als Skript für ein Verbindungsereignis bzw. heben die Registrierung wieder auf.

### Syntax

```
ml_add_dnet_connection_script (  
    'version',  
    'event',  
    'script'  
)
```

### Parameter

Syntax	Beschreibung
version	VARCHAR(128). Versionsname
event	VARCHAR(128). Der Ereignisname
script	TEXT. Skriptinhalt. Für Adaptive Server Enterprise hat dieser Parameter den Datentyp VARCHAR(16384). Für IBM DB2 LUW hat dieser Parameter den Datentyp VARCHAR(4000). Bei SAP HANA und Oracle ist dieser Parameter vom Typ CLOB.

### Bemerkungen

Um die Registrierung einer Methode aufzuheben, setzen Sie den Skriptinhaltsparameter auf NULL.

Der Wert "script" ist eine öffentliche Methode in einer Klasse in einer .NET-Assembly (z.B. MeineKlasse.MeineMethode).

Wenn Sie ml\_add\_dnet\_connection\_script aufrufen, wird die Methode mit dem Ereignis und der Skriptversion verknüpft, die Sie angeben. Wenn der Versionsname neu ist, wird er automatisch in die Tabelle ml\_version eingefügt.

### Siehe auch

- [Systemprozeduren zum Hinzufügen oder Löschen von Skripten auf Seite 648](#)
- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)
- [„ml\\_add\\_dnet\\_table\\_script-Systemprozedur“ auf Seite 654](#)
- [„ml\\_add\\_connection\\_script-Systemprozedur“ auf Seite 651](#)
- [„ml\\_add\\_table\\_script-Systemprozedur“ auf Seite 668](#)
- [„ml\\_add\\_java\\_table\\_script-Systemprozedur“ auf Seite 656](#)
- [„Methoden“ auf Seite 542](#)
- [„Synchronisationsskripten in .NET“ auf Seite 536](#)

### Beispiel

Das folgende Beispiel registriert die Methode beginDownloadConnection der Klasse ExampleClass für das Ereignis begin\_download.

```
call ml_add_dnet_connection_script( 'ver1',  
  'begin_download',  
  'ExamplePackage.ExampleClass.beginDownloadConnection' );
```

# ml\_add\_dnet\_table\_script-Systemprozedur

Mit dieser Systemprozedur registrieren Sie eine .NET-Methode als Skript für ein Tabellenereignis bzw. heben die Registrierung wieder auf.

## Syntax

```
ml_add_dnet_table_script (  
  'version',  
  'table',  
  'event',  
  'script'  
)
```

## Parameter

Syntax	Beschreibung
version	VARCHAR(128). Versionsname
table	VARCHAR(128). Der Tabellename.
event	VARCHAR(128). Der Ereignisname
script	TEXT. Der Skriptinhalt. Für Adaptive Server Enterprise hat dieser Parameter den Datentyp VARCHAR(16384). Für IBM DB2 LUW hat dieser Parameter den Datentyp VARCHAR(4000). Bei SAP HANA und Oracle ist dieser Parameter vom Typ CLOB.

## Bemerkungen

Um die Registrierung einer Methode aufzuheben, setzen Sie den Skriptinhaltparameter auf NULL.

Der Skriptwert ist eine öffentliche Methode in einer Klasse in einer .NET-Assembly (z.B. MeineKlasse.MeineMethode).

Wenn Sie ml\_add\_dnet\_table\_script aufrufen, wird die Methode mit der Tabelle, dem Ereignis und der Skriptversion verknüpft, die Sie angeben. Wenn der Versionsname neu ist, wird er automatisch in die Tabelle ml\_version eingefügt.

**Siehe auch**

- [Systemprozeduren zum Hinzufügen oder Löschen von Skripten auf Seite 648](#)
- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)
- [„ml\\_add\\_dnet\\_connection\\_script-Systemprozedur“ auf Seite 653](#)
- [„ml\\_add\\_connection\\_script-Systemprozedur“ auf Seite 651](#)
- [„ml\\_add\\_table\\_script-Systemprozedur“ auf Seite 668](#)
- [„ml\\_add\\_java\\_connection\\_script-Systemprozedur“ auf Seite 655](#)
- [„Methoden“ auf Seite 542](#)
- [„Synchronisationsskripten in .NET“ auf Seite 536](#)

**Beispiel**

Das folgende Beispiel weist die Methode empDownloadCursor der Klasse CustEmpScripts dem Ereignis download\_cursor für die Tabelle emp zu.

```
call ml_add_dnet_table_script( 'ver1', 'emp',
                             'download_cursor', 'EgPackage.EgClass.empDownloadCursor' )
```

## ml\_add\_java\_connection\_script-Systemprozedur

Mit dieser Systemprozedur registrieren Sie eine Java-Methode als Skript für ein Verbindungsereignis bzw. heben die Registrierung wieder auf.

**Syntax**

```
ml_add_java_connection_script (
  'version',
  'event',
  'script'
)
```

**Parameter**

Syntax	Beschreibung
version	VARCHAR(128). Versionsname
event	VARCHAR(128). Der Ereignisname
script	TEXT. Der Skriptinhalt. Für Adaptive Server Enterprise hat dieser Parameter den Datentyp VARCHAR(16384). Für IBM DB2 LUW hat dieser Parameter den Datentyp VARCHAR(4000). Bei SAP HANA und Oracle ist dieser Parameter vom Typ CLOB.

**Bemerkungen**

Um die Registrierung einer Methode aufzuheben, setzen Sie den Skriptinhaltparameter auf NULL.

Der Wert "script" ist eine öffentliche Methode in einer Klasse im Klassenpfad des MobiLink-Servers (z.B. MeineKlasse.MeineMethode).

Wenn Sie `ml_add_java_connection_script` aufrufen, wird die Methode mit dem Ereignis und der Skriptversion verknüpft, die Sie angeben. Wenn der Versionsname neu ist, wird er automatisch in die Tabelle `ml_version` eingefügt.

**Siehe auch**

- [Systemprozeduren zum Hinzufügen oder Löschen von Skripten auf Seite 648](#)
- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)
- [„ml\\_add\\_connection\\_script-Systemprozedur“ auf Seite 651](#)
- [„ml\\_add\\_table\\_script-Systemprozedur“ auf Seite 668](#)
- [„ml\\_add\\_dnet\\_connection\\_script-Systemprozedur“ auf Seite 653](#)
- [„ml\\_add\\_dnet\\_table\\_script-Systemprozedur“ auf Seite 654](#)
- [„ml\\_add\\_java\\_table\\_script-Systemprozedur“ auf Seite 656](#)
- [„Methoden“ auf Seite 468](#)
- [„Schreiben eines Synchronisationsskripts in Java“ auf Seite 463](#)

**Beispiel**

Das folgende Beispiel registriert die Methode `endConnection` der Klasse `CustEmpScripts` für das Ereignis `end_connection`

```
call ml_add_java_connection_script( 'ver1',  
  'end_connection',  
  'CustEmpScripts.endConnection' )
```

# **ml\_add\_java\_table\_script-Systemprozedur**

Mit dieser Systemprozedur registrieren Sie eine Java-Methode als Skript für ein Tabellenereignis bzw. heben die Registrierung wieder auf.

**Syntax**

```
ml_add_java_table_script (  
  'version',  
  'table',  
  'event',  
  'script'  
)
```

**Parameter**

Syntax	Beschreibung
version	VARCHAR(128). Versionsname
table	VARCHAR(128). Tabellenname
event	VARCHAR(128). Der Ereignisname

Syntax	Beschreibung
script	TEXT. Skriptinhalt Für Adaptive Server Enterprise hat dieser Parameter den Datentyp VARCHAR(16384). Für IBM DB2 LUW hat dieser Parameter den Datentyp VARCHAR(4000). Bei SAP HANA und Oracle ist dieser Parameter vom Typ CLOB.

### Bemerkungen

Um die Registrierung einer Methode aufzuheben, setzen Sie den Skriptinhaltparameter auf NULL.

Der Wert *script* ist eine öffentliche Methode in einer Klasse im Klassenpfad des MobiLink-Servers (z.B. MeineKlasse.MeineMethode).

Wenn Sie `ml_add_java_table_script` aufrufen, wird die Methode mit der Tabelle, dem Ereignis und der Skriptversion verknüpft, die Sie angeben. Wenn der Versionsname neu ist, wird er automatisch in die Tabelle `ml_version` eingefügt.

### Siehe auch

- [Systemprozeduren zum Hinzufügen oder Löschen von Skripten auf Seite 648](#)
- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)
- [„ml\\_add\\_connection\\_script-Systemprozedur“ auf Seite 651](#)
- [„ml\\_add\\_table\\_script-Systemprozedur“ auf Seite 668](#)
- [„ml\\_add\\_dnet\\_connection\\_script-Systemprozedur“ auf Seite 653](#)
- [„ml\\_add\\_dnet\\_table\\_script-Systemprozedur“ auf Seite 654](#)
- [„ml\\_add\\_java\\_connection\\_script-Systemprozedur“ auf Seite 655](#)
- [„Methoden“ auf Seite 468](#)
- [„Schreiben eines Synchronisationsskripts in Java“ auf Seite 463](#)

### Beispiel

Das folgende Beispiel registriert die Methode `empDownloadCursor` der Klasse `CustEmpScripts` für das Ereignis `download_cursor` der Tabelle `emp`.

```
call ml_add_java_table_script( 'ver1', 'emp',
'download_cursor', 'CustEmpScripts.empDownloadCursor' )
```

## ml\_add\_lang\_connection\_script-Systemprozedur

Diese Prozedur ist nur für den internen Gebrauch gedacht.

### Siehe auch

- [„ml\\_add\\_connection\\_script-Systemprozedur“ auf Seite 651](#)
- [„ml\\_add\\_dnet\\_connection\\_script-Systemprozedur“ auf Seite 653](#)
- [„ml\\_add\\_java\\_connection\\_script-Systemprozedur“ auf Seite 655](#)

## ml\_add\_lang\_connection\_script\_chk-Systemprozedur

Diese Prozedur ist nur für den internen Gebrauch gedacht.

### Siehe auch

- „ml\_add\_connection\_script-Systemprozedur“ auf Seite 651
- „ml\_add\_dnet\_connection\_script-Systemprozedur“ auf Seite 653
- „ml\_add\_java\_connection\_script-Systemprozedur“ auf Seite 655

## ml\_add\_lang\_table\_script-Systemprozedur

Diese Prozedur ist nur für den internen Gebrauch gedacht.

### Siehe auch

- „ml\_add\_table\_script-Systemprozedur“ auf Seite 668
- „ml\_add\_java\_table\_script-Systemprozedur“ auf Seite 656
- „ml\_add\_dnet\_table\_script-Systemprozedur“ auf Seite 654

## ml\_add\_lang\_table\_script\_chk-Systemprozedur

Diese Prozedur ist nur für den internen Gebrauch gedacht.

### Siehe auch

- „ml\_add\_table\_script-Systemprozedur“ auf Seite 668
- „ml\_add\_java\_table\_script-Systemprozedur“ auf Seite 656
- „ml\_add\_dnet\_table\_script-Systemprozedur“ auf Seite 654

## ml\_add\_ldap\_server-Systemprozedur

Erstellt, löscht oder aktualisiert LDAP-Server.

### Syntax

```
ml_add_ldap_server (  
  'ldsrv_name',  
  'search_url',  
  'access_dn',  
  'access_dn_pwd',  
  'auth_url',  
  'conn_retries',  
  'conn_timeout',  
  'use_tls'  
)
```

**Parameter**

Syntax	Beschreibung
ldsrv_name	VARCHAR(128). Ein eindeutiger LDAP-Servername.
search_url	VARCHAR(1024). Eine URL-Zeichenfolge, die Name oder IP-Adresse, Portnummer und Suchzeichenfolge für die DN-Suche (Distinguished Name) nach einer angegebenen Benutzer-ID verwendet, um den Host zu identifizieren.
access_dn	VARCHAR(1024). Der Distinguished Name für eine LDAP-Benutzer, der vom MobiLink-Server verwendet wird, um eine Verbindung mit dem LDAP-Server herstellen. Der LDAP-Benutzer muss die Berechtigung haben, auf dem LDAP-Server nach DN's zu suchen.
access_dn_pwd	VARCHAR(1024). Das Kennwort, das dem im access_dn-Parameter angegebenen DN zugeordnet ist.
auth_url	VARCHAR(1024). Eine URL-Zeichenfolge, die Name oder IP-Adresse sowie die Portnummer des für die Benutzerauthentifizierung eingesetzten LDAP-Servers verwendet, um den Host zu identifizieren.
conn_retries	TINYINT. Angabe, wie oft der MobiLink-Server versucht, für DN-Suchvorgänge und Authentifizierung eine Verbindung mit dem LDAP-Server herzustellen. Die gültige Bereich ist 1-60. Der Standardwert ist 3.
conn_timeout	TINYINT. Der Verbindungs-Timeout für Verbindungen vom MobiLink-Server zum LDAP-Server für DN-Suchen und Authentifizierung. Der Wert wird in Sekunden angegeben. Der Standardwert beträgt 10 Sekunden.
start_tls	TINYINT. Legt fest, dass TLS verwendet werden kann, um für DN-Suchvorgänge und Authentifizierung eine Verbindungen mit dem LDAP-Server herzustellen.

**Bemerkungen**

Diese Prozedur füllt die ml\_ldap\_server-Tabelle mit Informationen zum angegebenen LDAP-Server.

**Siehe auch**

- „ml\_add\_certificates\_file-Systemprozedur“ auf Seite 650
- „ml\_add\_user\_auth\_policy-Systemprozedur“ auf Seite 670

**Beispiel**

Im folgenden Beispiel wird ein LDAP-Server namens **my\_primary** zur ml\_ldap\_server-Tabelle hinzugefügt.

```
CALL ml_add_ldap_server(
  'my_primary',                                //server name
  'ldap://voyager:389/dc=MyCompany,dc=com??sub?cn=*', //search URL
  'cn=aseadmin, cn=Users, dc=mycompany, dc=com',      //access DN
  'Secret99Password',                               //access DN password
  'ldap://voyager:389/',                           //authentication URL
```

```
10,                                     //connection retries
5,                                     //connection timeout
0                                     //no TLS
)
```

## ml\_add\_missing\_dnld\_scripts-Systemprozedur

Verwenden Sie diese Systemprozedur, um fehlende download\_cursor- und download\_delete\_cursor-Skripten als ignorierte Skripten zu definieren.

### Syntax

```
ml_add_missing_dnld_scripts (
'script_version_name')
```

### Parameter

Syntax	Beschreibung
script_version_name	VARCHAR(128). Der Name der Skriptversion.

### Siehe auch

- [„download\\_cursor \(Tabellenereignis\)“](#) auf Seite 329
- [„download\\_delete\\_cursor \(Tabellenereignis\)“](#) auf Seite 331

## ml\_add\_passthrough-Systemprozedur

Mit dieser Systemprozedur legen Sie entfernte Datenbanken fest, die ein Skript ausführen sollen. Diese Prozedur fügt der Systemtabelle ml\_passthrough einen Eintrag hinzu. Wenn ein Eintrag mit den angegebenen Werten für "remote\_id" und "run\_order" bereits in der Tabelle enthalten ist, aktualisiert diese Prozedur den Eintrag.

### Syntax

```
ml_add_passthrough (
'remote_id',
'script_name',
run_order
)
```

**Parameter**

Syntax	Beschreibung
remote_id	<p>VARCHAR(128). Die entfernte ID der Datenbank, die das Skript ausführen soll. Dieser Wert kann eine gültige entfernte ID in der Tabelle ml_database sein, um das Skript auf einen bestimmten Client anzuwenden, oder NULL, um es auf alle in der Tabelle ml_database aufgelisteten Skriptclients anzuwenden.</p> <div> <p><b>Vorsicht</b></p> <p>Gehen Sie sehr vorsichtig vor, wenn Sie ein Skript auf alle oder viele entfernte Datenbanken anwenden. Ein schlecht geschriebenes Skript kann die meisten oder alle entfernten Datenbanken beschädigen oder deaktivieren.</p> </div>
script_name	<p>VARCHAR(128). Der Name des Skripts, das subskribiert wird. Dieser Wert muss ein gültiger Skriptname sein, der in der ml_passthrough_script-Tabelle festgelegt ist.</p>
run_order	<p>INTEGER. Der Parameter "run_order" legt die Reihenfolge fest, in der Skripten in der entfernten Datenbank angewendet werden. Skripten werden immer in der von "run_order" festgelegten Reihenfolge ausgeführt. Jede entfernte Datenbank speichert die Ausführungsreihenfolge des letzten Skripts, dessen Anwendung versucht wurde, lädt keine Skripten mit einer niedrigeren Ausführungsreihenfolge herunter und führt diese auch nicht aus.</p> <p>Dieser Wert muss ein nicht-negativer Ganzzahlwert oder NULL sein.</p>

**Bemerkungen**

Wenn Sie "run\_order" als NULL festlegen, weist die Prozedur eine Ganzzahl basierend auf dem Wert von "remote\_id" zu. Wenn "remote\_id" den Wert NULL hat, weist die Prozedur einen Wert zu, der dem Wert von "run\_order" in ml\_passthrough plus 10 entspricht. Wenn "remote\_id" nicht NULL ist, weist die Prozedur den Höchstwert der Ausführungsreihenfolge-Spalte für die entfernte ID in der Tabelle ml\_passthrough plus 10 zu.

**ml\_add\_passthrough\_repair-Systemprozedur**

Mit dieser Systemprozedur legen Sie Regeln zur Verarbeitung von Skriptfehlern fest. Jede Regel legt die Aktion fest, die ein Client ausführen muss, wenn ein bestimmtes Skript einen bestimmten Fehlercode generiert. Diese Prozedur fügt der Systemtabelle ml\_passthrough\_repair einen Eintrag hinzu. Wenn ein Eintrag mit den betreffenden Werten für "failed\_script\_name" und "error\_code" in der Tabelle bereits vorhanden ist, aktualisiert die Prozedur den Eintrag.

**Syntax**

```
ml_add_passthrough_repair (
'failed_script_name',
error_code,
'new_script_name',
```

```
'action'  
)
```

Parameter

Syntax	Beschreibung
failed_script_name	VARCHAR(128). Der Name des fehlgeschlagenen Skripts, für das diese Regel gilt. Dieser Wert muss ein gültiger Skriptname in der ml_passthrough_script-Tabelle sein.
error_code	INTEGER. Der SQL Anywhere-Fehlercode, der für diese Regel gilt.
new_script_name	VARCHAR(128). Der Name eines Skripts, das das fehlgeschlagene Skript ersetzen soll, wenn die Aktion R ist. Wenn die Aktion S, P oder H ist, muss dieser Wert NULL sein. Wenn die Aktion R ist, muss dieser Wert ein gültiger Skriptname in der Tabelle ml_passthrough_script sein, und der Wert kann mit failed_script_name übereinstimmen.
action	<p>CHAR(1). Die Aktion, die ein Client ausführen muss, wenn für failed_script_name "Fehlercode" generiert wird. Dabei muss es sich um einen der folgenden Werte handeln:</p> <ul style="list-style-type: none"><li>• <b>R</b> (Replace - ersetzen) Das fehlgeschlagene Skript soll durch das von <b>new script name</b> angegebene Skript ersetzt werden und es soll versucht werden, das neue Skript auszuführen. Um das fehlgeschlagene Skript erneut auszuführen, legen Sie <b>new script name</b> auf denselben Wert wie <b>failed script name</b> fest.</li><li>• <b>P</b> (Purge - entfernen) Die entfernte Datenbank soll alle empfangenen Skripten verwerfen und anschließend das Skript normal weiter ausführen.</li><li>• <b>S</b> (Skip - überspringen) Die entfernte Datenbank soll das fehlgeschlagene Skript ignorieren und die Skripten sollen weiter ausgeführt werden, so als wäre das fehlgeschlagene Skript erfolgreich ausgeführt worden.</li><li>• <b>H</b> (Halt - anhalten) Die entfernte Datenbank soll keine weiteren Skripten ausführen, bis sie weitere Anweisungen erhält.</li></ul>

Bemerkungen

Sie sollten den Fehlschlag von SQL Passthrough-Skripten unbedingt vermeiden, indem Sie die Skripten sorgfältig testen.

ml\_add\_passthrough\_script-Systemprozedur

Mit dieser Systemprozedur können Sie ein Passthrough-Skript erstellen. Diese Prozedur fügt der Systemtabelle ml\_passthrough\_script einen Eintrag hinzu.

**Syntax**

```

ml_add_passthrough_script (
  'script_name',
  'flags',
  'affected_pubs',
  'script',
  'description'
)

```

**Parameter**

Syntax	Beschreibung
script_name	VARCHAR(128). Der Skriptname. Dieser Wert muss eindeutig sein.
flags	<p>VARCHAR(256). Der Wert, der Clients anweist, wie das Skript ausgeführt werden soll. Dieser Wert kann NULL sein oder eine Kombination der folgenden Schlüsselwörter in einer durch Semikola getrennten Liste enthalten:</p> <ul style="list-style-type: none"> <li>• <b>manual</b> Das Skript kann nur im manuellen Ausführungsmodus ausgeführt werden. Standardmäßig können alle Skripten im automatischen oder manuellen Ausführungsmodus ausgeführt werden.</li> <li>• <b>Exklusiv</b> Das Skript kann nur dann automatisch am Ende einer Synchronisation ausgeführt werden, wenn für alle synchronisierten Tabellen Exklusivsperrungen vorliegen. Diese Option wird ignoriert, wenn im Betroffene_Publikationen-Wert keine Publikationen aufgelistet werden. Diese Option gilt nur für SQL Anywhere.</li> <li>• <b>schema_diff</b> Das Skript wird im Schemavergleichsmodus ausgeführt. In diesem Modus wird das Datenbankschema geändert, sodass es mit dem im Skript beschriebenen Schema übereinstimmt. Eine CREATE-Anweisung für eine vorhandene Tabelle wird z.B. als ALTER-Anweisung behandelt. Dieser Parameter gilt nur für entfernte UltraLite-Datenbanken.</li> </ul> <p>Beispiel:</p> <pre>'manual;exclusive;schema_diff'</pre>
affected_pubs	<p>TEXT. Eine Liste von Publikationen, die vor der Ausführung des Skripts synchronisiert werden müssen. Ist die Zeichenfolge leer oder enthält sie NULL, ist keine Synchronisation erforderlich. Dieser Wert gilt nur für SQL Anywhere-Clients. Für Adaptive Server Enterprise hat dieser Parameter den Datentyp VARCHAR(16384). Für IBM DB2 LUW hat dieser Parameter den Datentyp VARCHAR(4000). Für Oracle hat dieser Parameter den Datentyp CLOB.</p>

Syntax	Beschreibung
script	<p>TEXT. Der Inhalt des Passthrough-Skripts. Dieser Wert kann nicht NULL sein. Für Adaptive Server Enterprise hat dieser Parameter den Datentyp VARCHAR(16384). Für IBM DB2 LUW hat dieser Parameter den Datentyp VARCHAR(4000). Für Oracle hat dieser Parameter den Datentyp CLOB.</p> <p>Der Skriptinhalt muss Nicht-NULL sein. Bei entfernten UltraLite-Datenbanken muss der <b>script</b>-Inhalt eine Sammlung von SQL-Anweisungen sein, die durch das Wort <b>go</b> getrennt sind. Das Wort <b>go</b> muss in einer eigenen Zeile stehen. Bei entfernten SQL Anywhere-Datenbanken kann der <b>script</b>-Inhalt eine beliebige Sammlung von SQL-Anweisungen sein, die gültig sind, wenn sie von einem <b>begin...end</b>-Block umgeben sind.</p> <p>Beispiel von <b>script</b>-Inhalt in einer entfernten SQL Anywhere-Datenbank:</p> <pre>DECLARE val INTEGER; SELECT c1 INTO val FROM t1 WHERE pk = 5; IF val &gt; 100 THEN     INSERT INTO t2 VALUES ('c1 is big'); ENDIF</pre> <p>Beispiel von <b>script</b>-Inhalt in einer entfernten UltraLite-Datenbank:</p> <pre>CREATE TABLE myScript (c1 INT NOT NULL PRIMARY KEY) GO INSERT INTO myScript VALUES (1) GO</pre>
description	<p>VARCHAR(2000). Ein Kommentar oder eine Beschreibung des Skripts. Dieser Wert kann NULL sein.</p>

**Bemerkungen**

Diese Prozedur generiert einen Fehler, wenn der angegebene Skriptname bereits in ml\_passthrough\_script vorhanden ist.

**ml\_add\_property-Systemprozedur**

Mit dieser Systemprozedur können Sie MobiLink-Eigenschaften hinzufügen oder löschen. Diese Systemprozedur ändert Zeilen in der Systemtabelle ml\_property.

**Syntax**

```
ml_add_property (  
    'comp_name',  
    'prop_set_name',  
    'prop_name',  
    'prop_value'  
)
```

**Parameter**

<b>Syntax</b>	<b>Beschreibung</b>
comp_name	<p>VARCHAR(128). Der Komponentennamen. Um Eigenschaften mit einer Skriptversion zu speichern, setzen Sie diesen Parameter auf ScriptVersion. Für MobiLink-Servereigenschaften legen Sie ihn auf MLS fest. Für Eigenschaften der serverinitiierten Synchronisation setzen Sie diesen Parameter auf SIS.</p>
prop_set_name	<p>VARCHAR(128). Der Eigenschaftsgruppenname.</p> <p>Wenn der Komponententname den Wert ScriptVersion hat, ist dieser Parameter der Name der Skriptversion.</p> <p>Wenn der Komponententname MLS lautet, kann dieser Parameter ml_user_log_verbosity sein, um die Ausführlichkeitsstufe für einen MobiLink-Benutzer festzulegen, ml_remote_id_log_verbosity, um die Ausführlichkeitsstufe für eine entfernte ID festzulegen, oder locking_and_blocking_detection, damit Informationen zu Sperren und Blockierungen an den MobiLink-Profiler oder die Logdatei des MobiLink-Servers gemeldet werden.</p> <p>Wenn der Komponententname SIS ist, dann ist dieser Parameter der Name des Notifiers, Gateways oder Carriers, für den oder das eine Eigenschaft festgelegt wird.</p>
prop_name	<p>VARCHAR(128). Der Eigenschaftsname.</p> <p>Wenn der Komponententname ScriptVersion ist, dann ist dieser Parameter eine von Ihnen festgelegte Eigenschaft. Sie können diese Eigenschaften mithilfe von DBConnectionContext : getVersion und getProperties referenzieren, oder mit ServerContext: getPropertiesByVersion, getProperties und getPropertySetNames.</p> <p>Wenn der Komponententname MLS lautet, ist diese Eigenschaft entweder ein MobiLink-Benutzername oder eine entfernte ID (jeweils von Ihnen definiert) oder blocking_threshold_in_seconds, damit Informationen zu Sperren und Blockierungen an den MobiLink-Profiler oder die Logdatei des MobiLink-Servers gemeldet werden.</p>

Syntax	Beschreibung
prop_value	<p>TEXT. Der Eigenschaftswert.</p> <p>Wenn prop_set_name entweder ml_user_log_verbosity oder ml_remote_id_log_verbosity ist, muss dies ein gültiger Wert für die mlsrv-Option -v sein.</p> <p>Wenn Informationen zu Sperren und Blockierungen an den MobiLink-Profiler oder die Logdatei des MobiLink-Servers gemeldet werden sollen, ist dieser Wert time_in_seconds.</p> <p>Für Adaptive Server Enterprise hat dieser Parameter den Datentyp VARCHAR(16384). Für IBM DB2 LUW hat dieser Parameter den Datentyp VARCHAR(4000). Für Oracle hat dieser Parameter den Datentyp CLOB. Um eine Eigenschaft zu löschen, setzen Sie diesen Parameter auf NULL.</p>

**Sperren und Blockierungen melden**

Der MobiLink-Server erkennt alle benutzerdefinierten Skripten, die länger als eine bestimmte Zeit laufen (wobei der Standardwert 60 Sekunden beträgt). Er meldet die Informationen zu Sperren und Blockierungen an den MobiLink-Profiler, falls dieser mit dem MobiLink-Server verbunden ist, und protokolliert sie außerdem in der Logdatei des MobiLink-Servers.

Die Informationen zu Sperren und Blockierungen umfassen Folgendes:

- Die Synchronisations-ID
- Die MobiLink-Server-ID der Verbindung, die derzeit blockiert ist
- Die ID der Verbindung, die derzeit den MobiLink-Server blockiert
- Die Gesamtzeit der Blockierung in Sekunden
- Den Namen des Objekts oder Vorgangs, auf dem die Serververbindung blockiert ist

Sie können die Standardzeit ändern, indem Sie die folgende SQL-Anweisung in der konsolidierten Datenbank ausführen:

```
call ml_add_property( 'MLS', 'locking_and_blocking_detection',  
  'blocking_threshold_in_seconds', 'time_in_seconds' );
```

Dabei ist time\_in\_seconds eine Ganzzahl, die den Schwellenwert für die Blockierung in Sekunden angibt. Wenn time\_in\_seconds gleich Null ist, wird diese Funktion deaktiviert.

Dies ist eine statische Eigenschaft. Der MobiLink-Server muss neu gestartet werden, bevor die neuen Werte wirksam werden.

**Ausführlichkeitsstufe des Logs für bestimmte MobiLink-Benutzer und entfernte IDs**

Der MobiLink-Server kann so eingerichtet sein, dass für einen Ziel-MobiLink-Benutzer oder eine entfernte ID eine andere Log-Ausführlichkeitsstufe verwendet wird. Der MobiLink-Server prüft die Tabelle ml\_property alle fünf Minuten und sucht die Ausführlichkeitseinstellungen für einen MobiLink-

Benutzer oder eine entfernte ID. Wenn Ausführlichkeitseinstellungen vorhanden sind, verwendet der MobiLink-Server die neue Einstellung zum Protokollieren von Ausgabenachrichten für den angegebenen MobiLink-Benutzer oder die angegebene entfernte ID. Auf diese Weise können Sie die Details für einen bestimmten Benutzer oder eine bestimmte entfernte ID sehen, ohne hohe Ausführlichkeitseinstellungen zu verwenden, die die Performance der Serverfarm beeinträchtigen würden, und ohne einen Neustart der einzelnen Server in der Farm auszuführen.

Wenn Sie die maximale Ausführlichkeitsstufe für einen MobiLink-Zielbenutzer, z.B. *ML-Benutzer1*, festlegen möchten, melden Sie sich bei der konsolidierten Datenbank an und führen Sie die folgende SQL-Anweisung aus:

```
call ml_add_property( 'MLS', 'ml_user_log_verbosity', 'ml_user1', '-v+' )
```

Wenn Sie die maximale Ausführlichkeitsstufe für eine entfernte Ziel-ID, z.B. *e\_ID\_1*, festlegen möchten, melden Sie sich bei der konsolidierten Datenbank an und führen Sie die folgende SQL-Anweisung aus:

```
call ml_add_property( 'MLS', 'ml_remote_id_log_verbosity', 'rid_1', '-v+' )
```

Die Option *verbose\_setting* muss ein gültiger Wert für die MobiLink-Server-Option -v sein. Um z.B. Zeilendaten und undefinierte Tabellenskripten zu protokollieren, kann *verbose\_setting* -vru oder vru sein. Der MobiLink-Server verwendet diese Ausführlichkeitsstufe für *ml\_user1* oder *rid\_1* nach 5 Minuten. Siehe „[mlsrv16-Option -v](#)“ auf Seite 86.

Wenn Sie die Protokollierungsausführlichkeit für einen MobiLink-Benutzer deaktivieren möchten, melden Sie sich bei der konsolidierten Datenbank an und führen Sie die folgende SQL-Anweisung aus:

```
call ml_add_property( 'MLS', 'ml_user_log_verbosity', 'ml_user1', NULL )
```

Wenn Sie die Protokollierungsausführlichkeit für eine entfernte MobiLink-ID deaktivieren möchten, melden Sie sich bei der konsolidierten Datenbank an und führen Sie die folgende SQL-Anweisung aus:

```
call ml_add_property( 'MLS', 'ml_remote_id_log_verbosity', 'rid_1', NULL )
```

Der MobiLink-Server beendet die Verwendung der vorherigen Ausführlichkeitsstufe für *ml\_user\_1* oder *rid\_1* nach fünf Minuten.

Wenn sowohl **ml\_user\_log\_verbosity** als auch **ml\_remote\_id\_log\_verbosity** für einen angegebenen MobiLink-Benutzer und eine entfernte ID festgelegt werden, und wenn der MobiLink-Benutzername und die entfernte ID in einer Synchronisation für den angegebenen MobiLink-Zielbenutzer identisch sind, verwendet der MobiLink-Server die Einstellung **ml\_remote\_id\_log\_verbosity**, um Ausgabenachrichten zu protokollieren.

## Serverinitiierte Synchronisation

Bei der serverinitiierten Synchronisation ermöglicht es die Systemprozedur `ml_add_property`, Eigenschaften für Notifier, Gateways und Carrier festzulegen.

Wenn Sie beispielsweise die Eigenschaft `server=mailserver1` für ein SMTP-Gateway namens `x` hinzufügen möchten, führen Sie die folgende SQL-Anweisung aus:

```
ml_add_property( 'SIS', 'SMTP(x)', 'server', 'mailserver1' );
```

Die verbosity-Eigenschaft gilt für alle Notifier und Gateways, sodass Sie keinen speziellen Eigenschaftsgruppennamen festlegen können. Um die Einstellung der Ausführlichkeitsstufe zu ändern, lassen Sie "Eigenschaftsgruppename" leer. Beispiel:

```
ml_add_property( 'SIS','','verbosity',2 );
```

### Skriptversion

Für eine regelmäßige MobiLink-Synchronisation können Sie Eigenschaften mit dieser Systemprozedur einer Skriptversion zuordnen. In diesem Fall setzen Sie "Komponentenname" auf ScriptVersion. Sie können beliebige Eigenschaften festlegen und mit Java- und .NET-Klassen auf sie zugreifen.

Wenn Sie beispielsweise einen LDAP-Server einer Skriptversion namens MyVersion zuordnen möchten, führen Sie die folgende SQL-Anweisung aus:

```
ml_add_property( 'ScriptVersion','MyVersion','ldap-server','MyServer' );
```

### Siehe auch

- „MobiLink-Server-Einstellungen für serverinitiierte Synchronisation“ [[MobiLink - Serverinitiierte Synchronisation](#)]
- [DBConnectionContext.getProperties-Methode \[MobiLink-Server Java\] auf Seite 480](#)
- [DBConnectionContext.getVersion-Methode \[MobiLink-Server Java\] auf Seite 482](#)
- [DBConnectionContext.getProperties-Methode \[MobiLink-Server .NET\] auf Seite 568](#) und [DBConnectionContext.getVersion-Methode \[MobiLink-Server .NET\] auf Seite 570](#)
- [ServerContext.getPropertiesByVersion-Methode \[MobiLink-Server Java\] auf Seite 510](#)
- [ServerContext.getProperties-Methode \[MobiLink-Server Java\] auf Seite 509](#)
- [ServerContext.getPropertySetNames-Methode \[MobiLink-Server Java\] auf Seite 510](#)
- [ServerContext.getPropertiesByVersion-Methode \[MobiLink-Server .NET\] auf Seite 605](#), [ServerContext.getProperties-Methode \[MobiLink-Server .NET\] auf Seite 604](#), [ServerContext.getPropertySetNames-Methode \[MobiLink-Server .NET\] auf Seite 605](#)

## ml\_add\_table\_script-Systemprozedur

Mit dieser Systemprozedur können Sie der konsolidierten Datenbank SQL-Tabellenskripten hinzufügen.

### Syntax

```
ml_add_table_script (  
  'version',  
  'table',  
  'event',  
  'script'  
)
```

### Parameter

Syntax	Beschreibung
version	VARCHAR(128). Versionsname

Syntax	Beschreibung
table	VARCHAR(128). Der Tabellenname.
event	VARCHAR(128). Der Ereignisname
script	TEXT. Der Skriptinhalt. Für Adaptive Server Enterprise hat dieser Parameter den Datentyp VARCHAR(16384). Für IBM DB2 LUW hat dieser Parameter den Datentyp VARCHAR(4000). Bei SAP HANA und Oracle ist dieser Parameter vom Typ CLOB.

### Bemerkungen

Um ein Tabellenskript zu löschen, setzen Sie den Skriptinhaltparameter auf NULL.

Wenn Sie ein Skript hinzufügen, wird das Skript in die Tabelle ml\_script eingefügt, und die entsprechenden Referenzen werden definiert, um dem Skript die festgelegte Tabelle und die betreffende Skriptenversion zuzuordnen. Wenn der Versionsname neu ist, wird er automatisch in die Tabelle ml\_version eingefügt.

Der MobiLink-Server muss neu gestartet werden, damit die angegebenen Skriptänderungen wirksam werden, es sei denn, der MobiLink-Server wurde mit der mlsrv16-Option -zf gestartet. Die Option -zf bewirkt, dass der MobiLink-Server zu Beginn jeder Synchronisation eine Prüfung auf Skriptänderungen durchführt.

#### Vorsicht

Das Starten des MobiLink-Server mit der Option -zf hat negative Auswirkungen auf die Performance des MobiLink-Servers und sollte vermieden werden, soweit möglich.

### Siehe auch

- [Systemprozeduren zum Hinzufügen oder Löschen von Skripten auf Seite 648](#)
- [„Hinzufügungen und Löschungen von Skripten“ auf Seite 258](#)
- [„ml\\_add\\_connection\\_script-Systemprozedur“ auf Seite 651](#)
- [„ml\\_add\\_dnet\\_connection\\_script-Systemprozedur“ auf Seite 653](#)
- [„ml\\_add\\_dnet\\_table\\_script-Systemprozedur“ auf Seite 654](#)
- [„ml\\_add\\_java\\_connection\\_script-Systemprozedur“ auf Seite 655](#)
- [„ml\\_add\\_java\\_table\\_script-Systemprozedur“ auf Seite 656](#)
- [„mlsrv16-Option -zf“ auf Seite 102](#)

### Beispiel

Der folgende Befehl fügt ein Tabellenskript ein, das dem Ereignis upload\_insert in der Tabelle Customer zugeordnet ist.

```
call ml_add_table_script( 'default', 'Customer', 'upload_insert',
  'INSERT INTO Customer( cust_id, name, rep_id, active )
    VALUES ( {ml r.cust_id}, {ml r.name}, {ml r.rep_id}, 1 )' )
```

## ml\_add\_user-Systemprozedur

Diese Prozedur ist nur für den internen Gebrauch gedacht.

## ml\_add\_user\_auth\_policy-Systemprozedur

Verwenden Sie diese Systemprozedur, um Richtlinien für die MobiLink-Benutzerauthentifizierung hinzuzufügen.

### Syntax

```
ml_add_user_auth_policy (  
  'policy_name',  
  'primary_ldsrv_name',  
  'secondary_ldsrv_name',  
  'ldap_auto_failback_period'  
  'ldap_failover_to_std'  
)
```

### Parameter

Syntax	Beschreibung
policy_name	VARCHAR(128). Ein eindeutiger Name für eine Richtlinie zur Benutzer-authentifizierung.
primary_ldsrv_name	VARCHAR(128). Gibt den Namen des primären LDAP-Servers an, der zum Authentifizieren dieses Benutzers verwendet werden soll. Der angegebene LDAP-Servername muss bereits in der ml_ldap_server-Tabelle vorhanden sein.
secondary_ldsrv_name	VARCHAR(128). Gibt den Namen des sekundären LDAP-Servers für Failover-zwecke an. Der Name des sekundären LDAP-Servers muss bereits in der ml_ldap_server-Tabelle vorhanden sein.
ldap_auto_failback_period	<p>INTEGER. Verwenden Sie diesen Parameter, um den MobiLink-Server anzuweisen, wann er zur Benutzerauthentifizierung einen Failover auf den primären LDAP-Server durchführen soll. Die Zeit wird in Sekunden angegeben und der Standardwert beträgt 900 Sekunden (15 Minuten).</p> <p>Wenn der primäre LDAP-Server nicht für die Benutzerauthentifizierung verfügbar ist, speichert der MobiLink-Server die Zeit, zu der das Problem erkannt wurde, und schaltet für die Benutzerauthentifizierung auf den Sekundärserver um. Anschließend schaltet der MobiLink-Server auf den Primärserver zurück, um die Benutzerauthentifizierung für alle Benutzer durchzuführen, die derzeit diese Benutzerauthentifizierungsrichtlinie verwenden, sobald die verstrichene Zeit seit dem Erkennen des Fehlers @ldap_auto_failback_period erreicht hat.</p>

Syntax	Beschreibung
ldap_failover_to_std	INTEGER. Gibt an, ob der MobiLink-Server Standardmethoden (Kennwort- und Benutzerauthentifizierungsskripten) zum Authentifizieren des Benutzers verwenden soll, wenn weder der primäre noch der sekundäre LDAP-Server verfügbar ist. Der Wert kann 1 (TRUE) oder 0 (FALSE) sein. Standardwert ist "1".

### Bemerkungen

Diese Prozedur fügt eine Richtlinie zur Benutzerauthentifizierung zur ml\_user\_auth\_policy-Tabelle hinzu, wenn der angegebene Richtliniennamen nicht in der Tabelle vorhanden ist. Wenn der Richtliniennamen bereits in der Tabelle enthalten ist, werden durch Ausführen dieser Prozedur mit Nicht-NULL-Parametern alle entsprechenden Felder mit dem angegebenen Nicht-NULL-Parameter aktualisiert. Die folgende Anweisung aktualisiert beispielsweise die Benutzerauthentifizierungsrichtlinie *policy\_1* so, dass *ldap\_server2* als sekundärer LDAP-Server verwendet wird, und aktiviert den Failover auf die auf Kennwort- und Benutzerauthentifizierungsskripten basierende Standard-Authentifizierung, wenn weder der primäre noch der sekundäre LDAP-Server verfügbar ist.

```
call ml_add_user_auth_policy( 'policy_1', NULL, 'ldap_server2', NULL, 1 )
```

Wenn Sie eine Authentifizierungsrichtlinie löschen möchten, müssen alle Parameter außer *policy\_name* NULL sein.

Beim Hinzufügen einer Richtlinie für die MobiLink-Benutzerauthentifizierung kann der *primary\_ldsrv\_name*-Parameter nicht NULL sein, während der *secondary\_ldsrv\_name*-Parameter NULL sein kann.

### Siehe auch

- „ml\_add\_ldap\_server-Systemprozedur“ auf Seite 658
- „ml\_add\_certificates\_file-Systemprozedur“ auf Seite 650

## ml\_delete\_passthrough-Systemprozedur

Diese gespeicherte Prozedur entfernt die Zeile(n) aus der Tabelle ml\_passthrough, die verursachen, dass das angegebene Skript in der angegebenen Ausführungsreihenfolge in die angegebene entfernte Datenbank heruntergeladen wird. Wenn das Skript in die entfernte Datenbank heruntergeladen wird, bevor es gelöscht wird, dann wird sie nicht aus der entfernten Datenbank gelöscht und normal ausgeführt.

### Syntax

```
ml_delete_passthrough (
  'remote_id',
  'script_name',
  'run_order'
)
```

Parameter

Syntax	Beschreibung
remote_id	VARCHAR(128). Die entfernte ID. Wenn <b>remote_id</b> NULL ist, werden alle Zeilen in der Tabelle ml_passthrough für den angegebenen Skriptnamen und die Ausführungsreihenfolge entfernt.
script_name	VARCHAR(128). Der Skriptname.
run_order	INTEGER. Die Ausführungsreihenfolge des Skripts, das auf die entfernte Datenbank angewendet wird. Wenn <b>run_order</b> NULL ist, werden alle Zeilen für die angegebene <b>remote_id</b> und <b>script_name</b> unabhängig von ihrer Ausführungsreihenfolge aus der Tabelle ml_passthrough entfernt.

Bemerkungen

Der MobiLink-Server entfernt nicht automatisch Einträge aus der Tabelle ml\_passthrough. Sie müssen diese Prozedur verwenden, um veraltete Passthrough-Skripten zu entfernen.

ml\_delete\_passthrough\_repair-Systemprozedur

Mit dieser Systemprozedur löschen Sie eine Reparaturregel aus der Systemtabelle ml\_passthrough\_repair.

Syntax

```
ml_delete_passthrough_repair (  
  'failed_script_name',  
  error_code  
)
```

Parameter

Syntax	Beschreibung
failed_script_name	VARCHAR(128). Der Name des Skripts, auf das eine Regel angewendet wird.
error_code	INTEGER. Der Fehlercode, für den die Regel angewendet wird.

Bemerkungen

Der MobiLink-Server entfernt nicht automatisch Einträge aus der Tabelle ml\_passthrough\_repair. Sie müssen diese Prozedur verwenden, um veraltete Passthrough-Reparaturskripten zu entfernen.

ml\_delete\_passthrough\_script-Systemprozedur

Mit dieser Systemprozedur löschen Sie ein Passthrough-Skript aus der Systemtabelle ml\_passthrough\_script.

**Syntax**

```
ml_delete_passthrough_script (
  'script_name'
)
```

**Parameter**

Syntax	Beschreibung
script_name	VARCHAR(128). Der Name des zu entfernenden Skripts.

**Bemerkungen**

Skripten können nicht entfernt werden, wenn sie in der Systemtabelle ml\_passthrough oder ml\_passthrough\_repair referenziert werden.

Der MobiLink-Server entfernt nicht automatisch Einträge aus der Tabelle ml\_passthrough\_script. Sie müssen diese Prozedur verwenden, um veraltete Passthrough-Skripten zu entfernen.

## ml\_delete\_sync\_state-Systemprozedur

Mit dieser Prozedur löschen Sie einen ungenutzten oder unerwünschten Synchronisationsstatus.

**Syntax**

```
ml_delete_sync_state (
  'user',
  'remote_id'
)
```

**Parameter**

Syntax	Beschreibung
user	VARCHAR(128). Der MobiLink-Benutzername.
remote_id	VARCHAR(128). Die entfernte ID.

**Bemerkungen**

Diese Parameter können den Wert NULL haben. Wenn alle Parameter NULL sind, führt die Prozedur keine Aktionen aus.

Diese gespeicherte Prozedur löscht alle Zeilen für einen bestimmten MobiLink-Benutzernamen und eine entfernte ID aus der Tabelle ml\_subscription. Außerdem löscht sie diese entfernte ID aus der Tabelle ml\_database, wenn die entfernte ID nicht mehr durch Zeilen in der Tabelle ml\_subscription referenziert wird.

Wenn die entfernte ID den Wert NULL hat und der MobiLink-Benutzername nicht NULL ist, werden alle Zeilen, die der angegebene MobiLink-Benutzername referenziert, aus der Tabelle ml\_subscription und

alle entfernten IDs aus der Tabelle ml\_database gelöscht, falls diese entfernten IDs nicht mehr von Zeilen in der Tabelle ml\_subscription referenziert werden.

Wenn der MobiLink-Benutzername den Wert NULL hat und die entfernte ID nicht NULL ist, wird der MobiLink-Benutzer von dieser gespeicherten Prozedur nicht gelöscht, auch wenn alle entfernten IDs aus der Tabelle ml\_database gelöscht wurden und dieser Benutzer nicht weiter von Zeilen in der Tabelle ml\_subscription referenziert wird. Wenn dieser MobiLink-Benutzer gelöscht werden muss, können Sie ihn löschen, indem Sie einen Befehl wie den folgenden ausgeben:

```
delete from ml_user where name = 'user_name'
```

Dabei steht *Benutzername* für den MobiLink-Benutzer, den Sie löschen möchten.

Verwenden Sie diese gespeicherte Prozedur mit großer Vorsicht, da der MobiLink-Server bei der nächsten Synchronisationsanforderung des MobiLink-Clients diese entfernte ID in den Tabellen ml\_database und ml\_subscription automatisch wieder hinzufügt, ohne den Synchronisationsstatus für diese entfernte ID zu überprüfen. Dies kann Dateninkonsistenz verursachen, wenn der Synchronisationsstatus für eine entfernte ID gelöscht wird, die beim letzten Synchronisationsversuch nicht erfolgreich synchronisiert werden konnte.

Diese Prozedur entfernt für die angegebene entfernte ID alle Zeilen aus der Tabelle ml\_subscription und aus der Tabelle ml\_database.

**Beispiel**

Im nachfolgenden Beispiel werden die Informationen der MobiLink-Systemtabelle zu entfernten Datenbanken mit der entfernten ID remote\_db\_for\_John für den MobiLink-Benutzer John bereinigt:

```
CALL ml_delete_sync_state( 'John', 'remote_db_for_John' )
```

**ml\_delete\_sync\_state\_before-Systemprozedur**

Mit dieser Prozedur bereinigen Sie die MobiLink-Systemtabellen, wenn Sie entfernte Datenbanken gelöscht haben.

**Syntax**

```
ml_delete_sync_state_before (
'ts'
)
```

**Parameter**

Syntax	Beschreibung
ts	TIMESTAMP. Datum und Uhrzeit müssen genau in der Reihenfolge erscheinen wie in der konsolidierten Datenbank festgelegt. Wenn das datetime-Format in der konsolidierten Datenbank 'jjjj/mm/tt hh:mm:ss.ssss' lautet, muss der Zeitstempel in der Reihenfolge Jahr, Monat, Tag, Stunde, Minute, Sekunde und Sekundenbruchteil angezeigt werden.

## Bemerkungen

Diese gespeicherte Prozedur entfernt Zeilen aus MobiLink-Systemtabellen, die zu entfernten Datenbanken gehören, die nicht mehr verwendet werden. Beachten Sie insbesondere Folgendes:

- Löscht alle Zeilen aus der Systemtabelle ml\_subscription, bei denen sowohl last\_upload\_time als auch last\_download\_time einen früheren Zeitpunkt aufweisen als der angegebene Zeitstempel.
- Entfernt entfernte IDs aus der Systemtabelle ml\_database, wenn sie nicht weiter von Zeilen in der Tabelle ml\_subscription referenziert werden.

Verwenden Sie diese Systemprozedur nicht für einen so kurz zurückliegenden Zeitraum, dass sie unter Umständen Zeilen für entfernte Datenbanken löscht, die gar nicht gelöscht wurden. Tun Sie es doch, kann das Löschen der Zeilen in den Tabellen ml\_subscription und ml\_database Probleme für entfernte Datenbanken verursachen, die aufgrund eines erfolglosen Uploads einen unbekannten Status haben. In diesem unbekannten Status ist die entfernte Datenbank davon abhängig, dass die MobiLink-Systemtabellen Daten erneut senden.

Der Zeitstempel für diese Prozedur muss das richtige Datum-Zeit-Format haben, da die Prozedur das Datum-Zeit-Format des Parameters nicht validiert.

## Beispiel

Im folgenden Beispiel werden MobiLink-Systemtabelleninformationen über entfernte Datenbanken bereinigt, die seit 10. Januar 2004 nicht synchronisiert wurden. Das Beispiel gilt für eine konsolidierte SQL Anywhere-Datenbank, wobei das Datums- und Zeitformat in der konsolidierten Datenbank jjjj/mm/tt hh:mm:ss.ssss ist

```
CALL ml_delete_sync_state_before( '2004/01/10 00:00:00' )
```

## ml\_delete\_user-Systemprozedur

Diese Prozedur ist nur für den internen Gebrauch gedacht.

## ml\_model\_drop-Systemprozedur

Mithilfe dieser Prozedur können Sie Synchronisationsmodelle löschen, die mit dem MobiLink 16-Plug-In für Sybase Central installiert wurden.

## Syntax

```
ml_model_drop (
  'script_version'
)
```

**Parameter**

Syntax	Beschreibung
<i>script_version</i>	VARCHAR(128). Der Name der Skriptversion, die dem zu löschenden Synchronisationsmodell zugeordnet ist.

**Bemerkungen**

Diese gespeicherte Prozedur entfernt die in der benannten Skriptversion enthaltenen Synchronisationsskripten sowie alle Schemata, die beim Deployment des Synchronisationsmodells erstellt wurden, einschließlich Schattentabellen, Protokollierungsspalten, Trigger und Indizes.

Schemata, die gemeinsam mit einer anderen *script\_version* genutzt werden, werden nicht gelöscht.

Es wird kein Schema gelöscht, wenn eine *script\_version* außerhalb des MobiLink 16-Plug-Ins manuell installiert wurde.

**Siehe auch**

- „[ml\\_model\\_check\\_all\\_schema-Systemprozedur](#)“ auf Seite 676
- „[ml\\_model\\_check\\_version\\_schema-Systemprozedur](#)“ auf Seite 677

## ml\_model\_check\_all\_schema-Systemprozedur

Verwenden Sie diese Prozedur, um den Status der einzelnen Schemaobjekte zu überprüfen, die für bereitgestellte Synchronisationsmodelle erforderlich sind. Diese gespeicherte Prozedur gibt Informationen für alle bereitgestellten Synchronisationsmodelle zurück.

**Syntax**

**ml\_model\_check\_all\_schema**

**Bemerkungen**

Diese Prozedur gibt eine Ergebnismenge mit dem Status der einzelnen Schemaobjekte zurück, die für alle bereitgestellten Synchronisationsmodelle erforderlich sind.

Für Skriptversionen, die außerhalb von Sybase Central installiert wurden, oder für Synchronisationsmodelle, deren Deployment vor Version 16 durchgeführt wurde, werden keine Ergebnisse zurückgegeben .

Die Tabelle enthält die folgenden Spalten:

- **schema\_owner**    Gibt den Eigentümer des Schemas an.
- **table\_name**     Gibt den Tabellennamen an.

- **schema\_type** Gibt den Schematyp an. Es kann sich um einen der folgenden Typen handeln:
  - TABLE
  - INDEX
  - COLUMN
  - TRIGGER
  - PROCEDURE
- **object\_name** Gibt den Objektnamen an.
- **locked** Wenn diese Spalte auf 1 gesetzt ist, wird das Schema nie durch das Plug-In geändert oder gelöscht. Schemata, die von vor einem Deployment vorhandenen Synchronisationsmodellen verwendet werden, werden als gesperrt markiert.
- **used\_by** Die Skriptversion, die das Schemaobjekt erfordert.
- **status** Der Status kann einer der folgenden Werte sein:
  - INSTALLED** Das Schema ist korrekt installiert.
  - MISSING** Das Schema ist nicht installiert.
  - MISMATCH** Die installierte Schema ist ein anderes als das erforderliche.
  - UNVERIFIED** Das Schema ist vorhanden, aber es liegen nicht genug Informationen vor, um zu ermitteln, ob es korrekt definiert ist.
  - UNUSED** Kein Synchronisationsmodell verwendet dieses Schemaobjekt.
- **overwrite\_action** Dies kann einer der folgenden Werte sein:
  - REPLACE** Bei einem erneuten Deployment des Modells wird das vorhandene Schema gelöscht und neu erstellt
  - CREATE** Bei einem erneuten Deployment des Modells wird das Schema erstellt.
  - SKIP** Entweder ist das Schema bereits korrekt installiert, oder es besteht ein Schemakonflikt, durch den eine richtige Installation blockiert wird.
- **preserve\_action** Reserviert für spätere Verwendung.

**Siehe auch**

- „[ml\\_model\\_drop-Systemprozedur](#)“ auf Seite 675
- „[ml\\_model\\_check\\_version\\_schema-Systemprozedur](#)“ auf Seite 677

## **ml\_model\_check\_version\_schema-Systemprozedur**

Verwenden Sie diese Prozedur, um den Status der einzelnen Schemaobjekte zu überprüfen, die für bereitgestellte Synchronisationsmodelle erforderlich sind. Diese gespeicherte Prozedur gibt Informationen für die angegebene Skriptversion zurück.

## Syntax

```
ml_model_check_version_schema (  
    'script_version'  
)
```

## Parameter

Syntax	Beschreibung
<i>script_version</i>	VARCHAR(128). Der Name der Skriptversion, die dem zu überprüfenden Synchronisationsmodell zugeordnet ist.

## Bemerkungen

Diese Prozedur gibt eine Ergebnismenge zurück, die den Status der einzelnen Schemaobjekte für die angegebene Skriptversion enthält.

Für Skriptversionen, die außerhalb von Sybase Central installiert wurden, oder für Synchronisationsmodelle, deren Deployment vor Version 16 durchgeführt wurde, werden keine Ergebnisse zurückgegeben .

Die Tabelle enthält die folgenden Spalten:

- **schema\_owner**    Gibt den Eigentümer des Schemas an.
- **table\_name**    Gibt den Tabellennamen an.
- **schema\_type**    Gibt den Schematyp an. Es kann sich um einen der folgenden Typen handeln:
  - TABLE
  - INDEX
  - COLUMN
  - TRIGGER
  - PROCEDURE
- **object\_name**    Gibt den Objektnamen an.
- **locked**    Wenn diese Spalte auf 1 gesetzt ist, wird das Schema nie durch das Plug-In geändert oder gelöscht. Schemata, die von vor einem Deployment vorhandenen Synchronisationsmodellen verwendet werden, werden als gesperrt markiert.
- **status**    Der Status kann einer der folgenden Werte sein:
  - INSTALLED**    Das Schema ist korrekt installiert.
  - MISSING**    Das Schema ist nicht installiert.
  - MISMATCH**    Das installierte Schema ist ein anderes als das erforderliche.
  - UNVERIFIED**    Das Schema ist vorhanden, aber es liegen nicht genug Informationen vor, um zu ermitteln, ob es korrekt definiert ist.

**UNUSED** Kein Synchronisationsmodell verwendet dieses Schemaobjekt.

- **overwrite\_action** Dies kann einer der folgenden Werte sein:

**REPLACE** Bei einem erneuten Deployment des Modells wird das vorhandene Schema gelöscht und neu erstellt

**CREATE** Bei einem erneuten Deployment des Modells wird das Schema erstellt.

**SKIP** Entweder ist das Schema bereits korrekt installiert, oder es besteht ein Schemakonflikt, durch den eine richtige Installation blockiert wird.

- **preserve\_action** Reserviert für spätere Verwendung.

#### Siehe auch

- „[ml\\_model\\_drop-Systemprozedur](#)“ auf Seite 675
- „[ml\\_model\\_check\\_all\\_schema-Systemprozedur](#)“ auf Seite 676

## ml\_ra\_add\_agent\_id-Systemprozedur

Verwenden Sie diese Prozedur, um in der konsolidierten Datenbank einen neuen entfernten Agenten zu definieren.

#### Parameter

Syntax	Beschreibung
agent_id	VARCHAR(128). Dies ist ein IN-Parameter, der die ID des in der konsolidierten Datenbank zu definierenden neuen Agenten angibt.

#### Bemerkungen

Wenn ein Agent sich mit dem MobiLink-Server verbindet, ohne dass zuvor `ml_ra_add_agent_id` aufgerufen wurde, wird der Agent automatisch zur konsolidierten Datenbank hinzugefügt. Alle Eigenschaften für diesen Agenten sind jedoch Standardwerte.

#### Siehe auch

- „[ml\\_ra\\_manage\\_remote\\_db-Systemprozedur](#)“ auf Seite 693
- „[ml\\_ra\\_clone\\_agent\\_properties-Systemprozedur](#)“ auf Seite 681
- „[ml\\_ra\\_set\\_agent\\_property-Systemprozedur](#)“ auf Seite 695

## ml\_ra\_assign\_task-Systemprozedur

Verwenden Sie diese Prozedur, um einem bestimmten entfernten Agenten eine Aufgabe zuzuordnen.

**Parameter**

Syntax	Beschreibung
agent_id	VARCHAR(128). Dies ist ein IN-Parameter, der die ID des Agenten angibt, dem die Aufgabe zugeordnet werden soll.
task_name	VARCHAR(128). Dies ist ein IN-Parameter, der den Namen der zuzuordnenden Aufgabe angibt.

**Bemerkungen**

Aufgaben müssen zuerst in Sybase Central mit dem MobiLink 16-Plug-In definiert werden, bevor diese Systemprozedur aufgerufen wird.

Manche Aufgaben haben eine bestimmte entfernte Datenbank als Ziel. Wenn dies der Fall ist, muss der Agent eine entfernten Datenbank dieses Typs verwalten.

Wenn eine Aufgabe bereits einem Agent zugewiesen und danach abgeschlossen wurde, kann die Aufgabe erneut zugewiesen werden. Dadurch wird die Aufgabe wieder aktiv und sie wird gemäß ihrem Zeitplan ausgeführt.

Aufgaben müssen zunächst in Sybase Central definiert werden, bevor der Aufgabenname aufgerufen wird.

**Siehe auch**

- [„ml\\_ra\\_cancel\\_task\\_instance-Systemprozedur“ auf Seite 681](#)
- [„ml\\_ra\\_notify\\_task-Systemprozedur“ auf Seite 694](#)
- [„ml\\_ra\\_cancel\\_notification-Systemprozedur“ auf Seite 680](#)

## ml\_ra\_cancel\_notification-Systemprozedur

Verwenden Sie diese Prozedur, um eine vom Server eingeleitete entfernte Aufgabe (SIRT-Anforderung) abzuberechnen, die nicht mehr benötigt wird.

**Parameter**

Syntax	Beschreibung
agent_id	VARCHAR(128). Dies ist ein IN-Parameter, der die ID des Agenten angibt, der für die abzuberechnende Aufgabe zuständig ist.
task_name	VARCHAR(128). Dies ist ein IN-Parameter, der den Namen der abzuberechnenden Aufgabe angibt.

**Siehe auch**

- „ml\_ra\_notify\_task-Systemprozedur“ auf Seite 694
- „ml\_ra\_assign\_task-Systemprozedur“ auf Seite 679
- „ml\_ra\_delete\_task-Systemprozedur“ auf Seite 683
- „ml\_ra\_cancel\_task\_instance-Systemprozedur“ auf Seite 681

## ml\_ra\_cancel\_task\_instance-Systemprozedur

Verwenden Sie diese Prozedur, um eine entfernte Aufgabeninstanz abzuberechnen.

**Parameter**

Syntax	Beschreibung
agent_id	VARCHAR(128). Dies ist ein IN-Parameter, der die ID des Agenten angibt, der für die abzuberechnende Aufgabe zuständig ist.
task_name	VARCHAR(128). Dies ist ein IN-Parameter, der den Namen der abzuberechnenden Aufgabe angibt.

**Bemerkungen**

Mit dieser Systemprozedur können Sie eine Aufgabe abbrechen, die nicht mehr benötigt wird.

Die abgebrochene Aufgabe wird mit dem Status **Laufende Aufgaben abbrechen** gemeldet, bis der Agent alle aktiven Instanzen abgeschlossen hat und den Status "Abgebrochen" durch eine Synchronisation der Agentendatenbank bestätigt.

**Siehe auch**

- „ml\_ra\_cancel\_notification-Systemprozedur“ auf Seite 680
- „ml\_ra\_notify\_task-Systemprozedur“ auf Seite 694
- „ml\_ra\_assign\_task-Systemprozedur“ auf Seite 679
- „ml\_ra\_delete\_task-Systemprozedur“ auf Seite 683

## ml\_ra\_clone\_agent\_properties-Systemprozedur

Verwenden Sie diese Prozedur, um alle Eigenschaften eines entfernten Agenten auf einmal festzulegen.

**Parameter**

Syntax	Beschreibung
dst_agent_id	VARCHAR(128). Dies ist ein IN-Parameter, der die ID des zu erstellenden Ziel-Agenten angibt.
src_agent_id	VARCHAR(128). Dies ist ein IN-Parameter, der die ID des Agenten angibt, der für die Erstellung des neuen Agenten geklont werden soll.

## Bemerkungen

Verwenden Sie diese Prozedur, um alle Eigenschaften des angegebenen Agenten auf einmal festzulegen. Alle Eigenschaften eines vorhandenen Agenten werden in den neuen Agenten kopiert. Einzelne Agenteneigenschaften lassen sich leichter in Sybase Central festlegen.

Zugeordnete Aufgaben und verwaltete entfernte Datenbanken werden nicht in den neuen Agenten kopiert.

## Siehe auch

- „ml\_ra\_set\_agent\_property-Systemprozedur“ auf Seite 695
- „ml\_ra\_add\_agent\_id-Systemprozedur“ auf Seite 679
- „ml\_ra\_manage\_remote\_db-Systemprozedur“ auf Seite 693

## ml\_ra\_delete\_agent\_id-Systemprozedur

Verwenden Sie diese Prozedur, um einen definierten Agenten aus der konsolidierten Datenbank zu löschen.

## Parameter

Syntax	Beschreibung
agent_id	VARCHAR(128). Dies ist ein IN-Parameter, der die ID des zu löschenden Agenten angibt.

## Bemerkungen

Wenn Sie einen Agenten löschen, der entfernte Datenbanken verwaltet hat, sind diese entfernten Datenbanken nicht mehr verwaltet.

## Siehe auch

- „ml\_ra\_delete\_events\_before-Systemprozedur“ auf Seite 682
- „ml\_ra\_delete\_remote\_id-Systemprozedur“ auf Seite 683
- „ml\_ra\_unmanage\_remote\_db-Systemprozedur“ auf Seite 696

## ml\_ra\_delete\_events\_before-Systemprozedur

Verwenden Sie diese Prozedur, um nicht mehr benötigte Ereignisse aus der konsolidierten Datenbank zu löschen.

## Parameter

Syntax	Beschreibung
delete_rows_older_than	TIMESTAMP. Dies ist ein IN-Parameter, der festlegt, dass ältere Ereignisse als der angegebene Wert aus der konsolidierten Datenbank gelöscht werden.

**Bemerkungen**

Wenn die entfernten Aufgaben häufig den Status zurückgeben, kann sich in der konsolidierten Datenbank eine große Zahl von Ereignissen ansammeln.

**Siehe auch**

- [„ml\\_ra\\_get\\_latest\\_event\\_id-Systemprozedur“ auf Seite 688](#)

## ml\_ra\_delete\_remote\_id-Systemprozedur

Verwenden Sie diese Prozedur, um eine entfernte Datenbank, die nicht mehr benötigt wird, aus der konsolidierten Datenbank zu löschen.

**Parameter**

Syntax	Beschreibung
remote_id	VARCHAR(128). Dies ist ein IN-Parameter, der die entfernte ID für die zu löschende entfernte Datenbank angibt.

**Bemerkungen**

Diese Prozedur schlägt fehl, wenn noch Aufgaben für die angegebene entfernte ID aktiv sind. Um die Löschung zu erzwingen, müssen Sie zuerst die ID des Agenten löschen, der die entfernte Datenbank verwaltet.

**Siehe auch**

- [„ml\\_ra\\_delete\\_agent\\_id-Systemprozedur“ auf Seite 682](#)
- [„ml\\_ra\\_delete\\_task-Systemprozedur“ auf Seite 683](#)

## ml\_ra\_delete\_task-Systemprozedur

Verwenden Sie diese Prozedur, um eine entfernte Aufgabe aus der konsolidierten Datenbank zu löschen.

**Parameter**

Syntax	Beschreibung
task_name	VARCHAR(128). Dies ist ein IN-Parameter, der den Namen der zu löschenden entfernten Aufgabe angibt.

**Bemerkungen**

Diese Systemprozedur schlägt fehl, wenn noch aktive Aufgabeninstanzen vorhanden sind.

### Siehe auch

- „ml\_ra\_delete\_agent\_id-Systemprozedur“ auf Seite 682
- „ml\_ra\_delete\_events\_before-Systemprozedur“ auf Seite 682
- „ml\_ra\_delete\_remote\_id-Systemprozedur“ auf Seite 683

## ml\_ra\_get\_agent\_events-Systemprozedur

Verwenden Sie diese Prozedur, um Ereignisse abzufragen.

### Parameter

Syntax	Beschreibung
start_at_event_id	BIGINT. Dies ist ein IN-Parameter, der die ID des Ereignisses angibt, bei dem mit der Abfrage begonnen werden soll.
max_events_to_fetch	BIGINT. Dies ist ein IN-Parameter, der die maximale Anzahl der abzurufenden Ereignisse angibt.

### Rückgabe

Ergebnis	Beschreibung
event_id	BIGINT. Die jedem Ereignis zugeordnete eindeutige ID. Der Wert wird für jedes neue Ereignis um 1 erhöht.
event_class	VARCHAR(1). Die Ereignisklasse. Die Klasse kann entweder <b>I</b> für Informationen oder <b>E</b> für Fehler sein.

Ergebnis	Beschreibung
event_type	<p>VARCHAR(8). Die Ereignistypen werden nachfolgend aufgelistet.</p> <ul style="list-style-type: none"> <li>• <b>ANEW</b> Ein neuer Agent wurde in der konsolidierten Datenbank definiert. Dies kann vorkommen, wenn ml_ra_add_agent aufgerufen wird oder wenn der Agent bei seinem ersten Verbindungsaufbau mit der konsolidierten Datenbank nicht vorkonfiguriert ist.</li> <li>• <b>AFIRST</b> Tritt bei der ersten Synchronisation eines Agenten auf.</li> <li>• <b>ADUP</b> Eine mehrfach vorhandene Agenten-ID wurde gefunden, d.h. zwei oder mehr Agenten versuchen, dieselbe ID zu verwenden. Der Ergebnistext ist die entfernte ID der Agenten-Datenbank für den blockierten Agenten.</li> <li>• <b>ARESET</b> Ein Agent hat seine Agenten-Datenbank neu aufgebaut. Möglicherweise sind einige Verarbeitungsfortschritte und Ergebnisse von Aufgaben verloren gegangen.</li> <li>• <b>TByte</b> Die Ausführung einer Aufgabe hat begonnen.</li> <li>• <b>TE</b> Die Ausführung der Aufgabe wurde ohne schwerwiegenden Fehler beendet.</li> <li>• <b>TW</b> Eine Aufgabe kann erst weiter ausgeführt werden, wenn ein Wiederholungsintervall abgelaufen ist.</li> <li>• <b>TAC</b> Die Ausführung der Aufgabe wurde beendet, weil ein Befehl abgebrochen wurde.</li> <li>• <b>TAT</b> Die Ausführung der Aufgabe wurde beendet, weil die maximal zulässige Ausführungszeit überschritten wurde.</li> <li>• <b>TAR</b> Die Ausführung der Aufgabe wurde beendet, weil die maximale Anzahl an Wiederholungen überschritten wurde.</li> <li>• <b>TFS</b> Die Aufgabe wurde abgeschlossen und wird nicht erneut ausgeführt, weil es sich um eine einmalige Aufgabe handelt, die erfolgreich ausgeführt wurde.</li> <li>• <b>TFF</b> Die Aufgabe wurde abgeschlossen und wird nicht erneut ausgeführt, weil es sich um eine einmalige Aufgabe handelt, die fehlgeschlagen ist.</li> <li>• <b>TFE</b> Die Aufgabe wurde abgeschlossen und wird nicht erneut ausgeführt, weil der Zeitplan für die Aufgabe abgelaufen ist.</li> <li>• <b>TFC</b> Die Aufgabe wurde abgeschlossen und wird nicht erneut ausgeführt, weil die Aufgabe vom Server abgebrochen wurde.</li> </ul>

Ergebnis	Beschreibung
	<ul style="list-style-type: none"> <li>• <b>CR - Command Result (Befehlsergebnis)</b> Ergebniscode und Ergebnistext werden mit für den Typ des Befehls spezifischen Werten gefüllt.</li> <li>• <b>CE - Command Error (Befehlsfehler)</b> Ergebniscode und Ergebnistext werden mit für den Typ des Befehls spezifischen Werten gefüllt.</li> </ul>
agent_id	VARCHAR(128). Die ID des Agenten, der dieses Ereignis erzeugt hat.
remote_id	VARCHAR(128). Die ID der entfernten Datenbank, für die das Ereignis gilt. Dies wird nur für aufgabenbezogene Ereignisse festgelegt, die eine bestimmte entfernte Datenbank als Ziel haben.
task_name	VARCHAR(128). Der Name der Aufgabe. Dies wird nur für aufgabenbezogene Ereignisse festgelegt.
command_number	INTEGER. Die Befehlsnummer innerhalb einer Aufgabe, für die dieses Ereignis gilt. Dies wird nur für befehlspezifische Ereignisse festgelegt.
run_number	BIGINT. Die eindeutige Nummer, die jeder Ausführung einer Aufgabe zugewiesen wird. Dies wird nur für aufgabenspezifische Ereignisse festgelegt.
duration	INTEGER. Die durch das Ereignis in Anspruch genommene Zeit. Dies wird nur für befehlspezifische Ereignisse festgelegt.
event_time	TIMESTAMP. Der Zeitpunkt, zu dem das Ereignis stattgefunden hat. Für die meisten Ereignisse basiert die Zeit auf der Uhr des Computers, auf dem der Agent ausgeführt wird.
event_received	TIMESTAMP. Der Zeitpunkt, zu dem der Server das Ereignis empfangen hat. Dies wird immer anhand der Uhr für die konsolidierte Datenbank festgelegt.
result_code	BIGINT. Ein ereignisspezifischer BIGINT-Wert. Beispiel: Der Code für das Befehlsergebnis einer SQL-Abfrage wäre der SQLCODE-Wert.
result_text	LONG VARCHAR. Ein ereignisspezifischer LONG VARCHAR-Wert. Beispiel: Für das Befehlsergebnis einer SQL-Abfrage würde diese Spalte ein CSV-Format der Ergebnismenge enthalten.
p_crsr	SYS_REF_CURSOR. Dies ist ein OUT-Parameter nur für Oracle.

## Bemerkungen

Alternativ dazu können Sie die `ml_ra_get_task_results`-Prozedur verwenden, mit der nur Ereignisse in Bezug auf eine bestimmte Ausführung einer Aufgabe abgerufen werden. Sie können einen NULL-Wert für die Ausführungsnummer übergeben, um die letzte Ausführung einer Aufgabe abzurufen.

Eine Möglichkeit, diese Prozedur einzusetzen, besteht darin, mit `ml_ra_get_agent_events` auf ein Aufgabenende-Ereignis (**TE**) zu warten und anschließend `ml_ra_get_task_results` aufzurufen, um die einzelnen Befehlsergebnisse abzurufen, die möglicherweise verarbeitet werden müssen.

#### Siehe auch

- „`ml_ra_get_task_results`-Systemprozedur“ auf Seite 690
- „`ml_ra_get_task_results`-Systemprozedur“ auf Seite 690

## `ml_ra_get_agent_ids`-Systemprozedur

Verwenden Sie diese Prozedur, um alle Agenten in der konsolidierten Datenbank abzurufen.

#### Rückgabe

Ergebnis	Beschreibung
<code>agent_id</code>	VARCHAR(128). Die Agenten-ID.
<code>remote_id</code>	VARCHAR(128). Die entfernte ID der Agenten-Datenbank.
<code>last_download_time</code>	TIMESTAMP. Die Uhrzeit des letzten Downloads.
<code>last_upload_time</code>	TIMESTAMP. Die Uhrzeit des letzten Uploads.
<code>active_task_count</code>	INTEGER. Die Anzahl der aktiven Aufgaben.
<code>description</code>	VARCHAR(2048). Reserviert für spätere Verwendung.
<code>p_crsr</code>	SYS_REF_CURSOR. Dies ist ein OUT-Parameter nur für Oracle.

#### Siehe auch

- „`ml_ra_get_remote_ids`-Systemprozedur“ auf Seite 689

## `ml_ra_get_agent_properties`-Systemprozedur

Verwenden Sie diese Prozedur, um alle für einen Agenten festgelegten Eigenschaften anzuzeigen.

#### Parameter

Syntax	Beschreibung
<code>agent_id</code>	VARCHAR(128). Dies ist ein IN-Parameter, der die ID des Agenten angibt, für den Sie Eigenschaften festlegen.

## Rückgabe

Ergebnisse	Beschreibung
property_name	VARCHAR(128). Der Eigenschaftsname.
property_value	VARCHAR(2048). Der Wert der Eigenschaft
last_modified	TIMESTAMP. Die Zeit, zu der die Eigenschaft zuletzt geändert wurde.
p_crsr	SYS_REF_CURSOR. Dies ist ein OUT-Parameter nur für Oracle.

## Siehe auch

- [„ml\\_ra\\_get\\_agent\\_ids-Systemprozedur“ auf Seite 687](#)
- [„ml\\_ra\\_clone\\_agent\\_properties-Systemprozedur“ auf Seite 681](#)

## ml\_ra\_get\_latest\_event\_id-Systemprozedur

Verwenden Sie diese Prozedur, um zu ermitteln, wie viele neue Ereignisse vorhanden sind.

## Parameter

Syntax	Beschreibung
event_id	BIGINT. Dies ist ein OUT-Parameter, der die ID des letzten Ereignisses angibt.

## Bemerkungen

Um zu ermitteln, wie viele neue Ereignisse vorhanden sind, rufen Sie die Systemprozedur ml\_ra\_get\_latest\_event\_id auf und subtrahieren Sie die letzte verarbeitete Ereignis-ID.

## Siehe auch

- [„ml\\_ra\\_get\\_agent\\_events-Systemprozedur“ auf Seite 684](#)

## ml\_ra\_get\_orphan\_taskdb-Systemprozedur

Mit dieser Prozedur können Sie eine Liste der Agenten-Datenbanken anzeigen, die als Waisen gelten, d.h. nicht über eine gültige Agenten-ID verfügen.

## Rückgabe

Ergebnisse	Beschreibung
remote_id	VARCHAR(128). Die entfernte ID.
orig_agent_id	VARCHAR(128). Die ID des ursprünglichen Agenten, dem die Agent-Datenbank zugeordnet wurde.

Ergebnisse	Beschreibung
last_sync	TIMESTAMP. Die Uhrzeit der letzten Synchronisation.
p_crsr	SYS_REF_CURSOR. Dies ist ein OUT-Parameter nur für Oracle.

### Bemerkungen

Verwaiste Datenbanken können das Ergebnis einer Reihe von Problemen in einem Synchronisationssystem sein, z.B. der Erstellung mehrfach vorhandener Agenten-IDs auf verschiedenen Computern oder des Falls, dass zwei Agenten-Datenbanken die gleiche Agenten-ID zu verwenden versuchen.

Das Feld für die entfernte ID enthält den Namen des Computers, um die Problem diagnose zu erleichtern.

### Siehe auch

- „ml\_ra\_reassign\_taskdb-Systemprozedur“ auf Seite 694

## ml\_ra\_get\_remote\_ids-Systemprozedur

Verwenden Sie diese Prozedur, um alle entfernten Datenbanken in der konsolidierten Datenbank abzurufen, mit Ausnahme der Agenten-Datenbanken.

### Parameter

Keine.

### Rückgabe

Ergebnisse	Beschreibung
remote_id	VARCHAR(128). Die entfernte ID der entfernten Datenbank.
schema_name	VARCHAR(128). Der Typ der entfernten Datenbank.
agent_id	VARCHAR(128). Die Agenten-ID der entfernten Datenbank.
agent_conn_str	VARCHAR(2048). Die Agenten-Verbindungszeichenfolge.
last_download_time	TIMESTAMP. Die Uhrzeit des letzten Downloads.
last_upload_time	TIMESTAMP. Die Uhrzeit des letzten Uploads.
description	VARCHAR(128). Die Beschreibung der Datenbank.
p_crsr	SYS_REF_CURSOR. Dies ist ein OUT-Parameter nur für Oracle.

## Siehe auch

- „[ml\\_ra\\_get\\_agent\\_properties-Systemprozedur](#)“ auf Seite 687

# ml\_ra\_get\_task\_results-Systemprozedur

Verwenden Sie diese Prozedur für den Abruf von Ereignissen in Bezug auf eine bestimmte Ausführung einer Aufgaben.

## Parameter

Syntax	Beschreibung
agent_id	VARCHAR(128). Dies ist ein IN-Parameter, der die ID des Agenten angibt, für den Sie Ergebnisse abrufen wollen.
task_name	VARCHAR(128). Dies ist ein IN-Parameter, der den Namen der Aufgabe angibt, für die Sie Ergebnisse abrufen wollen.
run_number	INTEGER. Dies ist ein IN-Parameter, der die Ausführungsnummer angibt, für die Sie Ergebnisse abrufen wollen.

## Rückgabe

Ergebnis	Beschreibung
event_id	BIGINT. Die jedem Ereignis zugeordnete eindeutige ID. Der Wert wird für jedes neue Ereignis um 1 erhöht.
event_class	VARCHAR(1). Die Ereignisklasse. Die Klasse kann entweder <b>I</b> für Informationen oder <b>E</b> für Fehler sein.
event_type	VARCHAR(8). Der Ereignistyp.
agent_id	VARCHAR(128). Die ID des Agenten, der dieses Ereignis erzeugt hat.
remote_id	VARCHAR(128). Die ID der entfernten Datenbank, für die das Ereignis gilt. Dies wird nur für aufgabenbezogene Ereignisse festgelegt, die eine bestimmte entfernte Datenbank als Ziel haben.
task_name	VARCHAR(128). Der Name der Aufgabe. Dies wird nur für aufgabenbezogene Ereignisse festgelegt.
command_number	INTEGER. Die Befehlsnummer innerhalb einer Aufgabe, für die dieses Ereignis gilt. Dies wird nur für befehlspezifische Ereignisse festgelegt.
run_number	BIGINT. Die eindeutige Nummer, die jeder Ausführung einer Aufgabe zugewiesen wird. Dies wird nur für aufgabenspezifische Ereignisse festgelegt.

Ergebnis	Beschreibung
duration	INTEGER. Die durch das Ereignis in Anspruch genommene Zeit. Dies wird nur für befehlspezifische Ereignisse festgelegt.
event_time	TIMESTAMP. Der Zeitpunkt, zu dem das Ereignis stattgefunden hat. Für die meisten Ereignisse basiert die Zeit auf der Uhr des Computers, auf dem der Agent ausgeführt wird.
event_received	TIMESTAMP. Der Zeitpunkt, zu dem der Server das Ereignis empfangen hat. Dies wird immer anhand der Uhr für die konsolidierte Datenbank festgelegt.
result_code	BIGINT. Ein ereignisspezifischer BIGINT-Wert. Beispiel: Der Code für das Befehlsergebnis einer SQL-Abfrage wäre der SQLCODE-Wert.
result_text	LONG VARCHAR. Ein ereignisspezifischer LONG VARCHAR-Wert. Beispiel: Für das Befehlsergebnis einer SQL-Abfrage würde diese Spalte ein CSV-Format der Ergebnismenge enthalten.
p_crsr	SYS_REF_CURSOR. Dies ist ein OUT-Parameter nur für Oracle.

### Bemerkungen

Diese Systemprozedur ruft nur Ereignisse in Bezug auf eine bestimmte Ausführung einer Aufgabe ab. Sie ist eine Alternative zur Systemprozedur `ml_ra_get_agent_events`.

Sie können einen NULL-Wert für die Ausführungsnummer übergeben, um die letzte Ausführung einer Aufgabe abzurufen.

### Siehe auch

- „[ml\\_ra\\_get\\_agent\\_events-Systemprozedur](#)“ auf Seite 684

### Beispiel

Eine Möglichkeit, diese Prozedur einzusetzen, besteht darin, mit `ml_ra_get_agent_events` auf ein Aufgabenende-Ereignis zu warten und anschließend `ml_ra_get_task_results` aufzurufen, um die einzelnen Befehlsergebnisse abzurufen, die möglicherweise verarbeitet werden müssen.

## ml\_ra\_get\_task\_status-Systemprozedur

Verwenden Sie diese Prozedur, um den Status der Aufgaben zu prüfen.

### Parameter

Syntax	Beschreibung
agent_id	VARCHAR(128). Dies ist ein IN-Parameter, der die ID des Agenten angibt, für den Sie Statusinformationen abrufen wollen.

Syntax	Beschreibung
task_name	VARCHAR(128). Dies ist ein IN-Parameter, der den Namen der Aufgabe angibt, für die Sie Statusinformationen abrufen wollen.

## Rückgabe

Ergebnis	Beschreibung
agent_id	VARCHAR(128). Die ID des Agenten, der dieses Ereignis erzeugt hat.
remote_id	VARCHAR(128). Die ID der entfernten Datenbank, für die das Ereignis gilt.
task_name	VARCHAR(128). Der Name der Aufgabe.
task_id	BIGINT. Die Aufgaben-ID.
state	<p>VARCHAR(4). Der Status der bereitgestellten Aufgabe. Der Status kann einer der folgenden Werte sein:</p> <ul style="list-style-type: none"> <li>• <b>P</b> Offen. Es wird auf die Bestätigung gewartet, dass der Agent die Aufgabe empfangen hat.</li> <li>• <b>A</b> Aktiv. Der Agent hat die Aufgabe und wird sie zur geplanten Zeit ausführen.</li> <li>• <b>S</b> Erfolgreich. Die Aufgabe ist abgeschlossen und wird nicht erneut ausgeführt, es sei denn, sie wird neu zugewiesen.</li> <li>• <b>F</b> Fehlgeschlagen. Die Aufgabe ist abgeschlossen und wird nicht erneut ausgeführt, es sei denn, sie wird neu zugewiesen.</li> <li>• <b>CP</b> Laufende Aufgaben abbrechen. Es wird auf die Bestätigung gewartet, dass der Agent die Aufgabe abgebrochen hat.</li> <li>• <b>C</b> Abgebrochen. Die Aufgabe ist abgeschlossen und wird nicht erneut ausgeführt, es sei denn, sie wird neu zugewiesen.</li> <li>• <b>E</b> Abgelaufen. Die Aufgabe ist abgeschlossen und wird nicht erneut ausgeführt, es sei denn, sie wird neu zugewiesen.</li> </ul>
reported_exec_count	BIGINT. Die gemeldete Anzahl ausgeführter Aufgaben.
reported_error_count	BIGINT. Die gemeldete Anzahl an Fehlern.
reported_attempt_count	BIGINT. Die gemeldete Anzahl an Versuchen zum Ausführen einer Ausgabe.

Ergebnis	Beschreibung
last_status_update	TIMESTAMP. Die Uhrzeit der letzten Statusaktualisierung.
last_success	TIMESTAMP. Die Uhrzeit der letzten erfolgreichen Aufgabe.
assignment_time	TIMESTAMP. Die Uhrzeit, zu der die Aufgabe zugewiesen wurde.
p_crsr	SYS_REF_CURSOR. Dies ist ein OUT-Parameter nur für Oracle.

### Bemerkungen

Die Parameter @agent\_id und @task\_name können auf NULL gesetzt werden, um den Status für alle Agenten-IDs, alle Aufgabennamen oder beides abzurufen.

Der reported\_attempt\_count-Wert ist möglicherweise größer als reported\_exec\_count, d.h., die Auswertung der Vorbedingung für die Aufgabe hat bei einem Versuch den Wert FALSE geliefert und die Aufgabe wurde nicht ausgeführt.

Sie können die Anzahl der Erfolge berechnen, indem Sie reported\_error\_count von reported\_exec\_count subtrahieren.

### Siehe auch

- „ml\_ra\_get\_task\_results-Systemprozedur“ auf Seite 690

## ml\_ra\_manage\_remote\_db-Systemprozedur

Verwenden Sie diese Prozedur, um eine agentenverwaltete entfernte Datenbank hinzuzufügen.

### Parameter

Syntax	Beschreibung
agent_id	VARCHAR(128). Dies ist ein IN-Parameter, der die ID des in der konsolidierten Datenbank zu definierenden neuen Agenten angibt.
schema_name	VARCHAR(128). Dies ist ein IN-Parameter, der den Typ der zu erstellenden entfernten Datenbank angibt. Dieser Schemaname muss bereits über Sybase Central in der konsolidierten Datenbank definiert worden sein.
conn_str	VARCHAR(128). Dies ist ein IN-Parameter, der die Datenbank-Verbindungszeichenfolge angibt, die vom Agenten für die Verbindung mit der entfernten Datenbank verwendet wird.

### Siehe auch

- „ml\_ra\_add\_agent\_id-Systemprozedur“ auf Seite 679
- „ml\_ra\_clone\_agent\_properties-Systemprozedur“ auf Seite 681

## ml\_ra\_notify\_agent\_sync-Systemprozedur

Verwenden Sie diese Prozedur, um einen Agenten zur Synchronisation seines Status zu veranlassen.

### Parameter

Syntax	Beschreibung
agent_id	VARCHAR(128). Dies ist ein IN-Parameter, der die ID des Agenten angibt, den Sie synchronisieren wollen.

### Bemerkungen

Diese Systemprozedur sendet neue Aufgaben an den angegebenen Agenten und veranlasst diesen, alle Ergebnisse aus der Ausführung von Aufgaben an den MobiLink-Server zu senden.

### Siehe auch

- [„ml\\_ra\\_get\\_task\\_results-Systemprozedur“](#) auf Seite 690

## ml\_ra\_notify\_task-Systemprozedur

Verwenden Sie diese Prozedur, um eine Aufgabe mithilfe vom Server eingeleiteter entfernter Aufgaben auszuführen.

### Parameter

Syntax	Beschreibung
agent_id	VARCHAR(128). Dies ist ein IN-Parameter, der die ID des Agenten angibt, der die Aufgabe ausführen soll.
task_name	VARCHAR(128). Dies ist ein IN-Parameter, der den Namen der Aufgabe angibt, die der Agent ausführen soll.

### Siehe auch

- [„ml\\_ra\\_cancel\\_notification-Systemprozedur“](#) auf Seite 680
- [„ml\\_ra\\_delete\\_task-Systemprozedur“](#) auf Seite 683

## ml\_ra\_reassign\_taskdb-Systemprozedur

Verwenden Sie diese Prozedur, um eine Agenten-Datenbank neu zuzuweisen, wenn eine verwaiste Agenten-Datenbank vorliegt.

**Parameter**

Syntax	Beschreibung
taskdb_remote_id	VARCHAR(128). Dies ist ein IN-Parameter, der die entfernte ID der verwaisten Agenten-Datenbank angibt.
new_agent_id	VARCHAR(128). Dies ist ein IN-Parameter, der die ID des neuen Agenten angibt, dem Sie die verwaiste Agenten-Datenbank zuweisen möchten.

**Bemerkungen**

Wenn es zwei Agenten-Datenbanken gibt, die beide dieselbe Agenten-ID verwenden wollen, betrachtet das System die erste Agenten-Datenbank als die gültige und die zweite Agenten-Datenbank als Waisen, d.h. ihr ist keine gültige Agenten-ID zugewiesen.

**Siehe auch**

- [„ml\\_ra\\_get\\_orphan\\_taskdbs-Systemprozedur“ auf Seite 688](#)

## ml\_ra\_set\_agent\_property-Systemprozedur

Verwenden Sie diese Prozedur, um Eigenschaften von entfernten Agenten festzulegen.

**Parameter**

Syntax	Beschreibung
agent_id	VARCHAR(128). Dies ist ein IN-Parameter, der die Agenten-ID angibt.
property_name	VARCHAR(128). Dies ist ein IN-Parameter, der den Namen der festzulegenden Eigenschaft angibt.
property_value	VARCHAR(2048). Dies ist ein IN-Parameter, der den festzulegenden Eigenschaftswert angibt.

**Bemerkungen**

Der Agent unterstützt die folgenden Eigenschaften:

- **mlstream** Die MobiLink-Datenstromparameter, zum Beispiel `tcpip(host=localhost)`.
- **max\_taskdb\_sync\_interval** Die längste Zeit in Sekunden, die der Agent zwischen zwei Synchronisationen seiner Agenten-Datenbank warten soll.
- **lwp\_freq** Der Zeitraum zwischen Lightweight-Abrufen.

**Siehe auch**

- [„ml\\_ra\\_clone\\_agent\\_properties-Systemprozedur“ auf Seite 681](#)

# ml\_ra\_unmanage\_remote\_db-Systemprozedur

Verwenden Sie diese Prozedur, um eine entfernten Datenbank definiert zu lassen, aber die Verbindung zwischen der entfernten Datenbank und einen entferntem Agenten zu trennen, sodass die Datenbank nicht mehr von diesem Agenten verwaltet wird.

**Parameter**

Syntax	Beschreibung
remote_id	VARCHAR(128). Dies ist ein IN-Parameter, der die abzutrennende entfernte ID angibt.
schema_name	VARCHAR(128). Dies ist ein IN-Parameter, der den Typ der entfernten Datenbank angibt.

**Bemerkungen**

Diese Prozedur schlägt fehl, wenn der entfernten Datenbank Aufgaben zugewiesen sind.

Wenn Sie wollen, dass die entfernte Datenbank von einem anderen Agenten verwaltet wird, können Sie die Prozedur ml\_ra\_manage\_remote\_db mit einer neuen Agenten-ID erneut aufrufen.

**Siehe auch**

- [„ml\\_ra\\_manage\\_remote\\_db-Systemprozedur“ auf Seite 693](#)

# ml\_reset\_sync\_state-Systemprozedur

Mit dieser Prozedur legen Sie die Informationen zum Synchronisationsstatus in MobiLink-Systemtabellen fest.

**Syntax**

```
ml_reset_sync_state (  
  'user',  
  'remote_id'  
)
```

**Parameter**

Syntax	Beschreibung
user	VARCHAR(128). Der MobiLink-Benutzername.
remote_id	VARCHAR(128). Die entfernte ID.

**Bemerkungen**

Diese Parameter können den Wert NULL haben. Wenn beide Parameter NULL sind, führt diese Prozedur keine Aktion aus.

Diese gespeicherte Prozedur setzt die Spalten `progress`, `last_upload_time` und `last_download_time` in der Tabelle `ml_subscription` auf die Standardwerte für den gegebenen Benutzernamen und die entfernte ID. Der Standardwert für die Spalte `progress` ist 0. Die Standardwerte für die Spalten `last_upload_time` und `last_download_time` lautet "1900/01/01 00:00:00".

Wenn die entfernte ID den Wert NULL hat und der MobiLink-Benutzername nicht NULL lautet, setzt diese Prozedur diese Spalten auf die Standardwerte für die Zeilen in der Tabelle `ml_subscription`, die vom angegebenen MobiLink-Benutzernamen referenziert werden. Wenn der MobiLink-Benutzername NULL lautet und die entfernte ID nicht den Wert NULL hat, setzt sie sie auf die Standardwerte für die Zeilen mit der angegebenen entfernten ID in der Tabelle `ml_subscription`.

Verwenden Sie diese gespeicherte Prozedur mit großer Umsicht. Der MobiLink-Server führt keine Prüfung des Synchronisationsstatus für diese entfernte ID aus, wenn der MobiLink-Client das nächste Mal die Synchronisation für diese entfernte ID anfordert. Dies kann Dateninkonsistenz verursachen, wenn eine entfernte ID zurückgesetzt wird, die beim letzten Synchronisationsversuch nicht erfolgreich synchronisiert werden konnte.

## **ml\_server\_delete-Systemprozedur**

Diese Prozedur ist nur für den internen Gebrauch gedacht.

## **ml\_server\_update-Systemprozedur**

Diese Prozedur ist nur für den internen Gebrauch gedacht.

# **MobiLink-Dienstprogramme**

Der MobiLink-Server umfasst die folgenden Dienstprogramme:

- „Dienstprogramm zum Stoppen von MobiLink (`mlstop`)“
- „Dienstprogramm für die MobiLink-Benutzerauthentifizierung (`mluser`)“
- „Dienstprogramm MobiLink Replay (`mlreplay`)“
- „MobiLink-generiertes Replay-API-Dienstprogramm (`mlgenreplayapi`)“
- „MobiLink-Arbiterserver-Dienstprogramm für Windows (`mlarbiter`)“
- „MobiLink-Arbiterserver-Dienstprogramm für Unix (`mlarbiter.sh`)“
- „Dienstprogramm zum MobiLink-Arbiters-Stop (`mlarbstop`)“ auf Seite 713

### **Siehe auch**

- „Dienstprogramme für MobiLink-Clients“ [*MobiLink - Clientadministration*]
- „UltraLite-Dienstprogramme“ [*UltraLite - Datenbankverwaltung*]
- „Zertifikat-Dienstprogramme“ [*SQL Anywhere Server - Datenbankadministration*]
- „Dienstprogramme für die Datenbankadministration“ [*SQL Anywhere Server - Datenbankadministration*]

## Dienstprogramm zum Stoppen von MobiLink (mlstop)

Stoppt den MobiLink-Server auf dem lokalen Computer.

### Syntax

**mlstop** [ *options* ] [ *name* ]

Option	Beschreibung
<i>@data</i>	<p>Liest Optionen aus der angegebenen Umgebungsvariablen oder Konfigurationsdatei ein. Wenn beide mit demselben Namen vorhanden sind, wird die Umgebungsvariable verwendet. Siehe „Konfigurationsdateien“ [<a href="#">SQL Anywhere Server - Datenbankadministration</a>].</p> <p>Wenn Sie Kennwörter oder andere Informationen in einer Konfigurationsdatei schützen möchten, können Sie das Dienstprogramm zum Ausblenden von Dateien zum Verschleiern des Inhalts von Konfigurationsdateien verwenden. Siehe „Dienstprogramm zum Verschleiern von Dateien (dbfhide)“ [<a href="#">SQL Anywhere Server - Datenbankadministration</a>].</p>
<b>-h</b>	Sofortiges Herunterfahren. MobiLink stoppt alle Synchronisationen und fährt herunter. Einige entfernte Datenbanken melden möglicherweise einen Fehler.
<b>-q</b>	Dialogfreier Modus. Hiermit wird das Banner unterdrückt.
<b>-t</b> <i>time</i>	Sanftes Herunterfahren, mit einem sofortigen Herunterfahren nach der angegebenen Zeitdauer. <i>time</i> ist eine Zahl, gefolgt von T, H, M, oder S (Tage, Stunden, Minuten und Sekunden). Beispiel: <code>-t 10m</code> legt fest, dass der Server in 10 Minuten heruntergefahren werden soll oder nachdem die aktuellen Synchronisationen abgeschlossen sind, je nachdem, was früher eintritt. "D", "H", "M" und "S" können in Groß- oder auch Kleinbuchstaben angegeben werden.
<b>-w</b>	Wartet, dass der MobiLink-Server herunterfährt, bevor eine Rückkehr vom Befehl erfolgt.
<i>name</i>	Wenn der MobiLink-Server mit der Option <code>-zs</code> gestartet wird, muss er unter Angabe des Servernamens heruntergefahren werden. Siehe „ <a href="#">mlsrv16-Option -zs</a> “ auf Seite 103.

### Bemerkungen

Standardmäßig (wenn weder `-h` noch `-t` angegeben sind) führt `mlstop` ein sanftes Herunterfahren durch.

- **Sanftes Herunterfahren** Der MobiLink-Server akzeptiert keine weiteren Verbindungen mehr und beendet das Programm, wenn die laufenden Synchronisationen abgeschlossen sind.
- **Sofortiges Herunterfahren** Der MobiLink-Server stoppt alle Synchronisationen und fährt dann herunter. Einige entfernte Datenbanken melden möglicherweise einen Fehler.

## Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser)

Registriert MobiLink-Benutzer der konsolidierten Datenbank. Bei entfernten SQL Anywhere-Datenbanken müssen Benutzer zuvor auf den entfernten Datenbanken mit der Anweisung CREATE SYNCHRONIZATION USER erstellt werden.

### Syntax

```
mluser [ options ] -c "connection-string"
{ -f file | -u user [ -p password ] }
```

Option	Beschreibung
@data	<p>Liest Optionen aus der angegebenen Umgebungsvariablen oder Konfigurationsdatei ein. Wenn beide mit demselben Namen vorhanden sind, wird die Umgebungsvariable verwendet. Siehe „Konfigurationsdateien“ [<a href="#">SQL Anywhere Server - Datenbankadministration</a>].</p> <p>Wenn Sie Kennwörter oder andere Informationen in einer Konfigurationsdatei schützen möchten, können Sie das Dienstprogramm zum Ausblenden von Dateien zum Verschleiern des Inhalts von Konfigurationsdateien verwenden. Siehe „Dienstprogramm zum Verschleiern von Dateien (dbfhide)“ [<a href="#">SQL Anywhere Server - Datenbankadministration</a>].</p>
-c "keyword=value;..."	Hiermit können Sie Datenbank-Verbindungsparameter bereitstellen. Die Verbindungszeichenfolge muss dem Dienstprogramm ausreichende Privilegien erteilen, damit es eine Verbindung mit der konsolidierten Datenbank über eine ODBC-Datenquelle herstellen kann. Dieser Parameter ist erforderlich.
-d	Löscht den bzw. die durch -f oder -u angegebenen Benutzernamen. Diese Option kann nicht zusammen mit der mluser-Option -r verwendet werden.

Option	Beschreibung
<b>-f</b> <i>filename</i>	Liest Benutzernamen und Kennwörter aus der angegebenen Datei aus. Bei der Datei sollte es sich um eine Textdatei handeln, die auf jeder Zeile einen Benutzernamen und ein Kennwort getrennt durch ein Leerzeichen enthält. Sie müssen -f oder -u angeben.
<b>-fips</b>	Falls aktiviert, schlägt mluser fehl, wenn die Unterstützung für FIPS-zertifizierte Verschlüsselung nicht installiert ist.
<b>-n</b> <i>Name_Benutzerauthentifizierungsrichtlinie</i>	Registriert einen MobiLink-Benutzer bei der LDAP-Benutzerauthentifizierung.
<b>-o</b> <i>Dateiname</i>	Protokolliert Ausgabenachrichten in die festgelegte Datei.
<b>-ot</b> <i>Dateiname</i>	Logdatei kürzen und anschließend Ausgabemeldungen anfügen. Standardmäßig werden Ausgaben auf dem Bildschirm angezeigt.
<b>-pc</b> <i>collation-id</i>	<p>Gibt eine Datenbankkollations-ID für die Zeichensatzkonvertierung von Benutzernamen und Kennwort an. Dabei muss es sich um eine der SQL Anywhere-Kollationslabels handeln, wie unter „<a href="#">Alternative Kollationen</a>“ [<a href="#">SQL Anywhere Server - Datenbankadministration</a>] aufgelistet.</p> <p>Diese Option ist notwendig, wenn Benutzernamen und Kennwörter aus einer Datei gelesen werden, die mit einem anderen Zeichensatz kodiert ist als von der Sprachumgebung festgelegt wird.</p>
<b>-p</b> <i>password</i>	Kennwort, das dem Benutzer zugeordnet werden soll. Diese Option kann nur mit -u eingesetzt werden.

Option	Beschreibung
<b>-r</b> <i>remote-id</i>	<p>Wenn Sie diese Option zusammen mit <i>-u Benutzername</i> verwenden, setzt mluser den Synchronisationsstatus für die angegebene entfernte ID und den Benutzernamen zurück. Die Spalten <i>last_upload_time</i> und <i>last_download_time</i> in der <i>ml_subscription</i>-Tabelle werden für den angegebenen Benutzernamen und die entfernte ID auf ihre Standardwerte zurückgesetzt. Die Standardwerte für die Spalten <i>progress</i>, <i>last_upload_time</i> und <i>last_download_time</i> lauten 0, "1900/01/01 00: 00: 00" bzw. "1900/01/01 00: 00: 00".</p> <p>Diese Option kann nicht zusammen mit der mluser-Option <i>-d</i> verwendet werden.</p> <div style="border: 1px solid black; padding: 5px;"> <p><b>Vorsicht</b> Diese Option setzt die Informationen zum Synchronisationsstatus für den angegebenen Benutzernamen und die entfernte ID zurück. Diese Aktion kann nicht rückgängig gemacht werden. Nach dem Zurücksetzen des Synchronisationsstatus akzeptiert der MobiLink-Server immer die erste vom Client gesendete Synchronisationsanforderung, ohne den letzten Synchronisationsstatus zu überprüfen.</p> </div>
<b>-u</b> <i>username</i>	Gibt hinzuzufügenden (bzw. zu löschenden, falls mit <i>-d</i> benutzt) Benutzernamen an. In einer Befehlszeile kann jeweils nur ein Benutzer angegeben werden. Diese Option wird mit <i>-p</i> benutzt, wenn Kennwörter festgelegt wurden. Sie müssen <i>-f</i> oder <i>-u</i> angeben.
<b>-v</b>	Legt die ausführliche Protokollierung fest.

## Bemerkungen

Bei einem bestimmten Benutzer/Kennwort-Paar versucht das Dienstprogramm mluser zunächst, den Benutzer hinzuzufügen. Wenn der Benutzer bereits zur konsolidierten Datenbank hinzugefügt wurde, versucht das Programm, das Kennwort für den betreffenden Benutzer zu aktualisieren.

Es gibt verschiedene Möglichkeiten, Benutzernamen in der konsolidierten Datenbank zu registrieren:

- Verwenden Sie Sybase Central.
- Geben Sie bei mlsrv16 die Befehlszeilenoption *-zu+* an. In diesem Fall werden alle bestehenden MobiLink-Benutzer, die der konsolidierten Datenbank noch nicht hinzugefügt wurden, bei der ersten Synchronisation hinzugefügt.

Der MobiLink-Benutzer muss bereits in einer entfernten Datenbank existieren. Um Benutzer in die entfernte Datenbank einzufügen, benutzen Sie folgende Optionen:

- Bei entfernten SQL Anywhere-Datenbanken legen Sie den Namen mit CREATE SYNCHRONIZATION USER fest und synchronisieren mit diesem Benutzernamen.
- Für entfernte UltraLite-Datenbanken können Sie vor der Synchronisation entweder das Feld "user\_name" der Struktur ul\_sync\_info oder in Java die Methode SetUserName() der Klasse ULSynchInfo verwenden.

**Siehe auch**

- „MobiLink-Benutzer“ [*MobiLink - Clientadministration*]
- „mlsrv16-Option -zu “ auf Seite 104
- „CREATE SYNCHRONIZATION USER-Anweisung [MobiLink]“ [*SQL Anywhere Server - SQL-Referenzhandbuch*]
- „Transportschichtssicherheit“ [*SQL Anywhere Server - Datenbankadministration*]

# Dienstprogramm MobiLink Replay (mlreplay)

Das mlreplay-Dienstprogramm ist ein Tool, mit dem die vom MobiLink-Server aufgezeichneten MobiLink-Protokollinformationen wiedergegeben werden können.

**Syntax**

**mlreplay** [options] [name=value [name2=value2...]] [[dll\_name] filename]

Option	Beschreibung
@data	<p>Liest Optionen aus der angegebenen Umgebungsvariablen oder Konfigurationsdatei ein. Wenn beide mit demselben Namen vorhanden sind, wird die Umgebungsvariable verwendet. Siehe „Konfigurationsdateien“ [<i>SQL Anywhere Server - Datenbankadministration</i>].</p> <p>Wenn Sie Kennwörter oder andere Informationen in einer Konfigurationsdatei schützen möchten, können Sie das Dienstprogramm zum Ausblenden von Dateien zum Verschleiern des Inhalts von Konfigurationsdateien verwenden. Siehe „Dienstprogramm zum Verschleiern von Dateien (dbfhide)“ [<i>SQL Anywhere Server - Datenbankadministration</i>].</p>
-ap	<p>Passt den Fortschritt der in einer Wiedergabesitzung wiedergegebenen Synchronisationen so an, dass das Dienstprogramm mlreplay keine Warnungen wegen nicht übereinstimmenden Offsets auf dem MobiLink-Server verursacht, und passt Sequenznummern so an, dass diese keine Fehler verursachen.</p>

Option	Beschreibung
<b>-f</b> <i>time_scale_factor</i>	Auf die aufgezeichneten Zeiten gleichmäßig angewendeter Multiplikator.
<b>-ldt</b> <i>last_download_time</i>	Geben Sie die während der Wiedergabesitzung an MobiLink-Server zu sendende letzte Downloadzeit an. Wenn das gerade wiedergegebene aufgezeichnete Protokoll mehrere Synchronisationen enthält (was möglich ist, wenn eine beständige Verbindung aufgezeichnet wurde), wird nur die erste letzte Downloadzeit ersetzt. Der Rest wird durch die letzte Downloadzeit ersetzt, die der MobiLink-Server während der Wiedergabesitzung an mlreplay sendet. Auch wenn die Option -ldt nicht verwendet wird, ersetzt mlreplay in allen Synchronisationen außer der ersten die letzte Downloadzeit durch die während der Wiedergabesitzung vom MobiLink-Server empfangene letzte Downloadzeit. Eine letzte Downloadzeit kann auch mithilfe der Informationsdatei für simulierte Clients angegeben werden (wenn die Option -sci verwendet wird), oder über den IdentifySimulatedClient-Callback, wenn eine DLL bereitgestellt wird.
<b>-ls</b>	Die gesamte Laufzeit, die Gesamtdauer der Wiedergabe sowie für jeden simulierten Client die Gesamtzahl der Wiederholungen, die erfolgreich abgeschlossen wurden, fehlgeschlagen sind oder übersprungen wurden, werden protokolliert. Diese Informationen werden von mlreplay weiterhin vor dem Beenden protokolliert, auch wenn diese Option nicht angegeben ist.
<b>-n</b> <i>number_of_simulated_clients</i>	Die Anzahl auszuführender simulierter Clients. Das Minimum ist 1.  Diese Option kann mit der Option -sci verwendet werden, wenn die Anzahl der simulierten Clients, die durch die Option -n festgelegt wird, kleiner oder gleich der Anzahl der simulierten Clients in der Informationsdatei ist. Wenn die Optionen gemeinsam verwendet werden, gibt -n die Anzahl der ausgeführten simulierten Clients an. Diese Optionen ermöglichen es, mithilfe einer Informationsdatei, die <i>x</i> als Anzahl der simulierten Clients angibt, ein Protokoll mit 1 bis <i>x</i> simulierten Clients wiederzugeben.

Option	Beschreibung
<b>-o</b> <i>file</i>	Befehlszeilenoptionen und Ausgabemeldungen werden in der festgelegten Datei protokolliert.
<b>-os</b> <i>size</i>	Die maximale Größe der Logdatei wird festgelegt. Wenn das Log die angegebene Größe erreicht (mindestens 10 kB) wird es in <i>JJMMTTxx.rlg</i> umbenannt und eine neue Datei mit dem ursprünglichen Namen wird gestartet.
<b>-ot</b> <i>file</i>	Die Logdatei wird gekürzt. Befehlszeilenoptionen und Ausgabemeldungen werden in der festgelegten Datei protokolliert.
<b>-p</b> <i>password</i>	Ersetzt Kennwörter durch das angegebene Kennwort.
<b>-ping</b> <i>seconds</i>	<p>Pingt einen MobiLink-Server an, um zu ermitteln, ob der Server für den Empfang von Synchronisationen bereit ist. Standardmäßig pingt mlreplay den Server 60 Sekunden lang an.</p> <p>Wenn die Option -ping verwendet wird, sind die folgenden Rückgabecodes gültig:</p> <ul style="list-style-type: none"> <li>• <b>-1</b> Zeigt an, dass ein Fehler aufgetreten ist.</li> <li>• <b>0</b> Gibt an, dass mlreplay den Server anpingen konnte und dass der Server bereit ist, Synchronisationen zu empfangen.</li> <li>• <b>1</b> Gibt an, dass mlreplay versucht hat, den Server anzupingen, aber keine Antwort erhielt, und dass der Server folglich nicht bereit ist, Synchronisationen zu empfangen.</li> </ul>
<b>-r</b> <i>remote ID</i>	Entfernte IDs durch angegebene entfernte ID ersetzen. Diese Option kann nicht mit der Option -rg verwendet werden.

Option	Beschreibung
<b>-rep</b> <i>number_of_repetitions</i>	Geben Sie an, wie oft simulierte Clients das aufgezeichnete Protokoll wiedergeben sollen. Jede Wiederholung kann angepasst werden, wenn für die Wiedergabe eine DLL bzw. ein Shared Object verwendet wird. Wenn Sie die generierte Replay-API verwenden, werden für jede Wiederholung die Callback-Funktionen "GetUploadTransaction", "GetDownloadApplyTime", "ReportEndOfReplay" und "DelayStartOfReplay" aufgerufen.
<b>-rg</b>	Ersetzt entfernte IDs durch eine GUID.
<b>-rnt</b> <i>seconds</i>	<p>Die simulierten Clients werden angewiesen, neue Wiederholungen der Protokollwiedergabe zu starten, bis die angegebene Anzahl von Sekunden erreicht ist. Simulierte Clients werden zwar nicht gestoppt, aber es werden keine zusätzlichen Wiederholungen gestartet.</p> <p>Wenn angegeben, wird der <b>numRepetitions</b>-Parameter einer API-Callback-Funktion auf 0 gesetzt.</p>
<b>-rp</b> <i>pattern</i>	Das angegebene Muster wird in Benutzernamen, Kennwörtern und entfernten IDs in der Befehlszeile durch die simulierte Clientnummer ersetzt.

Option	Beschreibung
<b>-sci</b> <i>file</i>	<p>Geben Sie mlreplay eine Liste von Benutzernamen, Kennwörtern, entfernten IDs und letzten Downloadzeiten an, die für die Wiedergabe verwendet werden sollen. mlreplay erstellt für jede Zeile der Datei einen simulierten Client, um die Synchronisation mit diesen Clientinformationen wiederzugeben. Das Format jeder Zeile muss wie folgt lauten: [Benutzername],[Kennwort],[Entfernte ID],[Letzte Downloadzeit],[Skriptversion]. Die Zeit des letzten Downloads sollte in folgendem Format angegeben werden: JJJJ-MM-TT hh:mm:ss.SSS. Wenn die Felder für Benutzername, Kennwort, letzte Downloadzeit oder Skriptversion leer gelassen werden, verwendet mlreplay die entsprechenden Werte aus dem aufgezeichneten Protokoll. Wenn die entfernte ID leer ist, ersetzt mlreplay die entfernte ID durch eine GUID. Weder die Optionen -u, -p, -r, -rg und -ldt noch eine DLL können zusammen mit dieser Option verwendet werden.</p> <p>Diese Option kann zusammen mit der Option -n verwendet werden, wenn die durch -n angegebene Anzahl von simulierten Clients höchstens gleich der Anzahl der simulierten Clients in der Informationsdatei ist. Wenn die Optionen gemeinsam verwendet werden, gibt -n die Anzahl der ausgeführten simulierten Clients an. Diese Optionen ermöglichen es, mithilfe einer Informationsdatei, die <i>x</i> als Anzahl der simulierten Clients angibt, ein Protokoll mit 1 bis <i>x</i> simulierten Clients wiederzugeben.</p>
<b>-sv</b> <i>script version</i>	Ersetzen Sie Skriptversionen durch die angegebene Skriptversion.
<b>-u</b> <i>user name</i>	Ersetzt Benutzernamen durch den angegebenen Benutzernamen.
<b>-x</b> <i>stream(opts)</i>	Der Protokolldatenstrom und die Datenstromoptionen für die Verbindung mit dem MobiLink-Server. Der Verfügbarkeits-Timeout kann mit dieser Option festgelegt werden und wird automatisch angepasst, basierend auf dem vom MobiLink-Server verwendeten Wert.

## Bemerkungen

Der optionale `dll_name`-Parameter ist der Name der Replay-DLL, die `mlreplay` verwenden soll. Die Replay-DLL wird aus der Replay-API kompiliert. Siehe „[MobiLink-generiertes Replay-API-Dienstprogramm \(mlgenreplayapi\)](#)“ auf Seite 710.

Die `Name=Wert`-Paare fungieren als Befehlszeilenargumente für die Replay-API. Sie sind in allen `mlreplay`-Callback-Funktionen zugänglich und können verwendet werden, um das Verhalten der Replay-DLL anzupassen. Sie werden nur verwendet, wenn eine Replay-DLL verwendet wird. Wenn Sie beispielsweise dieselbe Replay-DLL verwenden möchten, um Synchronisationen in unterschiedlichen Datenbanken (mit verschiedenen Instanzen von `mlreplay`) auszuführen, und am Ende der Synchronisationen eine Verbindung mit der Datenbank herstellen möchten, um sich zu vergewissern, dass die Daten erfolgreich hochgeladen wurden, können Sie die Verbindungszeichenfolge für die Datenbank mithilfe eines `Name=Wert`-Paares angeben, statt sie in der Replay-DLL fest zu kodieren.

Jede aufgezeichnete Datei wird als **aufgezeichnete Protokolldatei** bezeichnet. Alle von Anfang bis Ende einer Verbindung empfangenen Daten werden in einer separaten Protokolldatei aufgezeichnet. Jede aufgezeichnete Protokolldatei heißt *recorded\_protocol\_x.mlr*. Dabei ist *x* die Auftrags-ID. Mit der MobiLink-Serveroption `-rp` wird festgelegt, dass der MobiLink-Server alle von Clients empfangenen MobiLink-Protokollinformationen aufzeichnen soll. Siehe „[mlsrv16-Option -rp](#)“ auf Seite 77.

Zusätzlich zu den Daten, die vom und an den MobiLink-Server gesendet wurden, enthält die aufgezeichnete Protokolldatei auch Zeitinformationen. Damit kann `mlreplay` die aufgezeichneten Protokollinformationen genauso wiedergeben wie sie ursprünglich ausgeführt wurden. Anhand der Zeitinformationen wird auch versucht, dass der simulierte Client genauso lange benötigt wie der ursprüngliche Client.

Standardmäßig wird die aufgezeichnete Protokolldatei von `mlreplay` ohne Änderungen wiedergegeben. Sie können die Wiedergabesitzung jedoch mithilfe verschiedener Optionen anpassen. Die Informationen zum simulierten Client bestehen aus Benutzername, Kennwort, entfernter ID letzter Downloadzeit und Skriptversion. Diese Informationen können mit den Optionen `-u`, `-p`, `-r` (oder `-rg`) `-ldt` und `-sv` angepasst werden.

Das Dienstprogramm `mlreplay` kann eine aufgezeichnete Protokolldatei mit verschiedenen simulierten Clients gleichzeitig wiedergeben. Sie haben hierzu drei Möglichkeiten:

- **Nur mit der Befehlszeile** Sie können eine aufgezeichnete Protokolldatei gleichzeitig wiedergeben, indem Sie eine Kombination der Optionen `-n`, `-u`, `-p`, `-sv`, `-r`, `-rg` und `-rp` verwenden. Mit der Option `-n` wird die Anzahl der simulierten Clients angegeben, während `-u`, `-p`, `-sv`, `-r` und `-rg` dazu dienen, Informationen zu den einzelnen Clients anzugeben. Standardmäßig können Sie ein Sternchen (\*) angeben, wenn Sie `-u`, `-p`, `-sv` bzw. `-r` verwenden (beliebig oft), damit `mlreplay` das Sternchen durch die Nummer des jeweiligen simulierten Clients ersetzt. Sie können das Sternchen mithilfe der Option `-rp` in jedes andere Zeichen ändern.

Beispiel: `mlreplay -ap -x tcpip -n 2 -rp $ -u user_$ -p pwd_$ -r rid_$ -sv test_script recorded_protocol.mlr` führt `mlreplay` mit zwei simulierten Clients aus. Der simulierte Client 1 enthält folgende Informationen:

- Benutzer: `user_1`
- Kennwort: `pwd_1`
- Entfernte ID: `rid_1`
- Skriptversion: `test_script`

Der simulierte Client 2 enthält folgende Informationen:

- Benutzer: `user_2`
- Kennwort: `pwd_2`
- Entfernte ID: `rid_2`
- Skriptversion: `test_script`

Die folgenden Regeln werden angewendet, wenn Optionen ausgelassen wurden:

- Wenn Benutzername, Kennwort oder Skriptversion nicht angegeben wird, verwenden die simulierten Clients für die fehlende Angabe den in der aufgezeichneten Protokolldatei erfassten Wert.
  - Wenn keine entfernte ID angegeben wurde und die Anzahl der simulierten Clients größer ist als 1, wird automatisch eine andere GUID für jede entfernte ID generiert. Wenn die Anzahl der simulierten Clients gleich 1 ist, wird die in der aufgezeichneten Protokolldatei erfasste entfernte ID verwendet. Mithilfe der Option `-rg` können Sie das Generieren einer GUID erzwingen.
  - Wenn Benutzername, Kennwort, entfernte ID oder Skriptversion angegeben wird, aber kein Sternchen (\*) enthält (oder nicht das durch die Option `-rp` festgelegte Zeichen enthält), verwendet jeder simulierte Client dieselben Werte für Benutzername, Kennwort, entfernte ID bzw. Skriptversion.
- **Mit einer Informationsdatei für simulierte Clients** Sie können eine aufgezeichnete Protokolldatei gleichzeitig wiedergeben, indem Sie eine Informationsdatei für simulierte Clients mithilfe der Option `-sci` angeben. Die Informationsdatei für simulierte Clients ist eine `.csv`-Datei, in der jede Zeile Benutzername, Kennwort, entfernte ID, letzte Downloadzeit und Skriptversion (in dieser Reihenfolge) enthält.

Das Dienstprogramm `mlreplay` füllt leere Felder nach denselben Regeln aus wie unter **Using only the command line** beschrieben.

Standardmäßig erstellt `mlreplay` einen simulierten Client für jede Zeile aus der Informationsdatei für simulierte Clients. Sie können jedoch die Option `-n` zusammen mit der Option `-sci` verwenden, um die Anzahl der simulierten Clients zu begrenzen. Wenn Ihre Informationsdatei für simulierte Clients `x` simulierte Clients angibt, können Sie mithilfe der Option `-n` eine Zahl von 1 bis `x` angeben, damit `mlreplay` nur diese Anzahl von simulierten Clients verwendet.

Die Verwendung einer Informationsdatei für simulierte Clients ist flexibler als das Arbeiten nur mit der Befehlszeile, aber weniger flexibel als die Verwendung einer Replay-DLL.

- **Mit einer Replay-DLL** Wenn Sie eine Replay-DLL verwenden möchten, geben Sie mithilfe der Option `-n` die Anzahl der simulierten Clients an. Alle anderen Informationen werden abgerufen, wenn

mlreplay Aufrufe in die vom Benutzer implementierten Callback-Funktionen ausführt. Dieser Ansatz bietet die größte Flexibilität, da weitere Teile der Wiedergabe angepasst werden können. Siehe [„MobiLink Replay-C++-Callback-Funktionen“ auf Seite 639](#) und [„MobiLink-generiertes Replay-API-Dienstprogramm \(mlgenreplayapi\)“ auf Seite 710](#).

Das mlreplay-Dienstprogramm kann mithilfe der Befehlszeile mehrere simulierte Clients ausführen, um ein Protokoll wiederzugeben. Die Anzahl auszuführender simulierter Clients kann mit der Option -n angegeben werden. Verwenden Sie das Sternchen-Zeichen, um die Nummer des simulierten Clients beim Angeben von Benutzernamen, Kennwörtern, entfernten IDs und Skriptversionen mit den Optionen -u, -p, -r bzw. -sv zu bezeichnen. Benutzername, Kennwort, entfernte ID und Skriptversion für jeden simulierten Client werden nach den folgenden Regeln ermittelt:

- Wenn kein Benutzername oder Kennwort angegeben ist, verwenden alle simulierten Clients den Benutzernamen bzw. das Kennwort aus der wiedergegebenen aufgezeichneten Protokolldatei.
- Wenn keine entfernte ID angegeben wurde und die Anzahl der simulierten Clients größer ist als 1, wird jede entfernte ID zu einer automatisch generierten GUID. Wenn die Anzahl der simulierten Clients gleich 1 ist, wird die entfernte ID aus der aufgezeichneten Protokolldatei verwendet. Mithilfe der Option -rg können Sie einen GUID-Wert erzwingen.
- Wenn die Angaben für Benutzername, Kennwort oder entfernte ID keine Sternchen enthalten, verwenden die simulierten Clients denselben Benutzername, dasselbe Kennwort bzw. dieselbe entfernte ID. Wenn die Angaben für Benutzername, Kennwort oder entfernte ID mindestens ein Sternchen enthalten, erhalten die simulierten Clients ihre eigenen eindeutigen Benutzernamen, Kennwörter und entfernten IDs, wobei jedes Sternchen durch die simulierte Clientnummer ersetzt wird.

Die für die ursprüngliche Synchronisation benötigte Zeit ist Teil der Aufzeichnung. mlreplay kann daher versuchen, die Synchronisation mit derselben Zeitdauer wiederzugeben.

Verwenden Sie das mlreplay-Dienstprogramm mit den folgenden MobiLink-Serveroptionen:

- **-rp** Geben Sie mit dieser Option das Verzeichnis an, in dem die Synchronisationen für die Wiedergabe mit dem mlreplay-Dienstprogramm aufgezeichnet wurden.
- **-rrp** Verwenden Sie diese Option, um das mlreplay- Dienstprogramm auszuführen, wenn der MobiLink-Server gestartet wird.
- **-lsc** Geben Sie mit dieser Option die Verbindungsinformationen für den lokalen Server an, damit das mlreplay-Dienstprogramm eine Verbindung zum Server herstellen kann.

Die Wiedergabesitzung kann mit dem von MobiLink generierten Replay-API-Dienstprogramm weiter angepasst werden.

#### Siehe auch

- [„MobiLink-generiertes Replay-API-Dienstprogramm \(mlgenreplayapi\)“ auf Seite 710](#)
- [„mlsrv16-Option -rrp“ auf Seite 76](#)
- [„mlsrv16-Option -rp“ auf Seite 77](#)
- [„mlsrv16-Option -lsc“ auf Seite 65](#)

## MobiLink-generiertes Replay-API-Dienstprogramm (mlgenreplayapi)

Das mlgenreplayapi-Tool liest eine aufgezeichnete Protokolldatei und generiert die **MobiLink-Replay-API** für das Schema dieser Datei.

### Syntax

**mlgenreplayapi** [*options*] *filename*

Option	Beschreibung
<i>@data</i>	<p>Liest Optionen aus der angegebenen Umgebungsvariablen oder Konfigurationsdatei ein. Wenn beide mit demselben Namen vorhanden sind, wird die Umgebungsvariable verwendet. Siehe „<a href="#">Konfigurationsdateien</a>“ [<i>SQL Anywhere Server - Datenbankadministration</i>].</p> <p>Wenn Sie Kennwörter oder andere Informationen in einer Konfigurationsdatei schützen möchten, können Sie das Dienstprogramm zum Ausblenden von Dateien zum Verschleiern des Inhalts von Konfigurationsdateien verwenden. Siehe „<a href="#">Dienstprogramm zum Verschleiern von Dateien (dbfhide)</a>“ [<i>SQL Anywhere Server - Datenbankadministration</i>].</p>
<b>-d</b> <i>directory</i>	Das Verzeichnis für die Ausgabe der erzeugten Dateien.
<b>-o</b> <i>file</i>	Protokolliert Befehlszeilenoptionen und Ausgabemeldungen in die festgelegte Datei.
<b>-oss</b> <i>size</i>	Legt die maximale Größe der Logdatei fest. Wenn das Log die angegebene Größe erreicht (mindestens 10 kB) wird es in <i>JJMMTTxx.rlg</i> umbenannt und eine neue Datei mit dem ursprünglichen Namen wird gestartet.
<b>-ot</b> <i>file</i>	Kürzt die Logdatei. Hängt Befehlszeilenoptionen und Ausgabemeldungen an die festgelegte Datei an. Standardmäßig werden Ausgaben auf dem Bildschirm angezeigt.

### Bemerkungen

Die Replay-API kann geändert werden (wobei nur der Code in *mlreplaycallbacks.cpp* geändert werden muss), um die während der Wiedergabesitzung auf den MobiLink-Server hochgeladenen Daten anzupassen. Die Replay-API kann dann in die **Replay-DLL** kompiliert werden, mit der mlreplay die Wiedergabesitzung anpasst. Die Replay-DLL und eine Informationsdatei für simulierte Clients können

nicht zur gleichen Zeit benutzt werden. Die Replay-API enthält einen Callback, der benutzt werden kann, um Informationen eines jeden simulierten Clients abzurufen. Die Anzahl der bei Verwendung der Replay-DLL zu startenden simulierten Clients wird beim Aufruf von mlreplay mit der Befehlszeilenoption -n angegeben.

**Siehe auch**

- „MobiLink Replay-C++-Callback-Funktionen“ auf Seite 639
- „Dienstprogramm MobiLink Replay (mlreplay)“ auf Seite 702

## MobiLink-Arbiterserver-Dienstprogramm für Windows (mlarbiter)

Der mlarbiter-Befehl startet den MobiLink-Arbiterserver.

**Syntax**

**mlarbiter**

**Bemerkungen**

Der MobiLink-Arbiterserver hört standardmäßig Port 4953 ab.

Der MobiLink-Arbiterserver stellt sicher, dass nur ein einziger MobiLink-Server in einer Serverfarm als Primärserver läuft, und verhindert so redundante Benachrichtigungen in einer Umgebung mit serverinitialisierter Synchronisation.

Dieser Befehl wird zusammen mit der MobiLink-Serveroption -ca verwendet, die dem MobiLink-Server den Hostnamen des Arbiters angibt.

Wenn der MobiLink-Server nach dem Starten des Arbiters keine Verbindung mit dem Arbiterserver herstellen kann, versucht der MobiLink-Server alle 15 Sekunden, die Verbindung herzustellen, und zeigt regelmäßig Fehlermeldungen an.

Wenn die Arbiterserververbindung gelöscht wird, nachdem die MobiLink-Server in der Serverfarm einen Primärserver gewählt haben, wird der Primärserver sofort heruntergefahren und die Sekundärserver versuchen alle 15 Sekunden, die Arbiterserververbindung wiederherzustellen. Wenn die Verbindung zum Arbiterserver hergestellt ist, wählen die MobiLink-Server erneut einen Primärserver.

**Beispiel**

Das folgende Beispiel beschreibt die Verwendung der MobiLink-Arbiterserver in einer MobiLink-Serverfarm.

1. Starten Sie den MobiLink-Arbiterserver auf einem Computer mit der folgenden Befehlszeile:

```
mlarbiter
```

2. Starten Sie die MobiLink-Server mit einer Befehlszeile wie im folgenden Beispiel. Die MobiLink-Server können auf demselben Computer wie der Arbiterserver oder auf anderen Computern gestartet werden.

```
mlsrv16 -c parameter1 -lsc parameter2 -ca Host_1 -notifier
```

Im oben gezeigten Beispiel ist *parameter1* der Verbindungsparameter für die konsolidierte Datenbank und *parameter2* ist der Verbindungsparameter für den lokalen MobiLink-Server. Alle MobiLink-Server derselben Serverfarm müssen dieselbe Einstellung für die Option -ca enthalten.

#### Siehe auch

- „mlsrv16-Option -ca“ auf Seite 53
- „mlsrv16-Option -lsc“ auf Seite 65
- Architektur [*MobiLink - Erste Orientierung*]

## MobiLink-Arbiterserver-Dienstprogramm für Unix (*mlarbiter.sh*)

Der Befehl *mlarbiter.sh* startet und stoppt den MobiLink-Arbiterserver.

#### Syntax

**mlarbiter.sh** [ *option* ]

Option	Beschreibung
<b>start</b>	Startet das MobiLink-Arbiterserver-Dienstprogramm.
<b>stop</b>	Stoppt das MobiLink-Arbiterserver-Dienstprogramm.

#### Bemerkungen

Der MobiLink-Arbiterserver hört standardmäßig Port 4953 ab.

Dieser Befehl wird zusammen mit der mlsrv16-Option -ca verwendet, die dem MobiLink-Server den Hostnamen des Arbiters angibt.

Der MobiLink-Arbiterserver stellt sicher, dass nur ein einziger MobiLink-Server in einer Serverfarm als Primärserver läuft. Dies verhindert redundante Benachrichtigungen in einer Umgebung mit serverinitialisierter Synchronisation.

Wenn der MobiLink-Server nach dem Starten des Arbiters keine Verbindung mit dem Arbiterserver herstellen kann, versucht der MobiLink-Server alle 15 Sekunden, die Verbindung herzustellen, und zeigt regelmäßig Fehlermeldungen an.

Wenn die Arbiterserver-Verbindung getrennt wird, nachdem die MobiLink-Server in der Serverfarm einen Primärserver gewählt haben, wird der Primärserver sofort heruntergefahren und die Sekundärserver versuchen alle 15 Sekunden, die Arbiterserver-Verbindung wiederherzustellen. Wenn die Verbindung zum Arbiterserver hergestellt ist, wählen die MobiLink-Server erneut einen Primärserver.

#### Beispiel

Das folgende Beispiel beschreibt die Verwendung der MobiLink-Arbiterserver in einer MobiLink-Serverfarm.

1. Starten Sie den MobiLink-Arbiter auf einem Computer mit der folgenden Befehlszeile:

```
mlarbiter.sh start
```

2. Starten Sie die MobiLink-Server mit einer Befehlszeile wie im folgenden Beispiel. Die MobiLink-Server können auf demselben Computer wie der Arbiterserver oder auf anderen Computern gestartet werden.

```
mlsrv16 -c parameter1 -lsc parameter2 -ca Host_1 -notifier
```

Im oben gezeigten Beispiel ist *parameter1* der Verbindungsparameter für die konsolidierte Datenbank und *parameter2* ist der Verbindungsparameter für den lokalen MobiLink-Server. Alle MobiLink-Server derselben Serverfarm müssen dieselbe Einstellung für die Option -ca enthalten.

### Siehe auch

- „mlsrv16-Option -ca“ auf Seite 53
- „mlsrv16-Option -lsc“ auf Seite 65
- Architektur [*MobiLink - Erste Orientierung*]

## Dienstprogramm zum MobiLink-Arbiter-Stopp (mlarbstop)

Mit dem mlarbstop-Befehl wird der MobiLink-Arbiterserver gestoppt.

### Syntax

**mlarbstop** [ *option* ]

Option	Beschreibung
-y	Stoppt den MobiLink-Arbiterserver sofort, auch wenn Verbindungen mit ihm bestehen.

### Bemerkungen

Das Dienstprogramm mlarbstop kann verwendet werden, um den MobiLink-Arbiterserver zu stoppen, wenn er auf dem lokalen Computer läuft.

Wenn Sie mlarbstop ohne die Option -y ausführen und keine Verbindungen mit dem Arbiterserver bestehen, wird der Arbiterserver sofort gestoppt.

Wenn Sie mlarbstop ohne die Option -y ausführen und Verbindungen mit dem Arbiterserver bestehen, gibt der MobiLink-Server einen Fehler aus.

### Beispiel

Das folgende Beispiel zeigt, wie Sie den MobiLink-Arbiterserver auf dem lokalen Computer stoppen können.

1. Stoppen Sie den MobiLink-Arbiter auf einem lokalen Computer mit der folgenden Befehlszeile:

```
mlarbstop -y
```

**Siehe auch**

- „MobiLink-Arbiterserver-Dienstprogramm für Windows (mlarbiter)“ auf Seite 711
- „MobiLink-Arbiterserver-Dienstprogramm für Unix (mlarbiter.sh)“ auf Seite 712
- „mlsrv16-Option -ca“ auf Seite 53
- „mlsrv16-Option -lsc“ auf Seite 65
- Architektur [*MobiLink - Erste Orientierung*]

# MobiLink-Datenzuordnungen zwischen entfernten und konsolidierten Datenbanken

Abhängig von der konsolidierten Datenbank, die Sie verwenden, kann der MobiLink-Server einen angegebenen Datentyp einem anderen Datentyp zuordnen. Die folgenden Abschnitte zeigen die Datentyp-Zuordnungen zwischen SQL Anywhere und/oder UltraLite und den unterstützten konsolidierten Datenbanken.

- „Zuordnung der Datentypen von Adaptive Server Enterprise“
- „IBM DB2 LUW-Datentypzuordnung“
- „Microsoft SQL Server-Datentypzuordnung“
- „MySQL-Datenzuordnung“
- „Oracle-Datentypzuordnung“
- „Sybase IQ Enterprise-Datenzuordnung“ auf Seite 761

## Zuordnung der Datentypen von Adaptive Server Enterprise

### Zuordnung zu konsolidierten Adaptive Server Enterprise-Datentypen

Die folgende Tabelle zeigt, wie entfernte SQL Anywhere- und UltraLite-Datentypen konsolidierten Adaptive Server Enterprise-Datentypen zugeordnet werden. Zum Beispiel sollte eine Spalte, die in der entfernten Datenbank den Datentyp LONG VARCHAR hat, in der konsolidierten Datenbank den Typ TEXT haben.

Maximum Column Length (MCL, Maximale Spaltenlänge) hängt von der Seitengröße in Adaptive Server Enterprise ab. Bei einer Seitengröße von 2 kB beträgt die MCL 1954 und bei einer Seitengröße von 4 kB beträgt die MCL 4002. Informationen über MCL finden Sie in der Dokumentation zu Adaptive Server Enterprise.

Datentyp in SQL Anywhere oder UltraLite	Datentyp in Adaptive Server Enterprise	Hinweise
BIGINT	BIGINT	

Datentyp in SQL Anywhere oder UltraLite	Datentyp in Adaptive Server Enterprise	Hinweise
BIT	BIT	
BINARY( $n \leq \text{MCL}$ )	BINARY( $n$ )	
BINARY( $n > \text{MCL}$ )	IMAGE	
CHAR( $n \leq \text{MCL}$ )	VARCHAR( $n$ )	
CHAR( $n > \text{MCL}$ )	TEXT	Sorgen Sie beim Download dafür, dass die Werte nicht zu lang sind.
DATE	DATE	Für Adaptive Server Enterprise DATETIME muss die Jahreszahl im Bereich von 1753 bis 9999 liegen.  Für SQL Anywhere und UltraLite muss die Zeitangabe das Format 00:00:00 haben.
DATETIME	DATETIME <sup>1</sup> oder BIGDATETIME <sup>2</sup>	Bei Adaptive Server Enterprise sind die DATETIME-Werte bis auf 1/300 Sekunde genau. Die letzte Ziffer des Sekundenbruchteils ist immer 0, 3 oder 6. Andere Ziffern werden auf eine dieser Ziffern gerundet. Somit werden 0 und 1 zu 0; 2, 3 und 4 werden zu 3; 5, 6, 7 und 8 werden zu 6 und 9 wird zu 10.  Bei einem Download behält SQL Anywhere die Originalwerte von Adaptive Server Enterprise bei, aber bei einem Upload können die Werte geringfügig von den Originalwerten abweichen.  Falls DATETIME für einen Primärschlüssel verwendet wird, könnte die Konfliktlösung fehlschlagen. Für eine einwandfreie Synchronisation von DATETIME können Sie die Sekundenbruchteile auf 10 Millisekunden runden. Außerdem muss das Jahr im Bereich von 1753 - 9999 liegen.
DECIMAL( $p < 39, s$ )	DECIMAL( $p, s$ )	Die Gesamtstellenzahl des Typs NUMERIC in Adaptive Server Enterprise kann 1 bis 38 Ziffern betragen ( $p < 39$ ).

Datentyp in SQL Anywhere oder UltraLite	Datentyp in Adaptive Server Enterprise	Hinweise
DECIMAL( $p \geq 39, s$ )		Es gibt keinen entsprechenden Datentyp in Adaptive Server Enterprise.
DOUBLE	DOUBLE PRECISION	
FLOAT( $p$ )	FLOAT( $p$ )	
IMAGE	IMAGE	
INTEGER	INTEGER	
LONG BINARY	IMAGE	
LONG NVARCHAR	UNITEXT	
LONG VARBIT	TEXT	
LONG VARCHAR	TEXT	
MONEY	MONEY	
NCHAR( $c \leq \text{MCL}$ )	UNIVARCHAR( $c/2$ )	
NCHAR( $c > \text{MCL}$ )	UNITEXT	Sorgen Sie beim Download dafür, dass die Werte nicht zu lang sind.
NTEXT	UNITEXT	
NUMERIC( $p < 39, s$ )	NUMERIC( $p, s$ )	Die Gesamtstellenzahl des Typs DECIMAL in Adaptive Server Enterprise kann 1 bis 38 Ziffern betragen ( $p < 39$ ).
NUMERIC( $p \geq 39, s$ )		Es gibt keinen entsprechenden Datentyp in Adaptive Server Enterprise.
NVARCHAR( $c \leq \text{MCL}$ )	UNIVARCHAR( $c/2$ )	
NVARCHAR( $c > \text{MCL}$ )	UNIVARCHAR( $c/2$ )	
REAL	REAL	

Datentyp in SQL Anywhere oder UltraLite	Datentyp in Adaptive Server Enterprise	Hinweise
SMALLDATETIME	DATETIME <sup>1</sup> oder BIGDATETIME <sup>2</sup>	<p>In SQL Anywhere und UltraLite wird SMALLDATETIME als TIMESTAMP implementiert.</p> <p>In Adaptive Server Enterprise ist DATETIME auf die Minute genau. 29,998 Sekunden oder weniger werden auf die Minute abgerundet, und Werte mit 29,999 Sekunden oder höher werden auf die nächste Minute aufgerundet. In SQL Anywhere oder UltraLite ist SMALLDATETIME auf die Mikrosekunde genau. Für eine einwandfreie Synchronisation in SQL Anywhere oder UltraLite muss SMALLDATETIME auf die Minute gerundet werden. Außerdem muss das Jahr im Bereich von 1753 - 9999 liegen.</p>
SMALLINT	SMALLINT	
SMALLMONEY	SMALLMONEY	
TEXT	TEXT	
TIME	TIME <sup>1</sup> oder BIGTIME <sup>2</sup>	<p>Bei Adaptive Server Enterprise sind die TIME-Werte bis auf 1/300 Sekunde genau. Die letzte Ziffer des Sekundenbruchteils ist immer 0, 3 oder 6. Andere Ziffern werden auf eine dieser Ziffern gerundet. Somit werden 0 und 1 zu 0; 2, 3 und 4 werden zu 3; 5, 6, 7 und 8 werden zu 6 und 9 wird zu 10. Bei einem Download behält SQL Anywhere die Originalwerte von Adaptive Server Enterprise bei, aber bei einem Upload können die Werte geringfügig von den Originalwerten abweichen. Falls TIME für einen Primärschlüssel verwendet wird, könnte die Konfliktlösung fehlschlagen. Für eine einwandfreie Synchronisation von TIME können Sie die Sekundenbruchteile auf 10 Millisekunden runden.</p>

Datentyp in SQL Anywhere oder UltraLite	Datentyp in Adaptive Server Enterprise	Hinweise
TIMESTAMP	DATETIME <sup>1</sup> oder BIGDATETIME <sup>2</sup>	<p>Bei Adaptive Server Enterprise sind die DATETIME-Werte bis auf 1/300 Sekunde genau. Die letzte Ziffer des Sekundenbruchteils ist immer 0, 3 oder 6. Andere Ziffern werden auf eine dieser Ziffern gerundet. Somit werden 0 und 1 zu 0; 2, 3 und 4 werden zu 3; 5, 6, 7 und 8 werden zu 6 und 9 wird zu 10.</p> <p>Bei einem Download behält SQL Anywhere die Originalwerte von Adaptive Server Enterprise bei, aber bei einem Upload können die Werte geringfügig von den Originalwerten abweichen.</p> <p>Falls DATETIME für einen Primärschlüssel verwendet wird, könnte die Konfliktlösung fehlschlagen. Für eine einwandfreie Synchronisation von DATETIME können Sie die Sekundenbruchteile auf 10 Millisekunden runden. Außerdem muss das Jahr im Bereich von 1753 - 9999 liegen.</p>
TIMESTAMP WITH TIME ZONE	VARCHAR(34)	<p>Es gibt keinen äquivalenten Datentyp in Adaptive Server Enterprise. Aus diesem Grund sollte eine TIMESTAMP WITH TIME ZONE-Spalte einer VARCHAR(34)-Spalte zugeordnet werden. Beim Upload konvertiert der MobiLink-Server zunächst die Daten in eine Zeichenfolge mit dem Format JJJJ-MM-TT HH:NN:SS.SSSSSS [+/-]HH:NN und übernimmt sie anschließend in die konsolidierte Datenbank. Beim Download konvertiert er die Daten aus der Zeichenfolge in das Format TIMESTAMP WITH TIME ZONE. Vergewissern Sie sich, dass die Daten in der konsolidierten Datenbank diesem Format entsprechen, sonst schlägt der Download fehl.</p>
TINYINT	TINYINT	
UNIQUEIDENTIFIER	CHAR(36)	

Datentyp in SQL Anywhere oder UltraLite	Datentyp in Adaptive Server Enterprise	Hinweise
UNIQUEIDENTIFIERSTR	CHAR(36)	UNIQUEIDENTIFIERSTR darf nicht verwendet werden. Verwenden Sie stattdessen UNIQUEIDENTIFIER.
UNSIGNED BIGINT	UNSIGNED BIGINT	
UNSIGNED INTEGER	UNSIGNED INT	
UNSIGNED SMALLINT	UNSIGNED SMALLINT	
UNSIGNED TINYINT	TINYINT	
VARBINARY( $n \leq \text{MCL}$ )	VARBINARY	
VARBINARY( $n > \text{MCL}$ )	IMAGE	
VARBIT( $n \leq \text{MCL}$ )	VARCHAR( $n$ )	
VARBIT( $n > \text{MCL}$ )	TEXT	
VARCHAR( $n \leq \text{MCL}$ )	VARCHAR( $n$ )	
VARCHAR( $n > \text{MCL}$ )	TEXT	
XML	TEXT	

<sup>1</sup> Gilt nur für Adaptive Server Enterprise vor Version 15.5.

<sup>2</sup> Gilt nur für Adaptive Server Enterprise ab Version 15.5.

### Zuordnung zu entfernten SQL Anywhere- oder UltraLite-Datentypen

Die folgende Tabelle zeigt, wie konsolidierte Adaptive Server Enterprise-Datentypen entfernten SQL Anywhere- und UltraLite-Datentypen zugeordnet werden. Zum Beispiel sollte eine Spalte, die in der konsolidierten Datenbank den Datentyp DOUBLE PRECISION hat, in der entfernten Datenbank den Typ DOUBLE haben.

Datentyp in Adaptive Server Enterprise	Datentyp in SQL Anywhere oder UltraLite	Hinweise
BIGINT	BIGINT	
BIGDATETIME <sup>1</sup>	TIMESTAMP	

Datentyp in Adaptive Server Enterprise	Datentyp in SQL Anywhere oder UltraLite	Hinweise
BIGTIME <sup>1</sup>	TIME	
BINARY( <i>n</i> )	BINARY( <i>n</i> )	
BIT	BIT	
CHAR( <i>n</i> )	VARCHAR( <i>n</i> )	Es gibt keine Äquivalenz zwischen SQL Anywhere CHAR/NCHAR und Adaptive Server Enterprise CHAR/NCHAR. SQL Anywhere CHAR/NCHAR ist äquivalent zu VARCHAR/NVARCHAR. Sie sollten CHAR/NCHAR nicht in konsolidierten Datenbankspalten verwenden, die synchronisiert sind. Falls Sie Nicht-SQL Anywhere CHAR/NCHAR verwenden müssen, führen Sie den MobiLink-Server mit der Option -b aus.
DATE	DATE	Für SQL Anywhere und UltraLite muss die Zeitangabe das Format 00:00:00 haben.
DATETIME	DATETIME	<p>Bei Adaptive Server Enterprise sind die DATETIME-Werte bis auf 1/300 Sekunde genau. Die letzte Ziffer des Sekundenbruchteils ist immer 0, 3 oder 6. Andere Ziffern werden auf eine dieser Ziffern gerundet. Somit werden 0 und 1 zu 0; 2, 3 und 4 werden zu 3; 5, 6, 7 und 8 werden zu 6 und 9 wird zu 10.</p> <p>Bei einem Download behält SQL Anywhere die Originalwerte von Adaptive Server Enterprise bei, aber bei einem Upload können die Werte geringfügig von den Originalwerten abweichen. Die Konfliktlösung kann fehlschlagen. Für eine einwandfreie Synchronisation von DATETIME können Sie die Sekundenbruchteile auf 10 Millisekunden runden. Außerdem muss das Jahr im Bereich von 1753 - 9999 liegen.</p>
DECIMAL( <i>p,s</i> )	DECIMAL( <i>p,s</i> )	
DOUBLE PRECISION	DOUBLE	

Datentyp in Adaptive Server Enterprise	Datentyp in SQL Anywhere oder UltraLite	Hinweise
FLOAT( <i>p</i> )	FLOAT( <i>p</i> )	
IMAGE	LONG BINARY	
INT	INT	
MONEY	MONEY	
NCHAR( <i>n</i> )	VARCHAR( <i>n</i> )	Adaptive Server Enterprise NCHAR und NVARCHAR speichern landesspezifische Mehrbyte-Zeichenfolgen, die sich von SQL Anywhere NCHAR und NVARCHAR unterscheiden. Verwenden Sie in einer Mehrbyte-Umgebung SQL Anywhere oder UltraLite VARCHAR.
NUMERIC( <i>p,s</i> )	NUMERIC( <i>p,s</i> )	
NVARCHAR( <i>n</i> )	VARCHAR( <i>n</i> )	Adaptive Server Enterprise NCHAR und NVARCHAR speichern landesspezifische Mehrbyte-Zeichenfolgen, die sich von SQL Anywhere NCHAR und NVARCHAR unterscheiden. Verwenden Sie in einer Mehrbyte-Umgebung SQL Anywhere oder UltraLite VARCHAR.
REAL	REAL	
SMALLDATETIME	SMALLDATETIME	<p>In SQL Anywhere und UltraLite wird SMALLDATETIME als TIMESTAMP implementiert.</p> <p>In Adaptive Server Enterprise ist SMALLDATETIME auf die Minute genau. 29,998 Sekunden oder weniger werden auf die Minute abgerundet, und Werte mit 29,999 Sekunden oder höher werden auf die nächste Minute aufgerundet. In SQL Anywhere oder UltraLite ist SMALLDATETIME auf die Mikrosekunde genau. Für eine einwandfreie Synchronisation in SQL Anywhere oder UltraLite muss SMALLDATETIME auf die Minute gerundet werden. Außerdem muss das Jahr im Bereich von 1900 - 2078 liegen.</p>

Datentyp in Adaptive Server Enterprise	Datentyp in SQL Anywhere oder UltraLite	Hinweise
SMALLINT	SMALLINT	
SMALLMONEY	SMALLMONEY	
TEXT	LONG VARCHAR	
TIME	TIME	<p>Bei Adaptive Server Enterprise sind die TIME-Werte bis auf 1/300 Sekunde genau. Die letzte Ziffer des Sekundenbruchteils ist immer 0, 3 oder 6. Andere Ziffern werden auf eine dieser Ziffern gerundet. Somit werden 0 und 1 zu 0; 2, 3 und 4 werden zu 3; 5, 6, 7 und 8 werden zu 6 und 9 wird zu 10.</p> <p>Bei einem Download behält SQL Anywhere die Originalwerte von Adaptive Server Enterprise bei, aber bei einem Upload können die Werte geringfügig von den Originalwerten abweichen. Die Konfliktlösung kann fehlschlagen. Für eine einwandfreie Synchronisation von TIME sollten Sie die Sekundenbruchteile auf 10 Millisekunden runden.</p>
TIMESTAMP	VARBINARY(8)	<p>In Adaptive Server Enterprise ist TIMESTAMP ein Binärzähler, der jeweils nach einer Änderung einer Zeile erhöht wird. Jede Tabelle kann daher eine TIMESTAMP-Spalte enthalten und es ist nicht sinnvoll, sie zu synchronisieren. Wenn sie in einer Synchronisation enthalten sein muss, sollten Sie sie in SQL Anywhere oder UltraLite dem Datentyp VARBINARY(8) zuordnen.</p> <p>Es ist nicht möglich, diese Zeitstempel-Spalte explizit einzufügen oder zu aktualisieren, da sie vom Server verwaltet wird. Berücksichtigen Sie dies, wenn Sie Uploadskripten für Tabellen implementieren, die solche Spalten enthalten.</p>
TINYINT	TINYINT	
UNSIGNED BIGINT	UNSIGNED BIGINT	

Datentyp in Adaptive Server Enterprise	Datentyp in SQL Anywhere oder UltraLite	Hinweise
UNSIGNED INT	UNSIGNED INT	
UNSIGNED SMALLINT	UNSIGNED SMALLINT	
VARBINARY( <i>n</i> )	VARBINARY( <i>n</i> )	
VARCHAR( <i>n</i> )	VARCHAR( <i>n</i> )	
UNICHAR( <i>n</i> )	NVARCHAR( <i>n</i> )	In UltraLite nicht verfügbar.
UNTEXT	LONG NVARCHAR	In UltraLite nicht verfügbar.
UNIVARCHAR( <i>n</i> )	NVARCHAR( <i>n</i> )	In UltraLite nicht verfügbar.

<sup>1</sup> Gilt nur für Adaptive Server Enterprise ab Version 15.5.

## IBM DB2 LUW-Datentypzuordnung

### Zuordnung zu konsolidierten IBM DB2 LUW-Datentypen

Die folgende Tabelle zeigt, wie entfernte SQL Anywhere- und UltraLite-Datentypen konsolidierten IBM DB2 LUW-Datentypen zugeordnet werden. Zum Beispiel sollte eine Spalte, die in der entfernten Datenbank den Datentyp BIT hat, in der konsolidierten Datenbank den Typ SMALLINT haben.

Wenn Sie eine IBM DB2 LUW-Tabelle erstellen, müssen Sie auf die DB2-Seitengröße achten. Für IBM DB2 LUW ist die maximale Zeilenlänge (MRL) von der Seitengröße abhängig: Die MRL beträgt 4005 bei einer Seitengröße von 4 kB, 8101 bei 8 kB, 16293 bei 16 kB und 32677 bei 32 kB. Die Länge der Spalten in einer Tabelle darf die oben genannten Grenzen nicht überschreiten. Falls eine Tabelle eine BLOB- oder CLOB-Spalte enthält, stellen Sie die Zeilenlänge mit dem LOB Locator fest, nicht direkt mit den BLOB- oder CLOB-Daten. Weitere Hinweise finden Sie in Ihrer IBM DB2 LUW-Dokumentation.

Datentyp in SQL Anywhere oder UltraLite	IBM DB2 LUW-Datentyp	Hinweise
BIGINT	BIGINT	
BINARY( <i>n</i> <MRL)	VARCHAR( <i>n</i> ) FOR BIT DATA	
BINARY( <i>n</i> >=MRL)	BLOB( <i>n</i> )	
BIT	SMALLINT	

Datentyp in SQL Anywhere oder UltraLite	IBM DB2 LUW-Datentyp	Hinweise
CHAR( $n < \text{MRL}$ )	VARCHAR( $n$ )	
CHAR( $n \geq \text{MRL}$ )	CLOB( $n$ )	IBM DB2 LUW-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.
DATE	DATE	Für SQL Anywhere und UltraLite muss die Zeitangabe das Format 00:00:00 haben.
DATETIME	TIMESTAMP	
DECIMAL( $p < 32, s$ )	DECIMAL( $p, s$ )	Die Gesamtstellenzahl des SQL Anywhere-Datentyps DECIMAL liegt zwischen 1 und 127. Die maximale Gesamtstellenzahl des IBM DB2 LUW-Datentyps beträgt 31.
DECIMAL( $p \geq 32, s$ )		Daten des SQL Anywhere-Datentyps DECIMAL mit einer Gesamtstellenzahl von über 31 können nicht mit IBM DB2 LUW synchronisiert werden.
DOUBLE	DOUBLE	DOUBLE ist ein ungenauer numerischer Datentyp, der gerundet wird. Bei verschiedenen Computertypen ist die Basisspeicherung für DOUBLE oft unterschiedlich, was zu unterschiedlichem Runden führt. DOUBLE ist keine gute Wahl für einen Primärschlüssel, da bei Primärschlüsseln auf Gleichheit geprüft wird. Dies gilt besonders in einer Synchronisationsumgebung, da die konsolidierte Datenbank oft auf anderer Hardware läuft wie die entfernte Datenbank.
FLOAT(1-24)	REAL	FLOAT kann Probleme verursachen, wenn die konsolidierten und entfernten Datenbanken nicht die exakt gleichen (ungenauen) Werte erlauben. Da nicht alle möglichen Werte getestet werden, ist Vorsicht angebracht. Um Probleme zu vermeiden, sollten Sie diese Typen nicht als Teil eines Primärschlüssels verwenden.

Datentyp in SQL Anywhere oder UltraLite	IBM DB2 LUW-Datentyp	Hinweise
FLOAT(25-53)	DOUBLE	FLOAT kann Probleme verursachen, wenn die konsolidierten und entfernten Datenbanken nicht die exakt gleichen (ungenauen) Werte erlauben. Da nicht alle möglichen Werte getestet werden, ist Vorsicht angebracht. Um Probleme zu vermeiden, sollten Sie diese Typen nicht als Teil eines Primärschlüssels verwenden.
IMAGE	BLOB( <i>n</i> )	
INTEGER	INTEGER	
LONG BINARY	BLOB( <i>n</i> )	
LONG NVARCHAR	CLOB( <i>n</i> )	Es gibt keinen entsprechenden Datentyp in IBM DB2 LUW. Falls es sich beim IBM DB2 LUW-Zeichensatz um Unicode handelt, kann der SQL Anywhere-Datentyp LONG NVARCHAR mit dem IBM DB2 LUW-Datentyp CLOB synchronisiert werden. UltraLite verfügt nicht über LONG NVARCHAR.
LONG VARBIT	CLOB( <i>n</i> )	
LONG VARCHAR	CLOB( <i>n</i> )	
MONEY	DECIMAL (19,4)	

Datentyp in SQL Anywhere oder UltraLite	IBM DB2 LUW-Datentyp	Hinweise
NCHAR( <i>c</i> )	VARCHAR( <i>n</i> ) oder CLOB( <i>n</i> )	Es gibt keinen entsprechenden Datentyp in IBM DB2 LUW. Falls es sich beim IBM DB2 LUW-Zeichensatz um Unicode handelt, kann der SQL Anywhere-Datentyp NCHAR mit dem IBM DB2 LUW-Datentyp VARCHAR oder CLOB synchronisiert werden. Die Größeneinheit ist beim SQL Anywhere-Datentyp NCHAR Zeichen und beim IBM DB2 LUW-Datentyp VARCHAR Byte. Im Fall einer Zuordnung zu VARCHAR darf die Gesamtzahl der Bytes von NCHAR nicht größer sein als MRL. Andernfalls sollte NCHAR CLOB zugeordnet werden. Es ist schwierig, die Anzahl der Bytes in NCHAR( <i>c</i> ) zu berechnen, aber sie entspricht ungefähr $c \approx n \cdot 4$ . Allgemein gilt: Wenn <i>c</i> kleiner als MRL/4 ist, sollte zu VARCHAR( <i>n</i> ) zugeordnet werden. Wenn dagegen <i>c</i> größer oder gleich MRL/4 ist, sollte zu CLOB( <i>n</i> ) zugeordnet werden.
NUMERIC( <i>p</i> <32, <i>s</i> )	NUMERIC( <i>p</i> , <i>s</i> )	
NUMERIC( <i>p</i> ≥32, <i>s</i> )		Es gibt keinen entsprechenden Datentyp in IBM DB2 LUW.
NTEXT	CLOB( <i>n</i> )	Es gibt keinen entsprechenden Datentyp in IBM DB2 LUW. Falls es sich beim IBM DB2 LUW-Zeichensatz um Unicode handelt, kann NTEXT mit dem IBM DB2 LUW-Datentyp CLOB synchronisiert werden.

Datentyp in SQL Anywhere oder UltraLite	IBM DB2 LUW-Datentyp	Hinweise
NVARCHAR( <i>c</i> )	VARCHAR( <i>n</i> ) oder CLOB( <i>n</i> )	Es gibt keinen entsprechenden Datentyp in IBM DB2 LUW. Falls es sich beim IBM DB2 LUW-Zeichensatz um Unicode handelt, kann NVARCHAR mit dem IBM DB2 LUW-Datentyp VARCHAR oder CLOB synchronisiert werden. Die Größeneinheit ist beim SQL Anywhere-Datentyp NVARCHAR Zeichen und beim IBM DB2 LUW-Datentyp VARCHAR Byte. Im Fall einer Zuordnung zu VARCHAR darf die Gesamtzahl der Bytes von NVARCHAR nicht größer sein als MRL. Andernfalls sollte NVARCHAR CLOB zugeordnet werden. Es ist schwierig, die Anzahl der Bytes in NVARCHAR( <i>c</i> ) zu berechnen, aber sie entspricht ungefähr $c=n/4$ . Allgemein gilt: Wenn <i>c</i> kleiner als MRL/4 ist, sollte zu VARCHAR( <i>n</i> ) zugeordnet werden. Wenn dagegen <i>c</i> größer oder gleich MRL/4 ist, sollte zu CLOB( <i>n</i> ) zugeordnet werden.
REAL	REAL	REAL kann Probleme verursachen, wenn die konsolidierten und entfernten Datenbanken nicht die exakt gleichen (ungenauen) Werte erlauben. Da nicht alle möglichen Werte getestet werden, ist Vorsicht angebracht. Um Probleme zu vermeiden, sollten Sie diese Typen nicht als Teil eines Primärschlüssels verwenden.
SMALLDATETIME	TIMESTAMP	
SMALLINT	SMALLINT	
SMALLMONEY	DECIMAL(10,4)	
ST_GEOMETRY	ST_GEOMETRY	
TEXT	CLOB( <i>n</i> )	

<b>Datentyp in SQL Anywhere oder UltraLite</b>	<b>IBM DB2 LUW-Datentyp</b>	<b>Hinweise</b>
TIME	TIMESTAMP oder TIME	TIME-Werte in SQL Anywhere und UltraLite mit Sekundenbruchteilen erfordern den IBM DB2 LUW-Datentyp TIMESTAMP. TIME-Werte in SQL Anywhere und UltraLite mit Sekundenbruchteilen, die immer Null sind, können den IBM DB2 LUW-Datentyp TIME verwenden. Um die Gesamtstellenzahl einer TIME-Spalte beizubehalten, bindet der MobiLink-Server die TIME-Spalte immer an den ODBC-Datentyp SQL_TYPE_TIMESTAMP. Wenn die konsolidierte Datenbank auf einem DB2 9.7-Server läuft, müssen Sie die Spalte möglicherweise unter Verwendung von DB2-Konvertierungsfunktionen explizit zwischen TIMESTAMP und TIME konvertieren, falls die Spalte Teil eines Primärschlüssels ist.
TIMESTAMP	TIMESTAMP	
TIMESTAMP WITH TIME ZONE	VARCHAR(34)	Es gibt keinen äquivalenten Datentyp in IBM DB2 LUW. Aus diesem Grund sollte eine TIMESTAMP WITH TIME ZONE-Spalte einer VARCHAR(34)-Spalte zugeordnet werden. Beim Upload konvertiert der MobiLink-Server zunächst die Daten in eine Zeichenfolge mit dem Format JJJJ-MM-TT HH:NN:SS.SSSSSS [+-]HH:NN und übernimmt sie anschließend in die konsolidierte Datenbank. Beim Download konvertiert er die Daten aus der Zeichenfolge in das Format TIMESTAMP WITH TIME ZONE. Vergewissern Sie sich, dass die Daten in der konsolidierten Datenbank diesem Format entsprechen, sonst schlägt der Download fehl.
TINYINT	SMALLINT	Bei Downloads dürfen IBM DB2 LUW-Werte nicht negativ sein.
UNIQUEIDENTIFIER	CHAR(36)	
UNIQUEIDENTIFIERSTR	CHAR(36)	UNIQUEIDENTIFIERSTR wird für IBM DB2 LUW nicht empfohlen. Verwenden Sie stattdessen UNIQUEIDENTIFIER.

Datentyp in SQL Anywhere oder UltraLite	IBM DB2 LUW-Datentyp	Hinweise
UNSIGNED BIGINT	DECIMAL (20)	Bei Downloads dürfen IBM DB2 LUW-Werte nicht negativ sein.
UNSIGNED INTEGER	DECIMAL (11)	Bei Downloads dürfen IBM DB2 LUW-Werte nicht negativ sein.
UNSIGNED SMALLINT	DECIMAL (5)	Bei Downloads dürfen IBM DB2 LUW-Werte nicht negativ sein.
UNSIGNED TINYINT	SMALLINT	Bei Downloads dürfen IBM DB2 LUW-Werte nicht negativ sein.
VARBINARY( $n < \text{MRL}$ )	VARCHAR( $n$ ) FOR BIT DATA	
VARBINARY( $n \geq \text{MRL}$ )	BLOB( $n$ )	
VARBIT( $n < \text{MRL}$ )	VARCHAR( $n$ )	
VARBIT( $n \geq \text{MRL}$ )	CLOB( $n$ )	
VARCHAR( $n < \text{MRL}$ )	VARCHAR( $n$ )	
VARCHAR( $n \geq \text{MRL}$ )	CLOB( $n$ )	IBM DB2 LUW-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.
XML	CLOB( $n$ )	

### Zuordnung zu entfernten SQL Anywhere- oder UltraLite-Datentypen

Die folgende Tabelle zeigt, wie konsolidierte IBM DB2 LUW-Datentypen entfernten SQL Anywhere- und UltraLite-Datentypen zugeordnet werden. Zum Beispiel sollte eine Spalte, die in der konsolidierten Datenbank den Datentyp INT hat, in der entfernten Datenbank den Typ INTEGER haben.

Wenn Sie eine IBM LUW-Tabelle erstellen, müssen Sie auf die DB2-Seitengröße achten. Für IBM DB2 LUW ist die maximale Zeilenlänge von der Seitengröße abhängig: Die MRL beträgt 4005 bei einer Seitengröße von 4 kB, 8101 bei 8 kB, 16293 bei 16 kB und 32677 bei 32 kB. Die Länge der Spalten in einer Tabelle darf die oben genannten Grenzen nicht überschreiten. Falls eine Tabelle eine BLOB- oder CLOB-Spalte enthält, stellen Sie die Zeilenlänge mit dem LOB Locator fest, nicht direkt mit den BLOB- oder CLOB-Daten. Weitere Hinweise finden Sie in Ihrer IBM DB2 LUW-Dokumentation.

IBM DB2 LUW-Datentyp	Datentyp in SQL Anywhere oder UltraLite	Hinweise
BLOB	LONG BINARY	
BIGINT	BIGINT	
CHAR( <i>n</i> )	VARCHAR( <i>n</i> )	In SQL Anywhere gibt es kein Äquivalent zum IBM DB2 LUW-Datentyp CHAR. Sie sollten CHAR nicht in konsolidierten Datenbankspalten verwenden, die synchronisiert werden. Falls Sie CHAR-Spalten aus IBM DB2 LUW synchronisieren müssen, führen Sie den MobiLink-Server mit der Option -b aus.
CHAR( <i>n</i> ) FOR BIT DATA	BINARY( <i>n</i> )	
CLOB( <i>n</i> )	LONG VARCHAR	
DATE	DATE	Für SQL Anywhere und UltraLite muss die Zeitangabe das Format 00:00:00 haben.
DB2GSE.ST_GEOMETRY	ST_GEOMETRY	
DBCLOB( <i>n</i> )	LONG VARCHAR	Der Datentyp DBCLOB( <i>n</i> ) wird nur für Doppelbyte-Zeichen verwendet. SQL Anywhere verfügt nicht über einen entsprechenden Datentyp. Wenn es sich beim IBM DB2 LUW-Zeichensatz um Unicode handelt, ist DBCLOB( <i>n</i> ) gleichwertig mit CLOB.
DECIMAL( <i>p,s</i> )	DECIMAL( <i>p,s</i> )	
DOUBLE	DOUBLE	DOUBLE ist ein ungenauer numerischer Datentyp, der gerundet wird. Bei verschiedenen Computertypen ist die Basisspeicherung für DOUBLE oft unterschiedlich, was zu unterschiedlichem Runden führt. DOUBLE ist keine gute Wahl für einen Primärschlüssel, da bei Primärschlüsseln auf Gleichheit geprüft wird. Dies gilt besonders in einer Synchronisationsumgebung, da die konsolidierte Datenbank oft auf anderer Hardware läuft wie die entfernte Datenbank.

IBM DB2 LUW-Datentyp	Datentyp in SQL Anywhere oder UltraLite	Hinweise
FLOAT	DOUBLE	FLOAT kann Probleme verursachen, wenn die konsolidierten und entfernten Datenbanken nicht die exakt gleichen (ungenauen) Werte erlauben. Da nicht alle möglichen Werte getestet werden, ist Vorsicht angebracht. Um Probleme zu vermeiden, sollten Sie diese Typen nicht als Teil eines Primärschlüssels verwenden.
GRAPHIC( <i>n</i> )	VARCHAR( <i>2n</i> )	<p>Der IBM DB2 LUW-Datentyp GRAPHIC füllt mit Leerzeichen auf, der SQL Anywhere-Datentyp CHAR dagegen nicht. Es wird empfohlen, diesen Datentyp nicht zu verwenden.</p> <p>Der Datentyp GRAPHIC wird nur für Doppelbyte-Zeichen verwendet. SQL Anywhere verfügt nicht über einen entsprechenden Datentyp. Wenn es sich beim IBM DB2 LUW-Zeichensatz um Unicode handelt, ist GRAPHIC gleichwertig mit CHAR.</p>
INT	INTEGER	
LONG VARCHAR	VARCHAR(32700)	
LONG VARCHAR FOR BIT DATA	VARBINARY(32700)	
LONG VARGRAPHIC( <i>n</i> )	VARCHAR(32700)	Der Datentyp LONG VARGRAPHIC wird nur für Doppelbyte-Zeichen verwendet. SQL Anywhere verfügt nicht über einen entsprechenden Datentyp. Wenn es sich beim IBM DB2 LUW-Zeichensatz um Unicode handelt, ist LONG VARGRAPHIC gleichwertig mit LONG VARCHAR.
NUMERIC( <i>p,s</i> )	NUMERIC( <i>p,s</i> )	

IBM DB2 LUW-Datentyp	Datentyp in SQL Anywhere oder UltraLite	Hinweise
REAL	REAL	REAL kann Probleme verursachen, wenn die konsolidierten und entfernten Datenbanken nicht die exakt gleichen (ungenauen) Werte erlauben. Da nicht alle möglichen Werte getestet werden, ist Vorsicht angebracht. Um Probleme zu vermeiden, sollten Sie diese Typen nicht als Teil eines Primärschlüssels verwenden.
SMALLINT	SMALLINT	
TIME	TIME	Die Sekundenbruchteile von SQL Anywhere TIME-Werten werden beim Download weggekürzt. Um Probleme zu vermeiden, sollten Sie keine Sekundenwerte mit Bruchteilen verwenden. Um die Gesamtstellenzahl einer TIME-Spalte beizubehalten, bindet der MobiLink-Server die TIME-Spalte immer an den ODBC-Datentyp SQL_TYPE_TIMESTAMP. Wenn die konsolidierte Datenbank auf einem DB2 9.7-Server läuft, müssen Sie die Spalte möglicherweise unter Verwendung von DB2-Konvertierungsfunktionen explizit zwischen TIMESTAMP und TIME konvertieren, falls die Spalte Teil eines Primärschlüssels ist.
TIMESTAMP	TIMESTAMP	
VARCHAR( <i>n</i> )	VARCHAR( <i>n</i> )	
VARCHAR( <i>n</i> ) FOR BIT DATA	VARBINARY( <i>n</i> )	
VARGRAPHIC( <i>n</i> )	VARCHAR( <i>2n</i> )	Der Datentyp VARGRAPHIC wird nur für Doppelbyte-Zeichen verwendet. SQL Anywhere verfügt nicht über einen entsprechenden Datentyp. Wenn es sich beim IBM DB2 LUW-Zeichensatz um Unicode handelt, ist VARGRAPHIC gleichwertig mit VARCHAR.

## Microsoft SQL Server-Datentypzuordnung

### Zuordnung zu konsolidierten Microsoft SQL Server-Datentypen

Die folgende Tabelle zeigt, wie entfernte SQL Anywhere- und UltraLite-Datentypen konsolidierten Microsoft SQL Server-Datentypen zugeordnet werden. Zum Beispiel sollte eine Spalte, die in der entfernten Datenbank den Datentyp DATETIME hat, in der konsolidierten Datenbank den Typ DATETIME2 haben.

Datentyp in SQL Anywhere oder UltraLite	Microsoft SQL Server-Datentyp	Hinweise
BIGINT	BIGINT	
BINARY( $n \leq 8000$ )	VARBINARY( $n$ )	
BINARY( $n > 8000$ )	VARBINARY(MAX)	
BIT	BIT	
CHAR( $n \leq 8000$ )	VARCHAR( $n$ )	
CHAR( $n > 8000$ )	VARCHAR(MAX)	
DATE	DATE	
DATETIME	DATETIME2	DATETIME2- und TIME-Werte von Microsoft SQL Server sind bis auf 100 Nanosekunden genau. TIMESTAMP- und TIME-Werte sind dagegen nur bis auf 1 Mikrosekunde genau. Für eine einwandfreie Synchronisation von DATETIME2 und TIME sollten Sie die Sekundenbruchteile auf 1 Mikrosekunde runden.
DECIMAL( $p \leq 38, s$ )	DECIMAL( $p, s$ )	Die Gesamtstelligenzahl des Microsoft SQL Server-Datentyps DECIMAL/NUMERIC reicht von 1 bis 38, daher muss $p$ kleiner als 39 sein.
DECIMAL( $p > 38, s$ )		Es gibt keinen entsprechenden Datentyp in Microsoft SQL Server.
DOUBLE	FLOAT(53)	
FLOAT( $p$ )	FLOAT( $p$ )	
IMAGE	VARBINARY(MAX)	

Datentyp in SQL Anywhere oder UltraLite	Microsoft SQL Server-Datentyp	Hinweise
INTEGER	INT	
LONG BINARY	VARBINARY(MAX)	
LONG NVARCHAR	NVARCHAR(MAX)	
LONG VARBIT	VARCHAR(MAX)	
LONG VARCHAR	VARCHAR(MAX)	
MONEY	MONEY	
NCHAR( $n \leq 4000$ )	NVARCHAR( $c$ )	
NCHAR( $n > 4000$ )	NVARCHAR(MAX)	
NTEXT	NVARCHAR(MAX)	
NUMERIC( $p \leq 38, s$ )	NUMERIC( $p, s$ )	Die Gesamtstanzahl des Microsoft SQL Server-Datentyps DECIMAL/NUMERIC reicht von 1 bis 38, daher muss $p$ kleiner als 39 sein.
NUMERIC( $p > 38, s$ )		Es gibt keinen entsprechenden Datentyp in Microsoft SQL Server.
NVARCHAR( $n \leq 4000$ )	NVARCHAR( $c$ )	
NVARCHAR( $n > 4000$ )	NVARCHAR(MAX)	
REAL	REAL	

Datentyp in SQL Anywhere oder UltraLite	Microsoft SQL Server-Datentyp	Hinweise
SMALLDATETIME	SMALLDATETIME	In SQL Anywhere und UltraLite wird SMALLDATETIME als TIMESTAMP implementiert. SMALLDATETIME-Werte von Microsoft SQL Server sind auf die Minute genau. 29,998 Sekunden oder weniger werden auf die Minute abgerundet, und Werte mit 29,999 Sekunden oder höher werden auf die nächste Minute aufgerundet. In SQL Anywhere oder UltraLite ist SMALLDATETIME auf die Mikrosekunde genau. Für eine einwandfreie Synchronisation in SQL Anywhere oder UltraLite muss SMALLDATETIME auf die Minute gerundet werden. Das Jahr muss im Bereich von 1900 - 2078 liegen.
SMALLINT	SMALLINT	
SMALLMONEY	SMALLMONEY	
ST_GEOMETRY	GEOMETRY	
TEXT	VARCHAR(MAX)	
TIME	TIME	DATETIME2- und TIME-Werte von Microsoft SQL Server sind bis auf 100 Nanosekunden genau. TIMESTAMP- und TIME-Werte sind dagegen nur bis auf 1 Mikrosekunde genau. Für eine einwandfreie Synchronisation von DATETIME2 und TIME sollten Sie die Sekundenbruchteile auf 1 Mikrosekunde runden.
TIMESTAMP	DATETIME2	DATETIME2- und TIME-Werte von Microsoft SQL Server sind bis auf 100 Nanosekunden genau. TIMESTAMP- und TIME-Werte sind dagegen nur bis auf 1 Mikrosekunde genau. Für eine einwandfreie Synchronisation von DATETIME2 und TIME sollten Sie die Sekundenbruchteile auf 1 Mikrosekunde runden.
TIMESTAMP WITH TIME ZONE	DATETIMEOFFSET	

Datentyp in SQL Anywhere oder UltraLite	Microsoft SQL Server-Datentyp	Hinweise
TINYINT	TINYINT	Bei Downloads dürfen Werte nicht negativ sein.
UNIQUEIDENTIFIER	UNIQUEIDENTIFIER	
UNIQUEIDENTIFIERSTR	UNIQUEIDENTIFIER	
UNSIGNED BIGINT	NUMERIC(20)	Bei Downloads dürfen Werte nicht negativ sein.
UNSIGNED INTEGER	NUMERIC(11)	Bei Downloads dürfen Werte nicht negativ sein.
UNSIGNED TINYINT	TINYINT	Bei Downloads dürfen Werte nicht negativ sein.
UNSIGNED SMALLINT	INT	Bei Downloads dürfen Werte nicht negativ sein.
VARBINARY( $n \leq 8000$ )	VARBINARY( $n$ )	
VARBINARY( $n > 8000$ )	VARBINARY(MAX)	
VARBIT( $n \leq 8000$ )	VARCHAR( $n$ )	
VARBIT( $n > 8000$ )	VARCHAR(MAX)	
VARCHAR( $n \leq 8000$ )	VARCHAR( $c$ )	
VARCHAR( $n > 8000$ )	VARCHAR(MAX)	
XML	XML oder VARCHAR(MAX)	Verwenden Sie für Microsoft SQL Server 2005 XML. Für andere Versionen verwenden Sie VARCHAR(MAX).

### Zuordnung zu entfernten SQL Anywhere- oder UltraLite-Datentypen

Die folgende Tabelle zeigt, wie konsolidierte Microsoft SQL Server-Datentypen entfernten SQL Anywhere- und UltraLite-Datentypen zugeordnet werden. Zum Beispiel sollte eine Spalte, die in der entfernten Datenbank den Datentyp TEXT hat, in der konsolidierten Datenbank den Typ LONG VARCHAR haben.

Microsoft SQL Server-Datentyp	Datentyp in SQL Anywhere oder UltraLite	Hinweise
BIGINT	BIGINT	
BINARY( <i>n</i> )	BINARY( <i>n</i> )	
BIT	BIT	
CHAR( <i>n</i> )	VARCHAR( <i>n</i> )	Eine CHAR-Spalte von Microsoft SQL Server wird mit Leerzeichen aufgefüllt. Eine CHAR-Spalte von SQL Anywhere wird standardmäßig nicht mit Leerzeichen aufgefüllt und entspricht einer VARCHAR-Spalte. Sie sollten daher CHAR-Datentypen in Synchronisationstabellen in Microsoft SQL Server vermeiden. Wenn Sie den CHAR-Datentyp in der konsolidierten Datenbank in Microsoft SQL Server verwenden müssen, führen Sie den MobiLink-Server mit der Befehlszeilenoption -b aus, um die Unterschiede zwischen den CHAR-Datentypen in SQL Anywhere und anderen Datenbanken aufzulösen.
DATE	DATE	
DATETIME	TIMESTAMP oder DATETIME	DATETIME-Werte von Microsoft SQL Server sind bis auf 1/300 Sekunden genau. Die letzte Ziffer des Sekundenbruchteils ist immer 0, 3 oder 6. Andere Ziffern werden auf eine dieser Ziffern gerundet. Somit werden 0 und 1 zu 0; 2, 3 und 4 werden zu 3; 5, 6, 7 und 8 werden zu 6 und 9 wird zu 10. Bei einem Download behält SQL Anywhere die Originalwerte aus Microsoft SQL Server bei, aber bei einem Upload können die Werte geringfügig von den Originalwerten abweichen. Falls DATETIME für einen Primärschlüssel verwendet wird, könnte die Konfliktlösung fehlschlagen. Für eine einwandfreie Synchronisation von DATETIME können Sie die Sekundenbruchteile auf 10 Millisekunden runden. Das Jahr muss im Bereich von 1753 - 9999 liegen.

Microsoft SQL Server-Datentyp	Datentyp in SQL Anywhere oder UltraLite	Hinweise
DATETIME2	TIMESTAMP	DATETIME2- und TIME-Werte von Microsoft SQL Server sind bis auf 100 Nanosekunden genau. TIMESTAMP- und TIME-Werte sind dagegen nur bis auf 1 Mikrosekunde genau. Für eine einwandfreie Synchronisation von DATETIME2 und TIME sollten Sie die Sekundenbruchteile auf 1 Mikrosekunde runden.
DECIMAL( <i>p,s</i> )	DECIMAL( <i>p,s</i> )	
FLOAT( <i>p</i> )	FLOAT( <i>p</i> )	
GEOMETRY	ST_GEOMETRY	
IMAGE	LONG BINARY	
INT	INT	
MONEY	MONEY	
NCHAR( <i>n</i> )	NVARCHAR( <i>c</i> )	In UltraLite nicht verfügbar.  SQL Anywhere NCHAR und Nicht-SQL Anywhere NCHAR sind nicht äquivalent. SQL Anywhere NCHAR ist äquivalent zu NVARCHAR. Sie sollten NCHAR nicht in konsolidierten Datenbankspalten verwenden, die synchronisiert sind. Falls Sie Nicht-SQL Anywhere NCHAR verwenden müssen, führen Sie den MobiLink-Server mit der Option -b aus.
NTEXT	LONG NVARCHAR	In UltraLite nicht verfügbar.
NVARCHAR( <i>c</i> )	NVARCHAR( <i>c</i> )	In UltraLite nicht verfügbar.
NVARCHAR(MAX)	LONG NVARCHAR	In UltraLite nicht verfügbar.
NUMERIC( <i>p,s</i> )	NUMERIC( <i>p,s</i> )	

Microsoft SQL Server-Datentyp	Datentyp in SQL Anywhere oder UltraLite	Hinweise
REAL	REAL	REAL kann Probleme verursachen, wenn die konsolidierten und entfernten Datenbanken nicht die exakt gleichen (ungenauen) Werte erlauben. Da nicht alle möglichen Werte getestet werden, ist Vorsicht angebracht. Um Probleme zu vermeiden, sollten Sie diese Typen nicht als Teil eines Primärschlüssels verwenden.
SMALLDATETIME	SMALLDATETIME	In SQL Anywhere und UltraLite wird SMALLDATETIME als TIMESTAMP implementiert. SMALLDATETIME-Werte von Microsoft SQL Server sind auf die Minute genau. 29,998 Sekunden oder weniger werden auf die Minute abgerundet, und Werte mit 29,999 Sekunden oder höher werden auf die nächste Minute aufgerundet. In SQL Anywhere oder UltraLite ist SMALLDATETIME auf die Mikrosekunde genau. Für eine einwandfreie Synchronisation in SQL Anywhere oder UltraLite muss SMALLDATETIME auf die Minute gerundet werden. Das Jahr muss im Bereich von 1900 - 2078 liegen.
SMALLINT	SMALLINT	
SMALLMONEY	SMALLMONEY	
TEXT	LONG VARCHAR	
TIME	TIME	DATETIME2- und TIME-Werte von Microsoft SQL Server sind bis auf 100 Nanosekunden genau. TIMESTAMP- und TIME-Werte sind dagegen nur bis auf 1 Mikrosekunde genau. Für eine einwandfreie Synchronisation von DATETIME2 und TIME sollten Sie die Sekundenbruchteile auf 1 Mikrosekunde runden.

Microsoft SQL Server-Datentyp	Datentyp in SQL Anywhere oder UltraLite	Hinweise
TIMESTAMP	VARBINARY(8)	<p>In Microsoft SQL Server ist TIMESTAMP ein Binärzähler, der jeweils nach einer Änderung einer Zeile erhöht wird. Jede Tabelle kann daher eine TIMESTAMP-Spalte enthalten und es ist nicht sinnvoll, sie zu synchronisieren. Wenn sie in einer Synchronisation enthalten sein muss, sollten Sie sie in SQL Anywhere oder UltraLite dem Datentyp VARBINARY(8) zuordnen.</p> <p>Es ist nicht möglich, diese Zeitstempel-Spalte explizit einzufügen oder zu aktualisieren, da sie vom Server verwaltet wird. Berücksichtigen Sie dies, wenn Sie Uploadskripten für Tabellen implementieren, die solche Spalten enthalten.</p>
DATETIMEOFFSET	TIMESTAMP WITH TIME ZONE	
TINYINT	TINYINT	
UNIQUEIDENTIFIER	UNIQUEIDENTIFIER	
VARBINARY( <i>n</i> )	VARBINARY( <i>n</i> )	
VARBINARY(MAX)	LONG BINARY	
VARCHAR( <i>n</i> )	VARCHAR( <i>n</i> )	
VARCHAR(MAX)	LONG VARCHAR	
XML	XML	

## MySQL-Datenzuordnung

### Zuordnung zu konsolidierten MySQL-Datentypen

Die folgende Tabelle zeigt, wie entfernte SQL Anywhere- und UltraLite-Datentypen konsolidierten MySQL-Datentypen zugeordnet werden. Zum Beispiel sollte eine Spalte, die in der entfernten Datenbank den Datentyp TEXT hat, in der konsolidierten Datenbank den Typ LONGTEXT haben.

Datentyp in SQL Anywhere oder UltraLite	MySQL-Datentyp	Hinweise
BIGINT	BIGINT	
BINARY( $n \leq 255$ )	BINARY( $n$ )	
BINARY( $n > 255$ )	BLOB	
BIT	BIT	
CHAR( $n \leq 255$ )	CHAR( $n$ )	
CHAR( $n > 255$ )	TEXT( $n$ )	
DATE	DATE	Das Jahr muss im Bereich von 1000 bis 9999 liegen.
DATETIME	DATETIME	Der MySQL-Datentyp DATETIME unterstützt keine Sekundenbruchteile. Das Jahr muss im Bereich von 1000 bis 9999 liegen.
DECIMAL( $p \leq 65, s \leq 30$ )	DECIMAL( $p, s$ )	
DECIMAL( $p > 65, s > 30$ )		Es gibt keinen entsprechenden Datentyp in MySQL, wenn die Gesamtstellenzahl größer als 65 oder die Zahl der Dezimalstellen größer als 30 ist.
DOUBLE	DOUBLE	
FLOAT	FLOAT	
IMAGE	LOB	
INTEGER	INTEGER	
LONG BINARY	LOB	
LONG NVARCHAR	LONGTEXT CHARACTER SET UTF8	
LONG VARBIT	LONGTEXT	
LONG VARCHAR	LONGTEXT	
MONEY	NUMERIC(19,4)	

Datentyp in SQL Anywhere oder UltraLite	MySQL-Datentyp	Hinweise
NCHAR( $n \leq 255$ )	CHAR( $n$ ) CHARACTER SET UTF8	
NCHAR( $n > 255$ )	TEXT CHARACTER SET UTF8	
NTEXT	LONGTEXT CHARACTER SET UTF8	
NUMERIC( $p \leq 65, s \leq 30$ )	DECIMAL( $p, s$ )	
NUMERIC( $p > 65, s > 30$ )		Es gibt keinen entsprechenden Datentyp in MySQL.
NVARCHAR( $n$ )	VARCHAR( $n$ ) CHARACTER SET UTF8	
REAL	REAL	
SMALLDATETIME	DATETIME	Der MySQL-Datentyp DATETIME unterstützt keine Sekundenbruchteile. Das Jahr muss im Bereich von 1000 bis 9999 liegen.
SMALLINT	SMALLINT	
SMALLMONEY	NUMERIC(10,4)	
ST_GEOMETRY	GEOMETRY	
TEXT	LONGTEXT	
TIME	TIME	Der MySQL-Datentyp TIME unterstützt keine Sekundenbruchteile.
TIMESTAMP	DATETIME	Der MySQL-Datentyp DATETIME unterstützt keine Sekundenbruchteile. Das Jahr muss im Bereich von 1000 bis 9999 liegen.

Datentyp in SQL Anywhere oder UltraLite	MySQL-Datentyp	Hinweise
TIMESTAMP WITH TIME ZONE	VARCHAR(34)	Es gibt keinen äquivalenten Datentyp in MySQL. Aus diesem Grund sollte eine TIMESTAMP WITH TIME ZONE-Spalte einer VARCHAR(34)-Spalte zugeordnet werden. Beim Upload konvertiert der MobiLink-Server zunächst die Daten in eine Zeichenfolge mit dem Format JJJJ-MM-TT HH:NN:SS.SSSSSS [+-]HH:NN und übernimmt sie anschließend in die konsolidierte Datenbank. Beim Download konvertiert er die Daten aus der Zeichenfolge in das Format TIMESTAMP WITH TIME ZONE. Vergewissern Sie sich, dass die Daten in der konsolidierten Datenbank diesem Format entsprechen, sonst schlägt der Download fehl.
TINYINT	TINYINT UNSIGNED	TINYINT ist in SQL Anywhere und UltraLite immer ohne Vorzeichen.
UNIQUEIDENTIFIER	CHAR(36)	
UNIQUEIDENTIFIERSTR	CHAR(36)	
VARBINARY( <i>n</i> )	VARCHAR( <i>n</i> )	
VARBIT( <i>n</i> ≤8000)	VARCHAR( <i>n</i> )	
VARBIT( <i>n</i> >8000)	TEXT	
VARCHAR( <i>n</i> )	VARCHAR( <i>n</i> )	
XML	LONGTEXT	

### Zuordnung zu entfernten SQL Anywhere- oder UltraLite-Datentypen

Die folgende Tabelle zeigt, wie konsolidierte MySQL-Datentypen entfernten SQL Anywhere- und UltraLite-Datentypen zugeordnet werden. Zum Beispiel sollte eine Spalte, die in der konsolidierten Datenbank den Datentyp BOOL hat, in der entfernten Datenbank den Typ BIT haben.

MySQL-Datentyp	Datentyp in SQL Anywhere oder UltraLite	Hinweise
BIGINT	BIGINT	

MySQL-Datentyp	Datentyp in SQL Anywhere oder UltraLite	Hinweise
BINARY( <i>n</i> )	BINARY( <i>n</i> )	
BIT(1)	BIT	
BIT( <i>n</i> >1)	UNSIGNED BIGINT	
BLOB( <i>n</i> <=32767)	VARBINARY( <i>n</i> )	
BLOB( <i>n</i> >32767)	IMAGE	
BOOL	BIT	
CHAR( <i>n</i> )	CHAR( <i>n</i> )	
DATE	DATE	Das Jahr muss im Bereich von 1000 bis 9999 liegen.
DATETIME	DATETIME	Der MySQL-Datentyp DATETIME unterstützt keine Sekundenbruchteile. Das Jahr muss im Bereich von 1000 bis 9999 liegen.
DOUBLE	DOUBLE	
DECIMAL	DECIMAL	
ENUM		Es gibt keinen entsprechenden Datentyp in SQL Anywhere oder UltraLite.
GEOMETRY	ST_GEOMETRY	
INTEGER	INTEGER	
LINESTRING		Es gibt keinen entsprechenden Datentyp in SQL Anywhere oder UltraLite.
LOBLOB	IMAGE	
LONGTEXT	TEXT	
MEDIUMBLOB	IMAGE	
MEDIUMINT	INTEGER	
MEDIUMTEXT	TEXT	

MySQL-Datentyp	Datentyp in SQL Anywhere oder UltraLite	Hinweise
MULTILINESTRING		Es gibt keinen entsprechenden Datentyp in SQL Anywhere oder UltraLite.
MULTIPOINT		Es gibt keinen entsprechenden Datentyp in SQL Anywhere oder UltraLite.
MULTIPOLYGON		Es gibt keinen entsprechenden Datentyp in SQL Anywhere oder UltraLite.
NCHAR	NCHAR	In UltraLite nicht verfügbar.
NUMERIC	NUMERIC	
NVARCHAR	NVARCHAR	In UltraLite nicht verfügbar.
POINT		Es gibt keinen entsprechenden Datentyp in SQL Anywhere oder UltraLite.
POLYGON		Es gibt keinen entsprechenden Datentyp in SQL Anywhere oder UltraLite.
REAL	REAL	
SET		Es gibt keinen entsprechenden Datentyp in SQL Anywhere oder UltraLite.
SMALLINT	SMALLINT	
TEXT( $n \leq 32767$ )	VARCHAR( $n$ )	
TEXT( $n > 32767$ )	TEXT	
TIME	TIME	Der MySQL-Datentyp TIME unterstützt keine Sekundenbruchteile. Der Bereich von TIME in MySQL ist '-838:59:59' bis '838:59:59'. Der Bereich von TIME in SQL Anywhere oder UltraLite ist '00:00:00.000000' bis '23:59:59.999999'.

MySQL-Datentyp	Datentyp in SQL Anywhere oder UltraLite	Hinweise
TIMESTAMP	TIMESTAMP	Der MySQL-Datentyp DATETIME unterstützt keine Sekundenbruchteile. Das Jahr muss im Bereich von 1000 bis 9999 liegen. Wenngleich MySQL die automatische Initialisierung und Aktualisierung von Zeitstempel-Spalten bereitstellt, bieten SQL Anywhere und UltraLite nur die automatische Initialisierung.
TINYBLOB	VARBINARY	
TINYINT	SMALLINT	TINYINT ist in SQL Anywhere und UltraLite immer ohne Vorzeichen. Muss ein positiver Wert sein.
TINYINT UNSIGNED	TINYINT	TINYINT ist in SQL Anywhere und UltraLite immer ohne Vorzeichen.
TINYTEXT	VARCHAR	
VARBINARY( $n \leq 32767$ )	VARBINARY( $n$ )	
VARBINARY( $n > 32767$ )	IMAGE	
VARCHAR( $n \leq 32767$ )	VARCHAR( $n$ )	
VARCHAR( $n > 32767$ )	TEXT	
YEAR[(2 4)]	INTEGER	SQL Anywhere und UltraLite unterstützen den YEAR-Datentyp nicht. YEAR muss in einer entfernten Datenbank dem Datentyp INTEGER zugeordnet werden. Der INTEGER-Wert muss im Bereich von 1000 bis 9999 liegen.

## Oracle-Datentypzuordnung

### Zuordnung zu konsolidierten Oracle-Datentypen

Die folgende Tabelle zeigt, wie entfernte SQL Anywhere- und UltraLite-Datentypen konsolidierten Oracle-Datentypen zugeordnet werden. Zum Beispiel sollte eine Spalte, die in der entfernten Datenbank den Datentyp BIT hat, in der konsolidierten Datenbank den Typ NUMBER haben.

Datentyp in SQL Anywhere oder UltraLite	Datentyp in Oracle	Hinweise
BIGINT	NUMBER(20)	
BINARY( $n \leq 2000$ )	RAW( $n$ )	
BINARY( $n > 2000$ )	BLOB	Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.
BIT	NUMBER(1)	
CHAR( $n \leq 4000$ )	VARCHAR2( $n$ byte)	Oracle VARCHAR2 ermöglicht Ihnen, die Höchstzahl von Bytes oder Zeichen festzulegen. Die maximale Länge von VARCHAR2-Daten beträgt 4000 Bytes. Falls Sie die Anzahl der Zeichen festlegen, müssen Sie sicherstellen, dass die maximale Datenlänge nicht über 4000 Byte beträgt.
CHAR( $n > 4000$ )	CLOB	Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.
DATE	DATE <sup>2</sup> oder TIME-STAMP	<p>Sekundenbruchteile von SQL Anywhere oder UltraLite können nicht beibehalten werden, wenn ein Oracle DATE-Datentyp verwendet wird, der nicht über Sekundenbruchteile verfügt. Um Probleme zu vermeiden, sollten Sie keine Sekundenwerte mit Bruchteilen verwenden. Das Jahr muss im Bereich von 1 - 9999 liegen.</p> <p>Wenn Sie das Dienstprogramm Interactive SQL verwenden, deaktivieren Sie die Option Return_date_time_as_string, bevor die SQL-Anweisung ausgeführt wird.</p>

Datentyp in SQL Anywhere oder UltraLite	Datentyp in Oracle	Hinweise
DATETIME	DATE <sup>2</sup> oder TIME-STAMP	<p>Sekundenbruchteile von SQL Anywhere oder UltraLite können nicht beibehalten werden, wenn ein Oracle DATE-Datentyp verwendet wird, der nicht über Sekundenbruchteile verfügt. Um Probleme zu vermeiden, sollten Sie keine Sekundenwerte mit Bruchteilen verwenden. Das Jahr muss im Bereich von 1 - 9999 liegen.</p> <p>Wenn Sie das Dienstprogramm Interactive SQL verwenden, deaktivieren Sie die Option Return_date_time_as_string, bevor die SQL-Anweisung ausgeführt wird.</p>
DECIMAL( $p \leq 38, s$ )	NUMBER( $p, 0 \leq s \leq 38$ )	Beim SQL Anywhere-Datentyp DECIMAL liegt $p$ im Bereich von 1 bis 127 und $s$ ist immer kleiner oder gleich $p$ . Beim Oracle-Datentyp NUMBER liegt $p$ im Bereich von 1 bis 38 und $s$ im Bereich von -84 bis 127. Für die Synchronisation muss der Bereich des Oracle-Datentyps NUMBER auf 0 bis 38 eingeschränkt werden.
DECIMAL( $p > 38, s$ )		Es gibt keinen entsprechenden Datentyp in Oracle.
DOUBLE	DOUBLE PRECISION oder BINARY_DOUBLE <sup>1</sup>	Die speziellen Werte INF, -INF und NAN der Oracle Database 10g-Datentypen BINARY_FLOAT und BINARY_DOUBLE können nicht mit SQL Anywhere oder UltraLite synchronisiert werden.
FLOAT( $p$ )	FLOAT( $p$ )	
IMAGE	BLOB	Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.
INTEGER	INT	
LONG BINARY	BLOB	Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.

Datentyp in SQL Anywhere oder UltraLite	Datentyp in Oracle	Hinweise
LONG NVARCHAR	NCLOB	<p>Oracle CLOB und NCLOB können bis zu 4 GB Daten enthalten. SQL Anywhere LONG VARCHAR und LONG NVARCHAR können nur bis zu 2 GB Daten enthalten.</p> <p>Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.</p>
LONG VARBIT	CLOB	<p>Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.</p>
LONG VARCHAR	CLOB	<p>Oracle CLOB und NCLOB können bis zu 4 GB Daten enthalten. SQL Anywhere LONG VARCHAR und LONG NVARCHAR können nur bis zu 2 GB Daten enthalten.</p> <p>Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.</p>
MONEY	NUMBER(19,4)	
NCHAR(c)	NVARCHAR2(c char) oder NCLOB	<p>Die Größe des SQL Anywhere-Datentyps NCHAR und des Oracle-Datentyps NVARCHAR2 zeigt die Höchstzahl von Unicode-Zeichen an. Die Länge des Oracle-Datentyps NVARCHAR2 kann nicht mehr als 4000 Byte betragen. Es ist schwierig, die maximale Bytelänge aus der Zeichengröße zu berechnen. Allgemein gilt: Bei einer Größe von über 1000 sollte zu NCLOB, andernfalls zu NVARCHAR2 zugeordnet werden.</p>

Datentyp in SQL Anywhere oder UltraLite	Datentyp in Oracle	Hinweise
NTEXT	NCLOB	<p>Oracle NCLOB können bis zu 4 GB Daten enthalten. Der SQL Anywhere-Datentyp NTEXT (oder LONG NVARCHAR) kann nur bis zu 2 GB Daten enthalten.</p> <p>Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.</p>
NUMERIC( $p \leq 38, s$ )	NUMBER( $p, 0 \leq s \leq 38$ )	Beim SQL Anywhere-Datentyp NUMERIC liegt $p$ im Bereich von 1 bis 127 und $s$ ist immer kleiner oder gleich $p$ . Beim Oracle-Datentyp NUMBER liegt $p$ im Bereich von 1 bis 38 und $s$ im Bereich von -84 bis 127. Für die Synchronisation muss der Bereich des Oracle-Datentyps NUMBER auf 0 bis 38 eingeschränkt werden.
NUMERIC( $p > 38, s$ )		Es gibt keinen entsprechenden Datentyp in Oracle.
NVARCHAR	NVARCHAR2( $c$ char) oder NCLOB	Die Größe des SQL Anywhere-Datentyps NCHAR und des Oracle-Datentyps NVARCHAR2 zeigt die Höchstzahl von Unicode-Zeichen an. Die Länge des Oracle-Datentyps NVARCHAR2 kann nicht mehr als 4000 Byte betragen. Es ist schwierig, die maximale Bytelänge aus der Zeichengröße zu berechnen. Allgemein gilt: Bei einer Größe von über 1000 sollte zu NCLOB, andernfalls zu NVARCHAR2 zugeordnet werden.
REAL	REAL oder BINARY_FLOAT <sup>1</sup>	Die speziellen Werte INF, -INF und NAN der Oracle Database 10g-Datentypen BINARY_FLOAT und BINARY_DOUBLE können nicht mit SQL Anywhere oder UltraLite synchronisiert werden.

Datentyp in SQL Anywhere oder UltraLite	Datentyp in Oracle	Hinweise
SMALLDATETIME	DATE <sup>2</sup> oder TIME-STAMP	Sekundenbruchteile von SQL Anywhere oder UltraLite können nicht beibehalten werden, wenn ein Oracle DATE-Datentyp verwendet wird, der nicht über Sekundenbruchteile verfügt. Um Probleme zu vermeiden, sollten Sie keine Sekundenwerte mit Bruchteilen verwenden. Das Jahr muss im Bereich von 1 - 9999 liegen.
SMALLINT	NUMBER(5)	
SMALLMONEY	NUMBER(10,4)	
ST_GEOMETRY	SDO_GEOMETRY	
TEXT	CLOB	<p>Oracle CLOB kann bis zu 4 GB Daten enthalten. Der SQL Anywhere-Datentyp TEXT (oder LONG VARCHAR) kann nur bis zu 2 GB Daten enthalten.</p> <p>Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.</p>
TIME	DATE <sup>2</sup> oder TIME-STAMP	<p>Sekundenbruchteile von SQL Anywhere oder UltraLite können nicht beibehalten werden, wenn ein Oracle DATE-Datentyp verwendet wird, der nicht über Sekundenbruchteile verfügt. Um Probleme zu vermeiden, sollten Sie keine Sekundenwerte mit Bruchteilen verwenden.</p> <p>Wenn Sie das Dienstprogramm Interactive SQL verwenden, deaktivieren Sie die Option Return_date_time_as_string, bevor die SQL-Anweisung ausgeführt wird.</p>

Datentyp in SQL Anywhere oder UltraLite	Datentyp in Oracle	Hinweise
TIMESTAMP	DATE <sup>2</sup> oder TIME-STAMP	<p>Sekundenbruchteile von SQL Anywhere oder UltraLite können nicht beibehalten werden, wenn ein Oracle DATE-Datentyp verwendet wird, der nicht über Sekundenbruchteile verfügt. Um Probleme zu vermeiden, sollten Sie keine Sekundenwerte mit Bruchteilen verwenden. Das Jahr muss im Bereich von 1 - 9999 liegen.</p> <p>Wenn Sie das Dienstprogramm Interactive SQL verwenden, deaktivieren Sie die Option Return_date_time_as_string, bevor die SQL-Anweisung ausgeführt wird.</p>
TIMESTAMP WITH TIME ZONE	TIMESTAMP WITH TIME ZONE	
TINYINT	NUMBER(3)	Bei Downloads dürfen Oracle-Werte nicht negativ sein.
UNSIGNED BIGINT	NUMBER(20)	Bei Downloads dürfen Oracle-Werte nicht negativ sein.
UNSIGNED INTEGER	NUMBER(11)	Bei Downloads dürfen Oracle-Werte nicht negativ sein.
UNSIGNED SMALLINT	NUMBER(5)	Bei Downloads dürfen Oracle-Werte nicht negativ sein.
UNSIGNED TINYINT	NUMBER(3)	Bei Downloads dürfen Oracle-Werte nicht negativ sein.
UNIQUEIDENTIFIER	CHAR(36)	
UNIQUEIDENTIFIERSTR	CHAR(36)	Es wird empfohlen, UNIQUEIDENTIFIERSTR nicht für Oracle zu verwenden. Verwenden Sie stattdessen UNIQUEIDENTIFIER.
VARBINARY( $n \leq 2000$ )	RAW( $n$ )	
VARBINARY( $n > 2000$ )	BLOB	Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.

Datentyp in SQL Anywhere oder UltraLite	Datentyp in Oracle	Hinweise
VARBIT( $n \leq 4000$ )	VARCHAR2( $n$ byte)	
VARBIT( $n > 4000$ )	CLOB	Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.
VARCHAR( $n \leq 4000$ )	VARCHAR2( $n$ byte)	Oracle VARCHAR2 ermöglicht Ihnen, die Höchstzahl von Bytes oder Zeichen festzulegen. Die maximale Länge von VARCHAR2-Daten beträgt 4000 Bytes. Falls Sie die Anzahl der Zeichen festlegen, müssen Sie sicherstellen, dass die maximale Datenlänge nicht über 4000 Byte beträgt.
VARCHAR( $n > 4000$ )	CLOB	Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.
XML	XMLTYPE	Der Oracle-Datenbankserver überprüft die Syntax der XMLTYPE-Daten, SQL Anywhere dagegen nicht.  In UltraLite nicht verfügbar.  Siehe <a href="#">XMLTYPE-Datentyp in Oracle auf Seite 23</a> .

<sup>1</sup> Gilt nur für Oracle Database 10g oder höher.

<sup>2</sup> Gilt nur für Oracle-Version 8i.

#### Hinweis

Die LONG-Datentypen werden in Oracle 8, 8i und 9i nicht mehr unterstützt.

### Zuordnung zu entfernten SQL Anywhere- oder UltraLite-Datentypen

Die folgende Tabelle zeigt, wie konsolidierte Oracle-Datentypen entfernten SQL Anywhere- oder UltraLite-Datentypen zugeordnet werden. Zum Beispiel sollte eine Spalte, die in der konsolidierten Datenbank den Datentyp LONG hat, in der entfernten Datenbank den Typ LONG VARCHAR haben.

Datentyp in Oracle	Datentyp in SQL Anywhere oder UltraLite	Hinweise
BFILE	LONG BINARY	Nur Download.  Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.
BINARY_DOUBLE	DOUBLE	Die speziellen Werte INF, -INF und NAN von BINARY_FLOAT können nicht mit SQL Anywhere oder UltraLite synchronisiert werden. Die Genauigkeit von FLOAT und DOUBLE in Oracle unterscheidet sich von derjenigen in SQL Anywhere und UltraLite. Der Wert der Daten kann sich abhängig von der Genauigkeit ändern.
BINARY_FLOAT	REAL	Die speziellen Werte INF, -INF und NAN von BINARY_FLOAT können nicht mit SQL Anywhere oder UltraLite synchronisiert werden. Die Genauigkeit von FLOAT und DOUBLE in Oracle unterscheidet sich von derjenigen in SQL Anywhere und UltraLite. Der Wert der Daten kann sich abhängig von der Genauigkeit ändern.
BLOB	LONG BINARY	Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.
CHAR( <i>n</i> byte)	VARCHAR( <i>n</i> )	SQL Anywhere CHAR und Oracle CHAR sind nicht äquivalent. SQL Anywhere CHAR ist äquivalent zu VARCHAR. Sie sollten CHAR/NCHAR nicht in konsolidierten Datenbankspalten verwenden, die synchronisiert sind. Falls Sie Nicht-SQL Anywhere CHAR verwenden müssen, führen Sie den MobiLink-Server mit der Option -b aus.  SQL Anywhere- oder UltraLite-Werte können länger sein als Oracle-Werte. Stellen Sie daher sicher, dass beim Upload keine zu großen Werte auftauchen.

Datentyp in Oracle	Datentyp in SQL Anywhere oder UltraLite	Hinweise
CLOB	LONG VARCHAR	Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.
DATE	TIMESTAMP	Das Jahr muss im Bereich von 1 - 9999 liegen.
INTERVAL YEAR( <i>Jahr_Gesamtstellenzahl</i> ) TO MONTH		Es gibt keinen entsprechenden Datentyp in SQL Anywhere oder UltraLite.
INTERVAL DAY( <i>Tag_Gesamtstellenzahl</i> ) TO SECOND( <i>p</i> )		Es gibt keinen entsprechenden Datentyp in SQL Anywhere oder UltraLite.
LONG	LONG VARCHAR	
LONG RAW	LONG BINARY	
NCHAR( <i>c</i> char)	NVARCHAR( <i>c</i> )	<p>SQL Anywhere NCHAR und Oracle NCHAR sind nicht äquivalent. SQL Anywhere NCHAR ist äquivalent zu NVARCHAR. Sie sollten NCHAR nicht in konsolidierten Datenbankspalten verwenden, die synchronisiert sind. Falls Sie Nicht-SQL Anywhere NCHAR verwenden müssen, führen Sie den MobiLink-Server mit der Option -b aus.</p> <p>SQL Anywhere- oder UltraLite-Werte können länger sein als Oracle-Werte. Stellen Sie daher sicher, dass beim Upload keine zu großen Werte auftauchen.</p>
NCLOB	LONG NVARCHAR	<p>In UltraLite nicht verfügbar.</p> <p>Oracle-Werte können länger sein als SQL Anywhere- oder UltraLite-Werte. Stellen Sie daher sicher, dass beim Download keine zu großen Werte auftauchen.</p>

Datentyp in Oracle	Datentyp in SQL Anywhere oder UltraLite	Hinweise
NUMBER( <i>p,s</i> )	NUMBER( <i>p,s</i> )	Beim SQL Anywhere-Datentyp NUMBER liegt <i>p</i> im Bereich von 1 bis 127 und <i>s</i> ist immer kleiner oder gleich <i>p</i> . Beim Oracle-Datentyp NUMBER liegt <i>p</i> im Bereich von 1 bis 38 und <i>s</i> im Bereich von -84 bis 127. Für die Synchronisation muss die Anzahl der Dezimalstellen des Oracle-Datentyps NUMBER im Bereich von 0 bis 38 liegen.
NVARCHAR2( <i>c</i> char)	NVARCHAR( <i>c</i> )	In UltraLite nicht verfügbar.  SQL Anywhere- oder UltraLite-Werte können länger sein als Oracle-Werte. Stellen Sie daher sicher, dass beim Upload keine zu großen Werte auftauchen.
RAW	BINARY	SQL Anywhere- oder UltraLite-Werte können länger sein als Oracle-Werte. Stellen Sie daher sicher, dass beim Upload keine zu großen Werte auftauchen.
ROWID	VARCHAR(64)	UROWID und ROWID sind schreibgeschützt und werden daher wahrscheinlich nicht synchronisiert.
SDO_GEOMETRY	ST_GEOMETRY	
TIMESTAMP( <i>p</i> <=6)	TIMESTAMP	Wenn <i>p</i> <6 ist, müssen Sie möglicherweise sicherstellen, dass SQL Anywhere- oder UltraLite-Werte die gleiche Gesamtstellenzahl haben. Andernfalls könnte es sein, dass die Konflikterkennung fehlschlägt bzw. dass doppelte Zeilen vorkommen. Das Jahr muss im Bereich von 1 - 9999 liegen.
TIMESTAMP( <i>p</i> >6)		Es gibt keinen entsprechenden Datentyp in SQL Anywhere oder UltraLite.
TIMESTAMP( <i>p</i> ) WITH LOCAL TIME ZONE		Es gibt keinen entsprechenden Datentyp in SQL Anywhere oder UltraLite.
TIMESTAMP( <i>p</i> <=6) WITH TIME ZONE	TIMESTAMP WITH TIME ZONE	

Datentyp in Oracle	Datentyp in SQL Anywhere oder UltraLite	Hinweise
TIMESTAMP( <i>p</i> ) WITH TIME ZONE		Es gibt keinen entsprechenden Datentyp in SQL Anywhere oder UltraLite.
UROWID	VARCHAR(64)	UROWID und ROWID sind schreibgeschützt und werden daher wahrscheinlich nicht synchronisiert.
VARCHAR2( <i>n</i> byte)	VARCHAR( <i>n</i> )	SQL Anywhere- oder UltraLite-Werte können länger sein als Oracle-Werte. Stellen Sie daher sicher, dass beim Upload keine zu großen Werte auftauchen.
XMLTYPE	XML, LONG VARCHAR oder VARCHAR( <i>n</i> )	Der Oracle-Datenbankserver überprüft die Syntax der XMLTYPE-Daten. SQL Anywhere überprüft nicht den Inhalt der XML-Daten und behandelt die XML-Daten als VARCHAR-Daten. Siehe <a href="#">XMLTYPE-Datentyp in Oracle auf Seite 23</a> .

## Datentypzuordnung für SAP HANA-Datenbankserver

### Zuordnung zu konsolidierten SAP HANA-Datentypen

Die folgende Tabelle zeigt, wie entfernte SQL Anywhere- und UltraLite-Datentypen konsolidierten SAP HANA-Datentypen zugeordnet werden. Zum Beispiel sollte eine Spalte, die in der entfernten Datenbank den Datentyp LONG VARBIT hat, in der konsolidierten Datenbank den Typ LONG VARCHAR haben.

Datentyp in SQL Anywhere oder UltraLite	SAP HANA-Datentyp	Hinweise
BIT	TINYINT	
TINYINT	TINYINT	
SMALLINT	SMALLINT	
UNSIGNED SMALLINT	INTEGER	
INTEGER	INTEGER	
UNSIGNED INTEGER	BIGINT	
BIGINT	BIGINT	

Datentyp in SQL Anywhere oder UltraLite	SAP HANA-Datentyp	Hinweise
UNSIGNED BIGINT	DECIMAL (20,0)	
DECIMAL( <i>p,s</i> )	DECIMAL( <i>p,s</i> )	
NUMERIC( <i>p,s</i> )	DECIMAL( <i>p,s</i> )	
FLOAT	FLOAT	Der FLOAT-Datentyp sollte in entfernten Datenbanken vermieden werden, wenn darin enthaltene Tabellen Daten aus SAP HANA herunterladen.
REAL	REAL	Der REAL-Datentyp sollte in entfernten Datenbanken vermieden werden, wenn darin enthaltene Tabellen Daten aus SAP HANA herunterladen.
DOUBLE	DOUBLE	
SMALLMONEY	DECIMAL(10,4)	
MONEY	DECIMAL (19,4)	
DATE	DATE	
TIME	TIME	Sekundenbruchteile aus SQL Anywhere oder UltraLite können nicht beibehalten werden, wenn ein TIME-Datentyp von SAP HANA verwendet wird, der keine Sekundenbruchteile enthält. Um Probleme zu vermeiden, sollten Sie keine Sekundenbruchteile verwenden.
SMALLDATETIME	TIMESTAMP	
DATETIME	TIMESTAMP	
TIMESTAMP	TIMESTAMP	

Datentyp in SQL Anywhere oder UltraLite	SAP HANA-Datentyp	Hinweise
TIMESTAMP WITH TIME ZONE	VARCHAR(34)	Es gibt keinen äquivalenten Datentyp in SAP HANA, sodass eine TIMESTAMP WITH TIME ZONE-Spalte einer VARCHAR(34)-Spalte zugeordnet werden sollte. Beim Upload konvertiert der MobiLink-Server die Daten in eine Zeichenfolge mit dem Format "JJJJ-MM-TT HH:NN:SS.SSSSSS [+-]HH:NN" und übernimmt sie anschließend in die konsolidierte Datenbank. Beim Download konvertiert der MobiLink-Server die Daten aus einer Zeichenfolge in das Format TIMESTAMP WITH TIME ZONE. Vergewissern Sie sich, dass die Daten in der konsolidierten Datenbank diesem Format entsprechen, um Fehler und ein Fehlschlagen der Synchronisation zu vermeiden.
CHAR( $n \leq 5000$ )	VARCHAR( $n$ )	
CHAR( $n > 5000$ )	CLOB	
VARCHAR( $n \leq 5000$ )	VARCHAR( $n$ )	
VARCHAR( $n > 5000$ )	CLOB	
LONG VARCHAR	CLOB	
NCHAR( $n \leq 5000$ )	NVARCHAR	
NCHAR( $n > 5000$ )	NCLOB	
NVARCHAR( $n \leq 5000$ )	NVARCHAR( $n$ )	
NVARCHAR( $n > 5000$ )	NCLOB	
LONG NVARCHAR	NCLOB	
BINARY( $n \leq 5000$ )	VARBINARY( $n$ )	
BINARY( $n > 5000$ )	BLOB	
VARBINARY( $n \leq 5000$ )	VARBINARY( $n$ )	
VARBINARY( $n > 5000$ )	BLOB	
LONG BINARY	BLOB	

Datentyp in SQL Anywhere oder UltraLite	SAP HANA-Datentyp	Hinweise
VARBIT( $n \leq 5000$ )	VARCHAR( $n$ )	
VARBIT( $n > 5000$ )	CLOB	
LONG VARBIT	CLOB	
GUID	VARCHAR(40)	
GEOMETRY		Nicht unterstützt.

### Zuordnung zu entfernten SQL Anywhere- oder UltraLite-Datentypen

Die folgende Tabelle zeigt, wie konsolidierte SAP HANA-Datentypen entfernten SQL Anywhere- oder UltraLite-Datentypen zugeordnet werden. Zum Beispiel sollte eine Spalte, die in der konsolidierten Datenbank den Datentyp ALPHANUM( $n$ ) hat, in der entfernten Datenbank den Typ VARCHAR( $n$ ) haben.

SAP HANA-Datentyp	Datentyp in SQL Anywhere oder UltraLite	Hinweise
TINYINT	TINYINT	
SMALLINT	SMALLINT	
INTEGER	INTEGER	
BIGINT	BIGINT	
SMALLDECIMAL	DECIMAL( $p,s$ )	In SQL Anywhere liegt DECIMAL $p$ zwischen 1 und 127 und $s$ ist immer kleiner oder gleich $p$ . In SAP HANA liegt $p$ im Bereich von 1 bis 16 und $s$ im Bereich von -369 bis 368.
DECIMAL( $p,s$ )	DECIMAL( $p,s$ )	In SQL Anywhere liegt DECIMAL $p$ zwischen 1 und 127 und $s$ ist immer kleiner oder gleich $p$ . In SAP HANA liegt $p$ im Bereich von 1 bis 34 und $s$ im Bereich von -6111 bis 6176.
FLOAT	DOUBLE	SAP HANA erweitert FLOAT und REAL zu DOUBLE.
REAL	DOUBLE	SAP HANA erweitert FLOAT und REAL zu DOUBLE.

SAP HANA-Datentyp	Datentyp in SQL Anywhere oder UltraLite	Hinweise
DOUBLE	DOUBLE	
DATE	DATE	
TIME	TIME	Sekundenbruchteile aus SQL Anywhere oder UltraLite können nicht beibehalten werden, wenn ein TIME-Datentyp von SAP HANA verwendet wird, der keine Sekundenbruchteile enthält. Um Probleme zu vermeiden, sollten Sie keine Sekundenbruchteile verwenden.
SECONDDATE	TIMESTAMP	Sekundenbruchteile aus SQL Anywhere oder UltraLite können nicht beibehalten werden, wenn ein SECONDDATE-Datentyp von SAP HANA verwendet wird, der keine Sekundenbruchteile enthält. Um Probleme zu vermeiden, sollten Sie keine Sekundenbruchteile verwenden.
TIMESTAMP	TIMESTAMP	
VARCHAR( <i>n</i> )	VARCHAR( <i>n</i> )	
NVARCHAR( <i>n</i> )	NVARCHAR( <i>n</i> )	
ALPHANUM( <i>n</i> )	VARCHAR( <i>n</i> )	
VARBINARY( <i>n</i> )	VARBINARY( <i>n</i> )	
CLOB	LONG VARCHAR	
NCLOB	LONG NVARCHAR	
BLOB	LONG BINARY	

## Sybase IQ Enterprise-Datenzuordnung

### Zuordnung zu konsolidierten SAP Sybase IQ-Datentypen

Die folgende Tabelle zeigt, wie entfernte SQL Anywhere- und UltraLite-Datentypen konsolidierten SAP Sybase IQ-Datentypen zugeordnet werden. Zum Beispiel sollte eine Spalte, die in der entfernten Datenbank den Datentyp LONG VARBIT hat, in der konsolidierten Datenbank den Typ LONG VARCHAR haben.

Datentyp in SQL Anywhere oder UltraLite	SAP Sybase IQ	Hinweise
BIGINT	BIGINT	
BIT	BIT	
BINARY( <i>n</i> )	BINARY( <i>n</i> )	
CHAR( <i>n</i> )	CHAR( <i>n</i> )	Es gibt einige Einschränkungen für CHAR- und VARCHAR-Spalten über 255 Byte. Weitere Hinweise finden Sie in der SAP Sybase IQ-Dokumentation.
DATE	DATE	
DATETIME	DATETIME	
DECIMAL( <i>p,s</i> )	DECIMAL( <i>p,s</i> )	
DOUBLE	DOUBLE	
FLOAT( <i>p</i> )	FLOAT( <i>p</i> )	
INT	INT	
LONG BINARY / IMAGE	LONG BINARY / IMAGE	
LONG NVARCHAR / NTEXT		Dieser Datentyp ist in SAP Sybase IQ nicht verfügbar.
LONG VARBIT	LONG VARCHAR	
LONG VARCHAR / TEXT	TEXT	
MONEY	MONEY	
NCHAR( <i>n</i> )		Dieser Datentyp ist in SAP Sybase IQ nicht verfügbar.
NVARCHAR( <i>n</i> )		Dieser Datentyp ist in SAP Sybase IQ nicht verfügbar.
NUMERIC( <i>p,s</i> )	NUMERIC( <i>p,s</i> )	
SMALLDATETIME	SMALLDATETIME	
SMALLMONEY	SMALLMONEY	

Datentyp in SQL Anywhere oder UltraLite	SAP Sybase IQ	Hinweise
ST_GEOMETRY		Dieser Datentyp ist in SAP Sybase IQ nicht verfügbar.
TIME	TIME	
TIMESTAMP	TIMESTAMP	
TIMESTAMP WITH TIME ZONE	VARCHAR(34)	
TINYINT	TINYINT	
UNIQUEIDENTIFIER	UNIQUEIDENTIFIER	
UNSIGNED BIGINT	UNSIGNED BIGINT	
UNSIGNED INT	UNSIGNED INT	
UNSIGNED SMALLINT	SMALLINT	
UNSIGNED TINYINT	TINYINT	
VARBINARY( <i>n</i> )	VARBINARY( <i>n</i> )	
VARBIT( <i>n</i> )	VARCHAR( <i>n</i> )	
VARCHAR( <i>n</i> )	VARCHAR( <i>n</i> )	Es gibt einige Einschränkungen für CHAR- und VARCHAR-Spalten über 255 Byte. Weitere Hinweise finden Sie in der SAP Sybase IQ-Dokumentation.
XML	LONG BINARY / IMAGE	

### Zuordnung zu entfernten SQL Anywhere- oder UltraLite-Datentypen

Die folgende Tabelle zeigt, wie konsolidierte SAP Sybase IQ-Datentypen entfernten SQL Anywhere- oder UltraLite-Datentypen zugeordnet werden. Zum Beispiel sollte eine Spalte, die in der konsolidierten Datenbank den Datentyp DOUBLE PRECISION hat, in der entfernten Datenbank den Typ DOUBLE haben.

SAP Sybase IQ	Datentyp in SQL Anywhere oder UltraLite	Hinweise
BIGINT	BIGINT	
BINARY( <i>n</i> )	BINARY( <i>n</i> )	
BIT	BIT	
CHAR( <i>n</i> )	VARCHAR( <i>n</i> )	
DATE	DATE	
DATETIME	DATETIME	
DECIMAL( <i>p,s</i> )	DECIMAL( <i>p,s</i> )	
DOUBLE	DOUBLE	
FLOAT( <i>p</i> )	FLOAT( <i>p</i> )	
INT	INT	
LONG BINARY / IMAGE	LONG BINARY / IMAGE	
LONG VARCHAR / TEXT	LONG VARCHAR / TEXT	
MONEY	MONEY	
NUMERIC( <i>p,s</i> )	NUMERIC( <i>p,s</i> )	
REAL	REAL	
SMALLDATETIME	SMALLDATETIME	
SMALLINT	SMALLINT	
SMALLMONEY	SMALLMONEY	
TIME	TIME	
TIMESTAMP	TIMESTAMP	
TINYINT	TINYINT	
UNIQUEIDENTIFIER	UNIQUEIDENTI- FIER	

SAP Sybase IQ	Datentyp in SQL Anywhere oder UltraLite	Hinweise
UNSIGNED BIGINT	UNSIGNED BIGINT	
UNSIGNED INT	UNSIGNED INT	
VARBINARY( <i>n</i> )	VARBINARY( <i>n</i> )	
VARCHAR( <i>n</i> )	VARCHAR( <i>n</i> )	

## Hinweise zum Zeichensatz

Jedes Zeichen eines Textes wird intern durch ein oder mehrere Byte dargestellt. Die Zuordnung von Zeichen zu einem Binärcode wird als **Zeichensatzkodierung** bezeichnet. Einige Zeichensätze für Sprachen mit kleinem Alphabet, beispielsweise für europäische Sprachen, verwenden eine Einbyte-Darstellung. Andere, darunter Unicode, verwenden eine Doppelbyte-Darstellung. Da für jedes Zeichen der doppelte Speicherplatz verwendet wird, können mit Doppelbyte-Zeichen viel mehr Zeichen dargestellt werden.

Es können jedoch Konvertierungsfehler auftreten bzw. Daten verloren gehen, wenn ein Text von einem Zeichensatz in einen anderen konvertiert werden muss. Nicht alle Zeichen können in allen Zeichensätzen dargestellt werden. So können gerade Einbyte-Zeichensätze viel weniger Zeichen darstellen als Mehrbyte-Systeme, da viel weniger Zeichensatzcodes zur Verfügung stehen.

Wenn der Zeichensatz der entfernten MobiLink-Datenbank dem der konsolidierten Datenbank entspricht, treten natürlich keine Konvertierungsprobleme auf.

Die meisten Textdaten müssen sortiert werden, damit Indizes generiert und sortierte Ergebnismengen, wie z.B. Verzeichnislisten, vorbereitet werden können. Die **Sortierreihenfolge** kennzeichnet die Reihenfolge der Zeichen. Eine Sortierreihenfolge gibt in der Regel an, dass der Buchstabe "a" vor dem Buchstaben "b" kommt, der wiederum vor dem "c" zu stehen hat.

Jede Datenbank hat eine **Kollationssequenz**. Diese Kollationssequenz wird beim Erstellen der Datenbank festgelegt. Auf welche Weise dies geschieht, ist allerdings von Datenbanksystem zu Datenbanksystem unterschiedlich. Mithilfe der Kollationssequenz wird sowohl der Zeichensatz als auch die Sortierreihenfolge einer Datenbank definiert.

### Hinweis

Wenn möglich, sollten Sie in der konsolidierten Datenbank und in der entfernten Datenbank die gleiche Kollationssequenz definieren. Damit wird die Gefahr fehlerhafter Konvertierungen vermieden.

### Siehe auch

- SQL Anywhere-Clients: „Internationale Sprachen und Zeichensätze“ [[SQL Anywhere Server - Datenbankadministration](#)]
- UltraLite-Clients: „UltraLite-Zeichensätze“ [[UltraLite - Datenbankverwaltung](#)]
- Hinweise zum RDBMS: „MobiLink - konsolidierte Datenbanken“ auf Seite 1

## Zeichensatzkonvertierung während der Synchronisation

Während der Synchronisation müssen Zeichensätze möglicherweise von einem Zeichensatz in einen anderen konvertiert werden. Die folgenden Konvertierungen sind erforderlich, wenn Zeichen zwischen der entfernten Anwendung und der konsolidierten Datenbank übertragen werden.

### Zeichensatzkonvertierung während des Uploads

Der MobiLink-Client sendet Daten unter Verwendung des Zeichensatzes der entfernten Datenbank zum MobiLink-Server.

1. Der MobiLink-Server kommuniziert mit der konsolidierten Datenbank mithilfe der Unicode ODBC-API. Dazu konvertiert der MobiLink-Server alle von der entfernten Datenbank erhaltenen Zeichen in Unicode und sendet den Unicode an den ODBC-Treiber.
2. Falls erforderlich, konvertiert der ODBC-Treiber für den konsolidierten Datenbankserver die Zeichen von Unicode in den Zeichensatz der konsolidierten Datenbank. Diese Konvertierung wird ausschließlich durch den ODBC-Treiber für das konsolidierte Datenbanksystem gesteuert. Daher kann das Verhalten auf zwei unterschiedlichen Datenbanksystemen auch verschieden sein. Dies gilt besonders für Systeme unterschiedlicher Hersteller. Die MobiLink-Synchronisation kann mit einer Reihe verschiedener Datenbanksysteme ausgeführt werden. Einzelheiten hierzu können Sie der Dokumentation des betreffenden konsolidierten Servers und des ODBC-Treibers entnehmen.

### Zeichensatzkonvertierung während des Downloads

1. Der ODBC-Treiber des konsolidierten Datenbanksystems erhält die Zeichen in der Kodierung der konsolidierten Datenbank. Er konvertiert diese Zeichen in Unicode, um sie über die Unicode-API an den MobiLink-Server zu übertragen. Diese Konvertierung wird ausschließlich durch den ODBC-Treiber für das konsolidierte Datenbanksystem gesteuert. Einzelheiten hierzu können Sie der Dokumentation des betreffenden konsolidierten Servers und des ODBC-Treibers entnehmen.
2. Der MobiLink-Server erhält die Zeichen von der Unicode ODBC-API. Falls die entfernte Datenbank einen unterschiedlichen Zeichensatz verwendet, konvertiert der MobiLink-Server den Zeichensatz vor dem Download.

### Beispiele

- UltraLite-Anwendungen auf Windows Mobile-Geräten verwenden den Unicode-Zeichensatz.  
Wenn Sie eine Windows Mobile-Anwendung synchronisieren, wird innerhalb des MobiLink-Servers keine Zeichenkonvertierung durchgeführt. Der Server stellt fest, dass die von der Anwendung erhaltenen Daten bereits in Unicode sind und leitet sie daher unmittelbar an den ODBC-Treiber weiter. Beim Download von Daten findet in diesem Fall auch keine Zeichensatzkonvertierung statt.

- Bei allen SQL Anywhere-Datenbanken und UltraLite-Anwendungen auf anderen Plattformen als Windows Mobile wird der verwendete Zeichensatz von der Kollatierungssequenz der entfernten Datenbank bestimmt.

Beim Synchronisieren einer entfernten Datenbank konvertiert der MobiLink-Server die Daten vom Zeichensatz der entfernten Datenbank nach Unicode.

## Zeichensatzkonvertierung des ODBC-Treibers

Da die meisten konsolidierten Datenbanken Unicode wohl eher nicht verwenden, sollten Sie verstehen, wie der ODBC-Treiber des konsolidierten Datenbanksystems die betreffenden Daten aus und in Unicode konvertiert. Einige ODBC-Treiber ermitteln den zu verwendenden Zeichensatz anhand der Spracheinstellungen des Computers, auf dem MobiLink ausgeführt wird. In diesen Fällen ist es das Beste, wenn die Sprach- und Codepage-Einstellungen des Computers, auf dem der MobiLink-Server läuft, mit denen der konsolidierten Datenbank übereinstimmen.

Andere ODBC-Treiber, wie z.B. der Treiber für Sybase Adaptive Server Enterprise, ermöglichen für jede einzelne Verbindung das Festlegen eines eigenen Zeichensatzes. Um Konvertierungsfehler zu vermeiden, sollte der von MobiLink verwendete Zeichensatz so eingestellt werden, dass er mit dem der konsolidierten Datenbank übereinstimmt.

Eine detaillierte Erläuterung zur Zeichensatzkonvertierung durch den ODBC-Treiber des Servers der konsolidierten Datenbank können Sie der Dokumentation zum ODBC-Treiber dieses Produkts entnehmen.

## ODBC-Treiber für MobiLink

### Von MobiLink unterstützte ODBC-Treiber

Der MobiLink-Server kann mit einer Vielzahl von konsolidierten Datenbanken und ODBC-Treibern zusammenarbeiten, wie Sie aus der folgenden Tabelle ersehen können. Einige Treiber sind zwar kompatibel für den Einsatz mit MobiLink, können jedoch funktionelle Einschränkungen aufweisen.

Weitere Hinweise zu unterstützten Versionen finden Sie unter <http://www.sybase.com/detail?id=1002288>.

Datenbank	ODBC-Treiber
SQL Anywhere 16	SQL Anywhere 16
Oracle Database 10g oder Oracle Database 11g	SQL Anywhere 16 - Oracle <sup>1</sup>
Microsoft SQL Server	Microsoft SQL Server ODBC-Treiber <sup>2</sup>
Sybase Adaptive Server Enterprise 15.0 oder höher	Sybase Adaptive Server Enterprise-Treiber <sup>2</sup>

Datenbank	ODBC-Treiber
IBM DB2 LUW 9.5. für Windows, Linux und Unix	IBM DB2 9.5-CLI-Treiber <sup>2</sup>
IBM DB2 LUW 9.7. für Windows, Linux und Unix	IBM DB2 9.7-CLI-Treiber <sup>2</sup>
MySQL 5.1	MySQL ODBC-Treiber 5.1 <sup>2</sup>

<sup>1</sup> Enthalten in SQL Anywhere-Version 16. Siehe <http://www.sybase.com/detail?id=1011880>.

<sup>2</sup> Nicht in SQL Anywhere-Version 16 enthalten. Anweisungen zur Installation und Konfiguration finden Sie unter <http://www.sybase.com/detail?id=1011880>.

## SQL Anywhere 16 - Oracle-ODBC-Treiber

Der SQL Anywhere 16 - Oracle-ODBC-Treiber ist auf die Verwendung mit iAnywhere-Software abgestimmt. Dieser Treiber funktioniert nicht mit der Software anderer Hersteller.

Wenn Sie Oracle mit MobiLink oder Datenfernzugriff verwenden, müssen Sie auf dem Computer mit dem Oracle-Treiber einen Oracle-Client installieren.

Der Oracle-Treiber kann mit dem ODBC-Administrator, der Datei *.odbc.ini* (in Unix) oder dem Dienstprogramm "dbdsn" konfiguriert werden.

In der folgenden Tabelle werden die Konfigurationsoptionen für den Oracle-Treiber aufgeführt.

Windows-ODBC-Datenquellen-Administrator	Konfiguration für dbdsn-Befehlszeile oder Datei <i>.odbc.ini</i>	Beschreibung
Datenquellenname	Für dbdsn verwenden Sie die Option -w.	Ein Name zum Identifizieren der Datenquelle.
Benutzer-ID	<b>UserID</b>  In dbdsn setzen Sie diese Option in der Verbindungszeichenfolge.	Die Standardkennung für die Anmeldung, die die Anwendung für die Verbindung mit Ihrer Oracle-Datenbank verwendet. Wenn Sie dieses Feld leer lassen, werden Sie beim Aufbau der Verbindung danach gefragt.
Kennwort	<b>Password</b>  In dbdsn setzen Sie diese Option in der Verbindungszeichenfolge.	Das Kennwort, das die Anwendung für die Verbindung mit Ihrer Oracle-Datenbank verwendet. Wenn Sie dieses Feld leer lassen, werden Sie beim Aufbau der Verbindung danach gefragt.

Windows-ODBC-Datenquellen-Administrator	Konfiguration für dbdsn-Befehlszeile oder Datei <i>.odbc.ini</i>	Beschreibung
TNS-Dienstname	<b>ServiceName</b>	Der TNS-Dienstname, der in <i>network/admin/tnsnames.ora</i> unter Ihrem Oracle-Installationsverzeichnis gespeichert ist.
Kennwort verschlüsseln	Für dbdsn verwenden Sie die Option <i>-pe</i> .  Für <i>.odbc.ini</i> nicht unterstützt.	Aktivieren Sie dieses Kontrollkästchen, wenn das Kennwort in der Datenquelle verschlüsselt gespeichert werden soll.
Prozedur gibt Ergebnisse zurück oder verwendet VARRAY-Parameter	<b>ProcResults</b>  In dbdsn setzen Sie diese Option in der Verbindungszeichenfolge.  <b>ProcOwner</b>  In dbdsn setzen Sie diese Option in der Verbindungszeichenfolge.	Wählen Sie diese Option, wenn Ihre gespeicherte Prozedur Ergebnisse zurückgeben kann oder wenn die gespeicherten Prozeduren Oracle-VARRAYs verwenden. In der Standardeinstellung ist diese Option nicht aktiviert. Aktivieren Sie dieses Kontrollkästchen, falls es sich bei Ihren Skripten <i>download_cursor</i> oder <i>download_delete_cursor</i> um Aufrufe gespeicherter Prozeduren handelt.  Wenn keine gespeicherten Prozeduren VARRAYs verwenden und keine von ihnen eine Ergebnismenge zurückgibt, deaktivieren Sie dieses Kontrollkästchen, um die Performance zu steigern.
Arraygröße	<b>ArraySize</b>  In dbdsn setzen Sie diese Option in der Verbindungszeichenfolge.	Geben Sie die Größe des Byte-Arrays (in Byte) ein, das für den Prefetch von Zeilen verwendet wird (auf Anweisungsbasis). Standardwert ist "60000". Durch Erhöhen dieses Wertes kann die Abrufperformance auf Kosten zusätzlicher Speicherzuordnung beträchtlich verbessert werden (etwa bei Downloads vom MobiLink-Server).
Verteilte Microsoft-Transaktionen aktivieren	Für dbdsn verwenden Sie die Option <b>enableMSDIC</b> in der Verbindungszeichenfolge.  Für <i>.odbc.ini</i> nicht unterstützt.	Aktivieren Sie dieses Kontrollkästchen, wenn Sie Ihre Transaktionen im Microsoft Distributed Transaction Coordinator eintragen möchten. Wenn die Option aktiviert ist, benötigt der Oracle ODBC-Treiber die Oracle-Binärdatei <i>oramts10.dll</i> für Oracle 10g-Clients bzw. <i>oramts11.dll</i> für Oracle 11g-Clients.

## ODBC-Datenquellen für den Oracle-Treiber unter Windows erstellen

Verwenden Sie diese Prozedur, um eine ODBC-Datenquelle für den Oracle-Treiber unter Windows zu erstellen.

### Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

### Erstellen einer ODBC-Datenquelle für den Oracle-Treiber unter Windows

1. Starten Sie den ODBC-Administrator:
  - Wählen Sie **Start » Programme » SQL Anywhere 16 » Administrationstools » ODBC-Datenquellen-Administrator**.

Der **ODBC-Datenquellen-Administrator** wird aufgerufen.

2. Klicken Sie auf **Hinzufügen**.
3. Wählen Sie **SQL Anywhere 16 - Oracle** und klicken Sie auf **Fertig stellen**.
4. Legen Sie die gewünschten Konfigurationsoptionen fest. Die Felder wurden oben erläutert.
5. Klicken Sie auf **Verbindung testen** und dann auf **OK**.

### Ergebnisse

Die ODBC-Datenquelle für den Oracle-Treiber wird erstellt.

### Nächste Schritte

Verwenden Sie die ODBC-Datenquelle, um die Verbindung herzustellen.

## Unix-Konfiguration

Wenn Sie unter Unix den Treiber in einer ODBC-Systeminformationsdatei einrichten (normalerweise *.odbc.ini*), sollte der Abschnitt für diesen Treiber folgendermaßen aussehen (mit den entsprechenden Werten in jedem Feld):

```
[sample_dsn_using_the_ias_odbc_driver_for_oracle]
Driver=full-path/libdboraodbc12_r.so
UserID=user-id
Password=password
ServiceName=TNS-service-name
ProcResults=[yes|no]
ArraySize=bytes
```

Erklärungen der einzelnen Felder finden Sie in der Tabelle unter „[SQL Anywhere 16 - Oracle-ODBC-Treiber](#)“ auf [Seite 768](#).

## DBDSN-Konfiguration

Verwenden Sie die folgende Syntax, um einen Oracle DSN mit dem Dienstprogramm "dbdsn" zu erstellen:

**dbdsn -w data-source-name -or -c configuration-options**

Beschreibungen der *configuration-options* finden Sie unter „SQL Anywhere 16 - Oracle-ODBC-Treiber“ auf Seite 768.

Zum Beispiel:

```
dbdsn -w MyOracleDSN -or -pe -c
"Userid=dba;Password=sql;ServiceName=abcd;ArraySize=100000;ProcResults=y;enableMSDIC=n"
```

Siehe „Datenquellen-Dienstprogramm (dbdsn)“ [*SQL Anywhere Server - Datenbankadministration*].

### Siehe auch

- Empfohlene ODBC-Treiber für MobiLink: <http://www.sybase.com/detail?id=1011880>
- „Datenquellen-Dienstprogramm (dbdsn)“ [*SQL Anywhere Server - Datenbankadministration*]

## Deployment von MobiLink-Anwendungen

Das Deployment von MobiLink-Anwendungen umfasst folgende Aktionen:

- Deployment des MobiLink-Servers in eine Produktionsumgebung.
- Deployment von SQL Anywhere MobiLink-Clients.
- Deployment von UltraLite-MobiLink-Clients.

Dieser Abschnitt beschreibt die Dateien, die Sie in das Installationsprogramm Ihrer Anwendung einbeziehen müssen, um die angegebenen Elemente zu verwenden.

Der **Assistent für das Deployment eines Synchronisationsmodells** kann Sie beim Deployment unter Windows unterstützen. Siehe „Deployment-Assistent“ [*SQL Anywhere Server - Programmierung*].

### Hinweis

**Prüfen Sie Ihre Lizenzvereinbarung** Die Weitergabe von Dateien wird durch die Lizenzvereinbarung geregelt. In diesem Kapitel enthaltene Ausführungen können die Bestimmungen Ihrer Lizenzvereinbarung weder aufheben noch ändern. Bevor Sie daher ein Deployment vornehmen, prüfen Sie bitte Ihre Lizenzvereinbarung.

## Bereitstellung des MobiLink-Servers

Die einfachste Möglichkeit, das Deployment eines MobiLink-Servers in einer Produktionsumgebung durchzuführen, besteht darin, eine lizenzierte Kopie von SQL Anywhere auf dem Produktionscomputer zu installieren.

Wenn Sie jedoch einen MobiLink-Server in einem eigenen Installationsprogramm neu verteilen, sollten Sie nur einen Teil der Dateien einbeziehen. In diesem Fall müssen Sie die folgenden Dateien in Ihre Installation einbeziehen.

### Hinweise

- Testen Sie die Dateien auf einem neu aufgesetzten Computer vor der Verteilung.
- Die Dateien, ausgenommen die Beispieldateien, müssen im Installationsverzeichnis von SQL Anywhere installiert werden.
- Die Dateien müssen sich in demselben Verzeichnis befinden, wenn nichts anderes angegeben wurde.
- Wenn ein Speicherort angegeben wird, müssen die Dateien in ein Verzeichnis desselben Namens kopiert werden.
- Unter UNIX müssen Umgebungsvariable für das System so eingerichtet werden, dass es SQL Anywhere-Anwendungen und -Bibliotheken findet. Es wird empfohlen, die geeignete Datei für die Shell, d.h. entweder *sa\_config.sh* oder *sa\_config.csh* (im Verzeichnis *\$SQLANY16/bin32* für 32-Bit-Umgebungen bzw. *\$SQLANY16/bin64* für 64-Bit-Umgebungen) als Vorlage für die Einstellung der erforderlichen Umgebungsvariablen zu verwenden. Umgebungsvariablen, die von *sa\_config*-Dateien eingerichtet werden, sind z.B. *PATH*, *LD\_LIBRARY\_PATH*, *SQLANY16* und *SQLANY16SAMP16*.
- Unter Windows muss die Umgebungsvariable *PATH* für das System so eingerichtet werden, dass es SQL Anywhere-Anwendungen und -Bibliotheken findet. Überprüfen Sie die Variable *PATH*, um sicherzustellen, dass sie das Verzeichnis *%SQLANY16%/bin32* für 32-Bit-Umgebungen bzw. das Verzeichnis *%SQLANY16%/bin64* für 64-Bit-Umgebungen enthält. Wenn beide Einträge vorhanden sind, entfernen Sie den Pfad, der nicht für Ihre Umgebung zutrifft.
- Um die Java-Synchronisationslogik verwenden zu können, müssen Sie JRE 1.6.0 oder später installiert haben und für die grafischen Administrationstools (Sybase Central und MobiLink-Profiler) muss JRE 1.7.0 oder später installiert sein.
- Hinweise zum Deployment von Sybase Central finden Sie unter „[Deployment von Administrationstools](#)“ [*SQL Anywhere Server - Programmierung*].
- Unter Windows steht ein Deployment-Assistent zur Verfügung. Siehe „[Deployment-Assistent](#)“ [*SQL Anywhere Server - Programmierung*].

### 32-Bit-Anwendungen unter Windows

Alle Verzeichnisse sind relativ zum Verzeichnis *%SQLANY16%*. Weitere Details zur Dateistruktur einer 64-Bit-Windows-Umgebung finden Sie unter [64-Bit-Anwendungen unter Windows auf Seite 775](#).

Beschreibung	Windows-Dateien
MobiLink-Server	<ul style="list-style-type: none"> <li>• <i>Bin32\mlodbc16.dll</i></li> <li>• <i>Bin32\mlsrv16.exe</i></li> <li>• <i>Bin32\mlsrv16.lic</i></li> <li>• <i>Bin32\mlsql16.dll</i></li> <li>• <i>Bin32\dbicu16.dll</i></li> <li>• <i>Bin32\dbicudt16.dll</i></li> </ul>
Sprachenbibliothek	<ul style="list-style-type: none"> <li>• <i>Bin32\dblgen16.dll<sup>1</sup></i></li> </ul>
JavaSynchronisationslogik	<ul style="list-style-type: none"> <li>• <i>Java\activation.jar<sup>2</sup></i></li> <li>• <i>Java\imap.jar<sup>2</sup></i></li> <li>• <i>Java\jodbc4.jar</i></li> <li>• <i>Java\mailapi.jar<sup>2</sup></i></li> <li>• <i>Java\mlscript.jar</i></li> <li>• <i>Java\mlsupport.jar</i></li> <li>• <i>Java\pop3.jar<sup>2</sup></i></li> <li>• <i>Java\smtp.jar<sup>2</sup></i></li> <li>• <i>Bin32\mljava16.dll</i></li> <li>• <i>Bin32\dbjodbc16.dll</i></li> <li>• <i>Bin32\mljodbc16.dll</i></li> </ul>
.NET-Synchronisationslogik	<ul style="list-style-type: none"> <li>• <i>MobiLink\Setup\Dnet\mlDomConfig.xml</i></li> <li>• <i>Bin32\mldnet16.dll</i></li> <li>• <i>Bin32\dnetodbc16.dll</i></li> <li>• <i>Assembly\V2\iAnywhere.MobiLink.dll</i></li> <li>• <i>Assembly\V2\iAnywhere.MobiLink.Script.dll</i></li> <li>• <i>Assembly\V2\iAnywhere.MobiLink.Script.xml</i></li> <li>• <i>Bin32\mlDomConfig.xsd</i></li> </ul>
Sicherheitsoption <sup>3</sup>	<ul style="list-style-type: none"> <li>• <i>Bin32\mlrsa_tls16.dll</i></li> <li>• <i>Bin32\mlrsa_tls_fips16.dll</i></li> <li>• <i>Bin32\sbgs2.dll</i></li> </ul>
Setupskripten (schließen Sie die Skripten für Ihre konsolidierte Datenbank in das Deployment ein)	<ul style="list-style-type: none"> <li>• <i>MobiLink\Setup\</i></li> <li>• <i>MobiLink\Upgrade\</i></li> </ul>
mluser-Dienstprogramm	<ul style="list-style-type: none"> <li>• <i>Bin32\mluser.exe</i></li> <li>• <i>Bin32\mlodbc16.dll</i></li> <li>• <i>Bin32\dbicu16.dll</i></li> <li>• <i>Bin32\dbicudt16.dll</i></li> </ul>
mlstop-Dienstprogramm	<ul style="list-style-type: none"> <li>• <i>Bin32\mlstop.exe</i></li> <li>• <i>Bin32\dbicu16.dll</i></li> </ul>

Beschreibung	Windows-Dateien
mlreplay-Dienstprogramm <sup>6</sup>	<ul style="list-style-type: none"> <li>• <i>Bin32\mlreplay.exe</i></li> <li>• <i>Bin32\mlgenreplayapi.exe</i></li> </ul>
MobiLink-Arbiter	<ul style="list-style-type: none"> <li>• <i>Bin32\dbserv16.dll</i></li> <li>• <i>Bin32\mlarb16.exe</i></li> <li>• <i>Bin32\mlarb16.lic</i></li> <li>• <i>Bin32\mlarbiter.bat</i></li> <li>• <i>MobiLink\mlarbiter.control</i></li> <li>• <i>mlarbstop.exe</i></li> </ul>
MobiLink-Profiler	<ul style="list-style-type: none"> <li>• <i>Java\mlprof.jar</i></li> <li>• <i>Java\mlstream.jar</i></li> <li>• <i>Java\JComponents1600.jar</i></li> <li>• <i>Java\jsyblib1600.jar</i></li> <li>• <i>Bin32\jsyblib1600.dll</i></li> <li>• <i>Bin32\mlprof.exe</i></li> <li>• <i>Bin32\mljstrm16.dll</i></li> </ul> <p>Für die Sicherheit mit dem MobiLink-Profiler:<sup>3</sup></p> <ul style="list-style-type: none"> <li>• <i>Bin32\mlcrsa16.dll</i></li> <li>• <i>Bin32\mlcrsafips16.dll</i></li> <li>• <i>Bin32\mlczlib16.dll</i></li> </ul>
Online-Hilfe für das MobiLink 16-Plug-In und den MobiLink-Profiler	<ul style="list-style-type: none"> <li>• <i>\Documentation\[de]\htmlhelp\sqlanywhere_[de]16.chm<sup>1</sup></i></li> <li>• <i>\Documentation\[de]\htmlhelp\sqlanywhere_[de]16.map<sup>1</sup></i></li> </ul>
Notifier	<ul style="list-style-type: none"> <li>• <i>Java\activation.jar<sup>2</sup></i></li> <li>• <i>Java\jodbc4.jar</i></li> <li>• <i>Java\mailapi.jar<sup>2</sup></i></li> <li>• <i>Java\mlnotif.jar</i></li> <li>• <i>Java\mlscript.jar</i></li> <li>• <i>Java\smtp.jar<sup>2</sup></i></li> <li>• <i>Bin32\mljodbc16.dll</i></li> <li>• <i>Bin32\mljstrm16.dll</i></li> </ul> <p>Für die Sicherheit mit dem Notifier:<sup>3</sup></p> <ul style="list-style-type: none"> <li>• <i>Bin32\mlcrsa16.dll</i></li> <li>• <i>Bin32\mlcrsafips16.dll</i></li> <li>• <i>Bin32\mlczlib16.dll</i></li> </ul>

Beschreibung	Windows-Dateien
Relay Server Outbound Enabler	<ul style="list-style-type: none"> <li>• <i>Bin32\rsoe.exe</i></li> </ul> <p>Zur Sicherheit mit dem Outbound Enabler:</p> <ul style="list-style-type: none"> <li>• <i>Bin32\mlcrsa16.dll</i></li> </ul>
Integrierter Outbound Enabler	<ul style="list-style-type: none"> <li>• <i>Bin32\rsoesupp16.dll</i></li> </ul>

<sup>1</sup> Verwenden Sie für Deutsch, Japanisch und Chinesisch *dblgde16.dll*, *dblgja16.dll* bzw. *dblgzh16.dll*.

<sup>2</sup> Wenn Sie eine Anwendung neu verteilen, müssen Sie diese Dateien direkt von Oracle beziehen.

<sup>3</sup> FIPS erfordert, dass Sie die separat lizenzierte SQL Anywhere-Sicherheitsoption erwerben. Außerdem ist es Exportbeschränkungen unterworfen. RSA-Sicherheit ist in SQL Anywhere für Version 10 und höher enthalten. Hinweise zur Bestellung dieser Komponente finden Sie unter „[Getrennt lizenzierbare Komponenten](#)“ [*SQL Anywhere 16 - Einführung*].

<sup>4</sup> Zum Kompilieren von generiertem Code sind alle Dateien mit dem Präfix **mlreplay** im Verzeichnis *SDK/Include* erforderlich.

## 64-Bit-Anwendungen unter Windows

Alle Verzeichnisse sind relativ zum Verzeichnis *%SQLANY16%*. Weitere Details zur Dateistruktur einer 32-Bit-Windows-Umgebung finden Sie unter [32-Bit-Anwendungen unter Windows auf Seite 772](#).

Beschreibung	Windows-Dateien
MobiLink-Server	<ul style="list-style-type: none"> <li>• <i>Bin64\mlodbc16.dll</i></li> <li>• <i>Bin64\mlsrv16.exe</i></li> <li>• <i>Bin64\mlsrv16.lic</i></li> <li>• <i>Bin64\mlsql16.dll</i></li> <li>• <i>Bin64\dbicu16.dll</i></li> <li>• <i>Bin64\dbicudt16.dll</i></li> </ul>
Sprachenbibliothek	<ul style="list-style-type: none"> <li>• <i>Bin64\dblgcn16.dll</i><sup>1</sup></li> </ul>

Beschreibung	Windows-Dateien
JavaSynchronisationslogik	<ul style="list-style-type: none"> <li>• <i>Java\activation.jar<sup>2</sup></i></li> <li>• <i>Java\imap.jar<sup>2</sup></i></li> <li>• <i>Java\jodbc4.jar</i></li> <li>• <i>Java\mailapi.jar<sup>2</sup></i></li> <li>• <i>Java\mlscript.jar</i></li> <li>• <i>Java\mlsupport.jar</i></li> <li>• <i>Java\pop3.jar<sup>2</sup></i></li> <li>• <i>Java\smtp.jar<sup>2</sup></i></li> <li>• <i>Bin64\mljava16.dll</i></li> <li>• <i>Bin64\dbjodbc16.dll</i></li> <li>• <i>Bin64\mljodbc16.dll</i></li> </ul>
.NET-Synchronisationslogik	<ul style="list-style-type: none"> <li>• <i>MobiLink\Setup\Dnet\mlDomConfig.xml</i></li> <li>• <i>Bin64\mldnet16.dll</i></li> <li>• <i>Bin64\dnetodbc16.dll</i></li> <li>• <i>Assembly\V2\iAnywhere.MobiLink.dll</i></li> <li>• <i>Assembly\V2\iAnywhere.MobiLink.Script.dll</i></li> <li>• <i>Assembly\V2\iAnywhere.MobiLink.Script.xml</i></li> <li>• <i>Bin64\mlDomConfig.xsd</i></li> </ul>
Sicherheitsoption <sup>3</sup>	<ul style="list-style-type: none"> <li>• <i>Bin64\mlrsa_tls16.dll</i></li> <li>• <i>Bin64\mlrsa_tls_fips16.dll</i></li> <li>• <i>Bin64\sbgse2.dll</i></li> </ul>
Setupskripten (schließen Sie die Skripten für Ihre konsolidierte Datenbank in das Deployment ein)	<ul style="list-style-type: none"> <li>• <i>MobiLink\Setup\</i></li> <li>• <i>MobiLink\Upgrade\</i></li> </ul>
mluser-Dienstprogramm	<ul style="list-style-type: none"> <li>• <i>Bin64\mluser.exe</i></li> <li>• <i>Bin64\mlodbc16.dll</i></li> <li>• <i>Bin64\dbicu16.dll</i></li> <li>• <i>Bin64\dbicudt16.dll</i></li> </ul>
mlstop-Dienstprogramm	<ul style="list-style-type: none"> <li>• <i>Bin64\mlstop.exe</i></li> <li>• <i>Bin64\dbicu16.dll</i></li> </ul>
mlreplay-Dienstprogramm <sup>6</sup>	<ul style="list-style-type: none"> <li>• <i>Bin64\mlreplay.exe</i></li> <li>• <i>Bin64\mlgenreplayapi.exe</i></li> </ul>
MobiLink-Arbiter	<ul style="list-style-type: none"> <li>• <i>Bin64\dserv16.dll</i></li> <li>• <i>Bin64\mlarb16.exe</i></li> <li>• <i>Bin64\mlarb16.lic</i></li> <li>• <i>Bin64\mlarbiter.bat</i></li> <li>• <i>MobiLink\mlarbiter.control</i></li> <li>• <i>mlarbstop.exe</i></li> </ul>

Beschreibung	Windows-Dateien
MobiLink-Profiler	<ul style="list-style-type: none"> <li>• <i>Java\mlprof.jar</i></li> <li>• <i>Java\mlstream.jar</i></li> <li>• <i>Java\JComponents1600.jar</i></li> <li>• <i>Java\jsyblib1600.jar</i></li> <li>• <i>Bin64\jsyblib1600.dll</i></li> <li>• <i>Bin64\mlprof.exe</i></li> <li>• <i>Bin64\mljstrm16.dll</i></li> </ul> <p>Für die Sicherheit mit dem MobiLink-Profiler:<sup>3</sup></p> <ul style="list-style-type: none"> <li>• <i>Bin64\mlcrsa16.dll</i></li> <li>• <i>Bin64\mlcrsafips16.dll</i></li> <li>• <i>Bin64\mlczlib16.dll</i></li> </ul>
Online-Hilfe für das MobiLink 16-Plug-In und den MobiLink-Profiler	<ul style="list-style-type: none"> <li>• <i>\Documentation\[de]\htmlhelp\sqlanywhere_[de]16.chm<sup>1</sup></i></li> <li>• <i>\Documentation\[de]\htmlhelp\sqlanywhere_[de]16.map<sup>1</sup></i></li> </ul>
Notifier	<ul style="list-style-type: none"> <li>• <i>Java\activation.jar<sup>2</sup></i></li> <li>• <i>Java\jodbc4.jar</i></li> <li>• <i>Java\mailapi.jar<sup>2</sup></i></li> <li>• <i>Java\mlnotif.jar</i></li> <li>• <i>Java\mlscript.jar</i></li> <li>• <i>Java\smtp.jar<sup>2</sup></i></li> <li>• <i>Bin64\mljodbc16.dll</i></li> <li>• <i>Bin64\mljstrm16.dll</i></li> </ul> <p>Für die Sicherheit mit dem Notifier:<sup>3</sup></p> <ul style="list-style-type: none"> <li>• <i>Bin64\mlcrsa16.dll</i></li> <li>• <i>Bin64\mlcrsafips16.dll</i></li> <li>• <i>Bin64\mlczlib16.dll</i></li> </ul>
Relay Server Outbound Enabler	<ul style="list-style-type: none"> <li>• <i>Bin64\rsoe.exe</i></li> </ul> <p>Zur Sicherheit mit dem Outbound Enabler:</p> <ul style="list-style-type: none"> <li>• <i>Bin64\mlcrsa16.dll</i></li> </ul>
Integrierter Outbound Enabler	<ul style="list-style-type: none"> <li>• <i>Bin64\rsoesupp16.dll</i></li> </ul>

<sup>1</sup> Verwenden Sie für Deutsch, Japanisch und Chinesisch *dblgde16.dll*, *dblgja16.dll* bzw. *dblgzh16.dll*.

<sup>2</sup> Wenn Sie eine Anwendung neu verteilen, müssen Sie diese Dateien direkt von Oracle beziehen.

<sup>3</sup> FIPS erfordert, dass Sie die separat lizenzierte SQL Anywhere-Sicherheitsoption erwerben. Außerdem ist es Exportbeschränkungen unterworfen. RSA-Sicherheit ist in SQL Anywhere für Version 10 und höher enthalten. Hinweise zur Bestellung dieser Komponente finden Sie unter „[Getrennt lizenzierte Komponenten](#)“ [[SQL Anywhere 16 - Einführung](#)].

<sup>4</sup> Zum Kompilieren von generiertem Code sind alle Dateien mit dem Präfix **mlreplay** im Verzeichnis *SDK/Include* erforderlich.

## Unix-Anwendungen (32-Bit) unter Unix, Linux und Mac OS X

Alle Verzeichnisse sind relativ zum Verzeichnis `$SQLANY16`. Weitere Details zur Dateistruktur einer 64-Bit-Unix-Umgebung finden Sie unter [Unix-Anwendungen \(64-Bit\) unter Unix und Linux auf Seite 780](#).

Beschreibung	Unix-Dateien
MobiLink-Server	<ul style="list-style-type: none"> <li><code>bin32/mlsrv16</code></li> <li><code>bin32/mlsrv16.lic</code></li> <li><code>lib32/libdbodm16.so<sup>3</sup></code></li> <li><code>lib32/libmlodbc16_r.so<sup>3</sup></code></li> <li><code>lib32/libmlsql16_r.so<sup>3</sup></code></li> <li><code>lib32/libdbtasks16_r.so<sup>3</sup></code></li> <li><code>lib32/libdbicu16_r.so<sup>3</sup></code></li> <li><code>lib32/libdbicudt16_r.so<sup>3</sup></code></li> <li><code>lib32/libdbodbcinst16_r.so<sup>3</sup></code></li> </ul>
Sprachenbibliothek	<ul style="list-style-type: none"> <li><code>res/dblgen16.res<sup>1</sup></code></li> </ul>
JavaSynchronisationslogik	<ul style="list-style-type: none"> <li><code>java/activation.jar<sup>2</sup></code></li> <li><code>java/imap.jar<sup>2</sup></code></li> <li><code>java/jodbc4.jar</code></li> <li><code>java/mailapi.jar<sup>2</sup></code></li> <li><code>java/mlscript.jar</code></li> <li><code>java/mlsupport.jar</code></li> <li><code>java/pop3.jar<sup>2</sup></code></li> <li><code>java/smtp.jar<sup>2</sup></code></li> <li><code>lib32/libmljava16_r.so<sup>3</sup></code></li> <li><code>lib32/libmljodbc16.so<sup>3</sup></code></li> </ul>
.NET-Synchronisationslogik	<ul style="list-style-type: none"> <li>Nicht anwendbar</li> </ul>
Sicherheitsoption <sup>4</sup>	<ul style="list-style-type: none"> <li><code>lib32/libmlrsa_tls16_r.so<sup>3</sup></code></li> </ul>
Setupskripten (schließen Sie die Skripten für Ihre konsolidierte Datenbank in das Deployment ein)	<ul style="list-style-type: none"> <li><code>mobilink/setup</code></li> <li><code>mobilink/upgrade</code></li> </ul>

Beschreibung	Unix-Dateien
mluser-Dienstprogramm	<ul style="list-style-type: none"> <li>• <i>bin32/mluser</i></li> <li>• <i>lib32/libmlodbc16_r.so<sup>3</sup></i></li> <li>• <i>lib32/libdbicu16.so<sup>3</sup></i></li> <li>• <i>lib32/libdbicudt16.so<sup>3</sup></i></li> </ul>
mlstop-Dienstprogramm	<ul style="list-style-type: none"> <li>• <i>bin32/mlstop</i></li> <li>• <i>lib32/libdbicu16.so<sup>3</sup></i></li> </ul>
mlreplay-Dienstprogramm <sup>6</sup>	<ul style="list-style-type: none"> <li>• <i>bin32/mlreplay</i></li> <li>• <i>bin32/mlgenreplayapi</i></li> </ul>
MobiLink-Arbitrator	<ul style="list-style-type: none"> <li>• <i>bin32/libdbserv16_r.so</i></li> <li>• <i>bin32/mlarb16</i></li> <li>• <i>bin32/mlarb16.lic</i></li> <li>• <i>bin32/mlarbiter.sh</i></li> <li>• <i>mobilink/mlarbiter.control</i></li> <li>• <b>mlarbstop.exe</b></li> </ul>
MobiLink-Profiler	<ul style="list-style-type: none"> <li>• <i>bin32/mlprof</i></li> <li>• <i>java/mlprof.jar</i></li> <li>• <i>java/mlstream.jar</i></li> <li>• <i>java/JComponents1600.jar</i></li> <li>• <i>java/jsyblib1600.jar</i></li> <li>• <i>lib32/libjsyblib1600_r.so<sup>3</sup></i></li> </ul> <p>Für die Sicherheit mit dem Profiler:</p> <ul style="list-style-type: none"> <li>• <i>lib32/libmlcrsa16_r.so</i></li> <li>• <i>lib32/libmlcrsafips16_r.so</i></li> <li>• <i>lib32/libmlczlib16_r.so</i></li> </ul>
Notifier	<ul style="list-style-type: none"> <li>• <i>java/activation.jar<sup>2</sup></i></li> <li>• <i>java/jodbc4.jar</i></li> <li>• <i>java/mailapi.jar<sup>2</sup></i></li> <li>• <i>java/mlnotif.jar</i></li> <li>• <i>java/mlscript.jar</i></li> <li>• <i>java/smtp.jar<sup>2</sup></i></li> <li>• <i>lib32/libmljstrm16_r.so<sup>3</sup></i></li> </ul>
Relay Server Outbound Enabler	<ul style="list-style-type: none"> <li>• <i>bin32\rsoe</i></li> </ul> <p>Zur Sicherheit mit dem Outbound Enabler:</p> <ul style="list-style-type: none"> <li>• <i>lib32/libmlcrsa16_r.so<sup>3</sup></i></li> </ul>
Integrierter Outbound Enabler	<ul style="list-style-type: none"> <li>• <i>lib32\librsoesupp16_r.so</i></li> </ul>

<sup>1</sup> Verwenden Sie für Deutsch, Japanisch und Chinesisch *dblgde16.dll*, *dblgja16.dll* bzw. *dblgzh16.dll*.

<sup>2</sup> Wenn Sie eine Anwendung neu verteilen, müssen Sie diese Dateien direkt von Oracle beziehen.

<sup>3</sup> Für Linux lautet die Dateierweiterung *.so*. Für Mac OS X lautet die Dateierweiterung *.dylib*.

<sup>4</sup> Die Transportschichtssicherheit setzt voraus, dass Sie die getrennt lizenzierte SQL Anywhere-Sicherheitsoption erwerben. Außerdem ist sie Exportbeschränkungen unterworfen. Hinweise zur Bestellung dieser Komponente finden Sie unter „Getrennt lizenzierbare Komponenten“ [[SQL Anywhere 16 - Einführung](#)].

<sup>5</sup> Zum Kompilieren von generiertem Code sind alle Dateien mit dem Präfix **mlreplay** im Verzeichnis *sdk/include* erforderlich.

## Unix-Anwendungen (64-Bit) unter Unix und Linux

Alle Verzeichnisse sind relativ zum Verzeichnis *\$SQLANY16*. Weitere Details zur Dateistruktur einer 32-Bit-Unix-Umgebung finden Sie unter [Unix-Anwendungen \(32-Bit\) unter Unix, Linux und Mac OS X auf Seite 778](#).

Beschreibung	Unix-Dateien
MobiLink-Server	<ul style="list-style-type: none"> <li><i>bin64/mlsrv16</i></li> <li><i>bin64/mlsrv16.lic</i></li> <li><i>lib64/libdbodm16.so<sup>3</sup></i></li> <li><i>lib64/libmlodbc16_r.so<sup>3</sup></i></li> <li><i>lib64/libmlsql16_r.so<sup>3</sup></i></li> <li><i>lib64/libdbtasks16_r.so<sup>3</sup></i></li> <li><i>lib64/libdbicu16_r.so<sup>3</sup></i></li> <li><i>lib64/libdbicudt16_r.so<sup>3</sup></i></li> <li><i>lib64/libdbodbcinst16_r.so<sup>3</sup></i></li> </ul>
Sprachenbibliothek	<ul style="list-style-type: none"> <li><i>res/dblgen16.res<sup>1</sup></i></li> </ul>
JavaSynchronisationslogik	<ul style="list-style-type: none"> <li><i>java/activation.jar<sup>2</sup></i></li> <li><i>java/imap.jar<sup>2</sup></i></li> <li><i>java/jodbc4.jar</i></li> <li><i>java/mailapi.jar<sup>2</sup></i></li> <li><i>java/mlscript.jar</i></li> <li><i>java/mlsupport.jar</i></li> <li><i>java/pop3.jar<sup>2</sup></i></li> <li><i>java/smtp.jar<sup>2</sup></i></li> <li><i>lib64/libmljava16_r.so<sup>3</sup></i></li> <li><i>lib64/libmljodbc16.so<sup>3</sup></i></li> </ul>
.NET-Synchronisationslogik	<ul style="list-style-type: none"> <li>Nicht anwendbar</li> </ul>
Sicherheitsoption <sup>4</sup>	<ul style="list-style-type: none"> <li><i>lib64/libmlrsa_tls16_r.so<sup>3</sup></i></li> </ul>

Beschreibung	Unix-Dateien
Setupskripten (schließen Sie die Skripten für Ihre konsolidierte Datenbank in das Deployment ein)	<ul style="list-style-type: none"> <li>• <i>mobilink/setup</i></li> <li>• <i>mobilink/upgrade</i></li> </ul>
mluser-Dienstprogramm	<ul style="list-style-type: none"> <li>• <i>bin64/mluser</i></li> <li>• <i>lib64/libmlodbc16_r.so<sup>3</sup></i></li> <li>• <i>lib64/libdbicu16.so<sup>3</sup></i></li> <li>• <i>lib64/libdbicudt16.so<sup>3</sup></i></li> </ul>
mlstop-Dienstprogramm	<ul style="list-style-type: none"> <li>• <i>bin64/mlstop</i></li> <li>• <i>lib64/libdbicu16.so<sup>3</sup></i></li> </ul>
mlreplay-Dienstprogramm <sup>6</sup>	<ul style="list-style-type: none"> <li>• <i>bin64/mlreplay</i></li> <li>• <i>bin64/mlgenraplyapi</i></li> </ul>
MobiLink-Arbitrer	<ul style="list-style-type: none"> <li>• <i>bin64/libdbserv16_r.so</i></li> <li>• <i>bin64/mlarb16</i></li> <li>• <i>bin64/mlarb16.lic</i></li> <li>• <i>bin64/mlarbiter.sh</i></li> <li>• <i>mobilink/mlarbiter.control</i></li> <li>• <b>mlarbstop.exe</b></li> </ul>
MobiLink-Profiler	<ul style="list-style-type: none"> <li>• <i>bin64/mlprof</i></li> <li>• <i>java/mlprof.jar</i></li> <li>• <i>java/mlstream.jar</i></li> <li>• <i>java/JComponents1600.jar</i></li> <li>• <i>java/jsyblib1600.jar</i></li> <li>• <i>lib64/libjsyblib1600_r.so<sup>3</sup></i></li> </ul> <p>Für die Sicherheit mit dem Profiler:</p> <ul style="list-style-type: none"> <li>• <i>lib64/libmlcrsa16_r.so</i></li> <li>• <i>lib64/libmlcrsafips16_r.so</i></li> <li>• <i>lib64/libmlczlib16_r.so</i></li> </ul>
Notifier	<ul style="list-style-type: none"> <li>• <i>java/activation.jar<sup>2</sup></i></li> <li>• <i>java/jodbc4.jar</i></li> <li>• <i>java/mailapi.jar<sup>2</sup></i></li> <li>• <i>java/mlnotif.jar</i></li> <li>• <i>java/mlscript.jar</i></li> <li>• <i>java/smtp.jar<sup>2</sup></i></li> </ul>

Beschreibung	Unix-Dateien
Relay Server Outbound Enabler	<ul style="list-style-type: none"> <li><i>bin64\rsoe</i></li> </ul> <p>Zur Sicherheit mit dem Outbound Enabler:</p> <ul style="list-style-type: none"> <li><i>lib64/libmlcrsa16_r.so</i><sup>3</sup></li> </ul>
Integrierter Outbound Enabler	<ul style="list-style-type: none"> <li><i>lib64/librsoesupp16_r.so</i></li> </ul>

<sup>1</sup> Verwenden Sie für Deutsch, Japanisch und Chinesisch *dblgde16.dll*, *dblgja16.dll* bzw. *dblgzh16.dll*.

<sup>2</sup> Wenn Sie eine Anwendung neu verteilen, müssen Sie diese Dateien direkt von Oracle beziehen.

<sup>3</sup> Für Solaris SPARC und Linux lautet die Dateierweiterung *.so*. Für IBM AIX lautet die Dateierweiterung *.a*.

<sup>4</sup> Die Transportschichtssicherheit setzt voraus, dass Sie die getrennt lizenzierte SQL Anywhere-Sicherheitsoption erwerben. Außerdem ist sie Exportbeschränkungen unterworfen. Hinweise zur Bestellung dieser Komponente finden Sie unter „Getrennt lizenzierte Komponenten“ [[SQL Anywhere 16 - Einführung](#)].

<sup>5</sup> Zum Kompilieren von generiertem Code sind alle Dateien mit dem Präfix **mlreplay** im Verzeichnis *sdk/include* erforderlich.

## Bereitstellung von SQL Anywhere MobiLink-Clients

### Hinweise

- Für SQL Anywhere-Clients müssen Sie das Deployment eines SQL Anywhere-Datenbankservers und des MobiLink-Clients durchführen.  
Siehe „Deployment von Datenbanken und Anwendungen“ [[SQL Anywhere Server - Programmierung](#)].
- Wenn Sie MobiLink-Synchronisationsclients neu verteilen, müssen Sie zusätzlich zu den Dateien, die für die SQL Anywhere-Datenbank erforderlich sind, die folgenden Dateien in Ihre Installation einbeziehen.
- Wenn Sie ein Deployment der folgenden Dateien vornehmen, positionieren Sie sie in dieselbe Verzeichnisstruktur, sofern nicht anders angegeben.
- Hinweise zum Deployment von Sybase Central finden Sie unter „Deployment von Administrationstools“ [[SQL Anywhere Server - Programmierung](#)].
- Unter Windows steht ein Deployment-Assistent zur Verfügung. Siehe „Deployment-Assistent“ [[SQL Anywhere Server - Programmierung](#)].
- Für das Windows Mobile-Deployment befinden sich die Dateien, die im Folgenden unter den *bin32*-Verzeichnissen aufgelistet sind, im Verzeichnis *ce\arm.50*. .NET-Assemblys befinden sich im Verzeichnis *ce\Assembly.v2*.

## Windows-Anwendungen

Alle Verzeichnisse sind relativ zum Verzeichnis *%SQLANY16%*. 64-Bit Versionen dieser Dateien befinden sich im Verzeichnis *Bin64*.

Beschreibung	Windows-Dateien
MobiLink-Synchronisationsclient (dbmlsync)	<ul style="list-style-type: none"> <li>• <i>Bin32\dbcon16.dll</i><sup>2</sup></li> <li>• <i>Bin32\dbicu16.dll</i><sup>3</sup></li> <li>• <i>Bin32\dblgen16.dll</i><sup>1</sup></li> <li>• <i>Bin32\dblib16.dll</i></li> <li>• <i>Bin32\dbmlsync.exe</i></li> <li>• <i>Bin32\dbtool16.dll</i><sup>2</sup></li> </ul>
Dbmlsync-C++-API oder die SQL-Anweisung SYNCHRONIZE	<ul style="list-style-type: none"> <li>• MobiLink-Synchronisationsclient-Dateien</li> <li>• <i>Bin32\dbmlsynccli16.dll</i></li> </ul>
Dbmlsync-.NET-API	<ul style="list-style-type: none"> <li>• MobiLink-Synchronisationsclient-Dateien</li> <li>• <i>Assembly\V2\iAnywhere.MobiLink.Client.dll</i></li> </ul>
Sicherheitsoption <sup>2</sup>	<ul style="list-style-type: none"> <li>• <i>Bin32\mlcrsa16.dll</i></li> <li>• <i>Bin32\mlcrsafips16.dll</i></li> <li>• <i>Bin32\sbgse2.dll</i></li> </ul>
Microsoft ActiveSync-Dienstprogramm	<ul style="list-style-type: none"> <li>• <i>Bin32\mlasinst.exe</i></li> <li>• <i>Bin32\mlasdesk.dll</i></li> <li>• <i>Bin32\dbcon16.exe</i></li> <li>• <i>CE\chip\mlasdev.dll</i> (wobei <i>chip</i> jeder unterstützte Chip sein kann, wie z.B. arm.50)</li> </ul>
Listener	<ul style="list-style-type: none"> <li>• <i>Bin32\dblgen16.dll</i><sup>1</sup></li> <li>• <i>Bin32\dblsn.exe</i></li> <li>• <i>Bin32\lsn_udp16.dll</i></li> <li>• <i>Bin32\lsn_swi510.dll</i></li> <li>• <i>Bin32\maac555.dll</i></li> <li>• <i>Bin32\maac750.dll</i></li> <li>• <i>Bin32\maac750r3.dll</i></li> <li>• <i>Bin32\mabridge.dll</i></li> <li>• <i>dblsn_sms16.dll</i><sup>4</sup></li> </ul>

Beschreibung	Windows-Dateien
MobiLink-Agent (für die zentrale Verwaltung von entfernten Datenbanken erforderlich)	<p>Für die Verwaltung einer entfernten SQL Anywhere-Datenbank müssen sich die folgenden Dateien auf dem entfernten Gerät befinden:</p> <ul style="list-style-type: none"><li>• <i>dbcon16.dll</i></li><li>• <i>dbeng16.exe</i><sup>5</sup></li><li>• <i>dbeng16.lic</i><sup>5</sup></li><li>• <i>dbghelp.dll</i></li><li>• <i>dbicu16.dll</i></li><li>• <i>dbicudt16.dll</i><sup>5</sup></li><li>• <i>dbicudt16.dat</i><sup>4</sup></li><li>• <i>dblggen16.dll</i><sup>1</sup></li><li>• <i>dblib16.dll</i></li><li>• <i>dbmlsync.exe</i></li><li>• <i>dbmlsynccli16.dll</i></li><li>• <i>dbscript16.dll</i></li><li>• <i>dbsrv16.exe</i></li><li>• <i>dbsrv16.lic</i></li><li>• <i>dbtool16.dll</i></li><li>• <i>mlagent.exe</i></li><li>• <i>mlastop.exe</i></li><li>• <i>mlasaadapter16.dll</i></li></ul> <p>Bestimmte Funktionen von SQL Anywhere erfordern möglicherweise das Deployment zusätzlicher Dateien. Siehe „<a href="#">Deployment von Datenbanken und Anwendungen</a>“ [<i>SQL Anywhere Server - Programmierung</i>].</p>

<sup>1</sup> Verwenden Sie für Deutsch, Japanisch und Chinesisch *dblgde16.dll*, *dblgja16.dll* bzw. *dblgzh16.dll*.

<sup>2</sup> Unter Windows Mobile nicht erforderlich, außer Sie verwenden die dbtools-Schnittstelle.

<sup>3</sup> Nicht erforderlich, wenn die Datenbank mit dbinit -zn UTF8BIN initialisiert wird. Siehe „[Dienstprogramm Initialisierung \(dbinit\)](#)“ [*SQL Anywhere Server - Datenbankadministration*].

<sup>4</sup> Nur für Windows Mobile.

<sup>5</sup> Für Windows-Betriebssysteme, mit Ausnahme von Windows Mobile.

**Unix-Anwendungen unter Unix, Linux und Mac OS X**

Alle Verzeichnisse sind relativ zum Verzeichnis *\$SQLANY16*.

Beschreibung	Unix-Dateien
MobiLink-Synchronisationsclient	<ul style="list-style-type: none"> <li>• <i>bin32/dbmlsync</i></li> <li>• <i>res/dblgen16.res</i></li> <li>• <i>lib32/libdbicu16_r.so</i><sup>1</sup></li> <li>• <i>lib32/libdblib16_r.so</i><sup>1</sup></li> <li>• <i>lib32/libdbtool16_r.so</i><sup>1</sup></li> </ul>
Dbmlsync-C++-API oder die SQL-Anweisung SYNCHRONIZE	<ul style="list-style-type: none"> <li>• MobiLink-Synchronisationsclient-Dateien</li> <li>• <i>lib32/libdbmlsynccli16_r.so</i></li> </ul>
Sicherheitsoption <sup>2</sup>	<ul style="list-style-type: none"> <li>• <i>lib32/libmlcrsa16_r.so</i><sup>1</sup></li> </ul>

<sup>1</sup> Für Linux lautet die Dateierweiterung *.so*. Für Mac OS X lautet die Dateierweiterung *.dylib*.

<sup>2</sup> Die Transportschichtssicherheit setzt voraus, dass Sie die getrennt lizenzierte SQL Anywhere-Sicherheitsoption erwerben. Außerdem ist sie Exportbeschränkungen unterworfen. Hinweise zur Bestellung dieser Komponente finden Sie unter „Getrennt lizenzierte Komponenten“ [[SQL Anywhere 16 - Einführung](#)].

## Deployment des MobiLink-Clients für UltraLite

Für UltraLite-Clients enthält die UltraLite-Laufzeitbibliothek oder die UltraLite-Komponente die erforderlichen Funktionen für den Synchronisationsdatenstrom. Die UltraLite-Laufzeitbibliothek wird in Ihre Anwendung kompiliert. Das Deployment unterliegt Ihrer Lizenzvereinbarung.

Beschreibung	Windows-Dateien
MobiLink-Agent	<p>Für die Verwaltung einer entfernten UltraLite-Datenbank müssen sich die folgenden Dateien auf dem entfernten Gerät befinden:</p> <ul style="list-style-type: none"> <li>• <i>dblgen16.dll</i><sup>1</sup></li> <li>• <i>mlagent.exe</i></li> <li>• <i>mlastop.exe</i></li> <li>• <i>mlauladapt16.dll</i></li> <li>• <i>uleng16.exe</i></li> </ul>

<sup>1</sup> Verwenden Sie für Deutsch, Japanisch und Chinesisch *dblge16.dll*, *dblga16.dll* bzw. *dblgh16.dll*.

### Siehe auch

- „UltraLite-Deployment“ [[UltraLite - Datenbankverwaltung](#)]
- C/C++: „So erstellen und Sie UltraLite C++-Anwendungen und führen ein Deployment durch.“ [[UltraLite - C- und C++-Programmierung](#)]
- .NET: „Lektion 5: Erstellung und Deployment der Anwendung“ [[UltraLite - .NET-Programmierung](#)]



---

# Index

## Symbole

- a, Option
  - MobiLink-Serveroption (mlsrv16),51
- ap, Option
  - MobiLink Replay, Dienstprogramm (mlreplay),702
- b, Option
  - MobiLink-Serveroption (mlsrv16),51
- bn, Option
  - MobiLink-Serveroption (mlsrv16),52
- c, Option
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - MobiLink-Profiler (mlprof),194
  - MobiLink-Serveroption (mlsrv16),53
- ca, Option
  - MobiLink-Serveroption (mlsrv16),53
- cinit, Option
  - MobiLink-Serveroption (mlsrv16),54
- classic, Option
  - MobiLink-Server (mlsrv16), Option -sl java,79
- classpath, Option
  - MobiLink-Server (mlsrv16), Option -sl java,79
- clrConGC, Option
  - MobiLink-Server (mlsrv16), Option -sl dnet,78
- clrFlavor, Option
  - MobiLink-Server (mlsrv16), Option -sl dnet,78
- clrVersion, Option
  - MobiLink-Server (mlsrv16), Option -sl dnet,78
- cm, Option
  - MobiLink-Serveroption (mlsrv16),55
- cmax, Option
  - MobiLink-Serveroption (mlsrv16),55
- cmin, Option
  - MobiLink-Serveroption (mlsrv16),56
- cn, Option
  - MobiLink-Serveroption (mlsrv16),56
- cp, Option
  - MobiLink-Server (mlsrv16), Option -sl java,79
- cr, Option
  - MobiLink-Serveroption (mlsrv16),57
- cs, Option
  - MobiLink-Serveroption (mlsrv16),57
- d, Option
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - MobiLink-generiertes Replay-API-Dienstprogramm (mlgenreplayapi),710
  - MobiLink-Server (mlsrv16), Option -sl java,79
- dl, Option
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - MobiLink-Serveroption (mlsrv16),58
- DMLStartClasses, Option
  - Java, benutzerdefinierte Startklassen,471
  - MobiLink-Server (mlsrv16)-Option -sl java,79
- dr, Option
  - MobiLink-Serveroption (mlsrv16),59
- ds, Option
  - MobiLink-Serveroption (mlsrv16),59
- dsd, Option
  - MobiLink-Serveroption (mlsrv16),60
- dt, Option
  - MobiLink-Serveroption (mlsrv16),60
- e, Option
  - MobiLink-Serveroption (mlsrv16),61
- esu, Option
  - MobiLink-Serveroption (mlsrv16),62
- et, Option
  - MobiLink-Serveroption (mlsrv16),62
- f, Option
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - MobiLink Replay, Dienstprogramm (mlreplay),702
- ftr, Option
  - MobiLink-Serveroption (mlsrv16),63
- ftru, Option
  - MobiLink-Serveroption (mlsrv16),64
- h, Option
  - MobiLink, Stopp-Dienstprogramm (mlstop),698
- hotspot, Option
  - MobiLink-Server (mlsrv16), Option -sl java,79
- jrepath, Option
  - MobiLink-Server (mlsrv16), Option -sl java,79
- ldt, Option
  - MobiLink Replay, Dienstprogramm (mlreplay),702
- ls, Option
  - MobiLink Replay, Dienstprogramm (mlreplay),702
- lsc, Option
  - MobiLink-Serveroption (mlsrv16),65
- MLAutoLoadPath, Option
  - Info,548

- MobiLink-Server (mlsrv16), Option -sl dnet,78
- MLDomConfigFile, Option
  - Info,548
  - MobiLink-Server (mlsrv16), Option -sl dnet,78
- MLStartClasses
  - .NET, benutzerdefinierte Startklassen,542
  - MobiLink-Server (mlsrv16), Option -sl dnet,78
- mt, Option
  - MobiLink Replay, Dienstprogramm (mlreplay),702
- n, Option
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - MobiLink Replay, Dienstprogramm (mlreplay),702
- nc, Option
  - MobiLink-Serveroption (mlsrv16),65
- notifier, Option
  - MobiLink-Serveroption (mlsrv16),66
- o, Option
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - MobiLink Replay, Dienstprogramm (mlreplay),702
  - MobiLink-generiertes Replay-API-Dienstprogramm (mlgenreplayapi),710
  - MobiLink-Serveroption (mlsrv16),67
- on, Option
  - MobiLink-Serveroption (mlsrv16),68
- oq, Option
  - MobiLink-Serveroption (mlsrv16),68
- os, Option
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - MobiLink Replay, Dienstprogramm (mlreplay),702
  - MobiLink-Serveroption (mlsrv16),68
- ot, Option
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - MobiLink Replay, Dienstprogramm (mlreplay),702
  - MobiLink-generiertes Replay-API-Dienstprogramm (mlgenreplayapi),710
  - MobiLink-Serveroption (mlsrv16),69
- p, Option
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - MobiLink Replay, Dienstprogramm (mlreplay),702
  - MobiLink-Profiler (mlprof),194
- pc, Option
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
- ping, Option
  - MobiLink Replay, Dienstprogramm (mlreplay),702
- ppv, Option
  - MobiLink-Serveroption (mlsrv16),70
- q, Option
  - MobiLink, Stopp-Dienstprogramm (mlstop),698
  - MobiLink-Serveroption (mlsrv16),75
- r, Option
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - MobiLink Replay, Dienstprogramm (mlreplay),702
  - MobiLink-Profiler (mlprof),194
  - MobiLink-Serveroption (mlsrv16),75
- rd, Option
  - MobiLink-Serveroption (mlsrv16),76
- rep, Option
  - MobiLink Replay, Dienstprogramm (mlreplay),702
- rg, Option
  - MobiLink Replay, Dienstprogramm (mlreplay),702
- rp, Option
  - MobiLink Replay, Dienstprogramm (mlreplay),702
  - MobiLink-Serveroption (mlsrv16),77
- rrp, Option
  - MobiLink-Serveroption (mlsrv16),76
- s, Option
  - MobiLink-Serveroption (mlsrv16),77
- sci, Option
  - MobiLink Replay, Dienstprogramm (mlreplay),702
- server, Option
  - MobiLink-Server (mlsrv16), Option -sl java,79
- sl dnet, Option
  - MLAutoLoadPath verwenden,539
  - MLDomConfigFile verwenden,548
  - benutzerdefinierte Startklassen,542
  - MobiLink-Serveroption (mlsrv16),78
- sl java, Option
  - benutzerdefinierte Startklassen,471
  - MobiLink-Serveroption (mlsrv16),79
- sm, Option
  - MobiLink-Serveroption (mlsrv16),81
- sv, Option
  - MobiLink Replay. Dienstprogramm (mlreplay),702
- t, Option
  - MobiLink, Stopp-Dienstprogramm (mlstop),698
- tc, Option
  - MobiLink-Serveroption (mlsrv16),82
- tf, Option
  - MobiLink-Serveroption (mlsrv16),83

- 
- ts, Option
    - MobiLink-Serveroption (mlsrv16),84
  - tx, Option
    - MobiLink-Serveroption (mlsrv16),85
  - u, Option
    - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
    - MobiLink Replay, Dienstprogramm (mlreplay),702
    - MobiLink-Profiler (mlprof),194
  - ud, Option
    - MobiLink-Serveroption (mlsrv16),85
  - ui, Option
    - MobiLink-Serveroption (mlsrv16),86
  - urc, Option
    - MobiLink-Performance-Vorteile,148
  - ux, Option
    - MobiLink-Serveroption (mlsrv16),86
  - v+, Option
    - MobiLink-Serveroption (mlsrv16),86
  - v, Option
    - MobiLink-Serveroption (mlsrv16),86
    - Performance des MobiLink SQL Anywhere Client-Dienstprogramms (dbmlsync),147
  - vc, Option
    - MobiLink-Serveroption (mlsrv16),86
  - ve, Option
    - MobiLink-Serveroption (mlsrv16),86
  - verbose, Option
    - MobiLink-Server (mlsrv16), Option, -sl java,79
  - vf, Option
    - MobiLink-Serveroption (mlsrv16),86
  - vh, Option
    - MobiLink-Serveroption (mlsrv16),86
  - vm, Option
    - MobiLink-Serveroption (mlsrv16),86
  - vn, Option
    - MobiLink-Serveroption (mlsrv16),86
  - vp, Option
    - MobiLink-Serveroption (mlsrv16),86
  - vr, Option
    - MobiLink-Serveroption (mlsrv16),86
  - vs, Option
    - MobiLink-Serveroption (mlsrv16),86
  - vt, Option
    - MobiLink-Serveroption (mlsrv16),86
  - vu, Option
    - MobiLink-Serveroption (mlsrv16),86
  - w, Option
    - MobiLink, Stopp-Dienstprogramm (mlstop),698
    - MobiLink-Serveroption (mlsrv16),90
  - wm, Option
    - MobiLink-Serveroption (mlsrv16),91
  - wn, mlsrv16-Option
    - Performance,144
  - wn, Option
    - MobiLink-Serveroption (mlsrv16),82
  - wu, Option
    - MobiLink-Serveroption (mlsrv16),92
  - x, Option
    - MobiLink Replay, Dienstprogramm (mlreplay),702
    - MobiLink-Profiler (mlprof),194
    - MobiLink-Server (mlsrv16), Option -sl java,79
    - MobiLink-Serveroption (mlsrv16),93
  - zf, Option
    - MobiLink-Serveroption (mlsrv16),102
  - zp, Option
    - MobiLink-Serveroption (mlsrv16),103
  - zs, Option
    - gemeinsamer Serverstatus,103
    - MobiLink-Serveroption (mlsrv16),103
  - zt, Option
    - MobiLink-Serveroption (mlsrv16),104
  - zus, Option
    - MobiLink-Serveroption (mlsrv16),105
  - zw, Option
    - MobiLink-Serveroption (mlsrv16),105
  - zwd, Option
    - MobiLink-Serveroption (mlsrv16),106
  - zwe, Option
    - MobiLink-Serveroption (mlsrv16),106
  - .NET
    - MobiLink, objektbasierter Datenfluss,627
    - MobiLink-Datentypen,541
    - MobiLink-Synchronisationsskripten,536
  - .NET CLR
    - MobiLink-Optionen,78
  - .NET MobiLink-Server-API (*Siehe* MobiLink-Server, API für .NET)
  - .NET, Synchronisationsbeispiel
    - MobiLink, .NET-Synchronisationslogik,550
  - .NET, Synchronisationsmethoden
    - Info,548
  - .NET-Klassen
    - Instanziierung für .NET-Synchronisationslogik,540
  - .NET-Synchronisationslogik
    - .NET-Klassen-Instanziierungen,540

- Beispiel,550
- Deployment unter 32-Bit-Unix,778
- Deployment unter 32-Bit-Windows,772
- Deployment unter 64-Bit-Unix,780
- Deployment unter 64-Bit-Windows,775
- Fehlersuche,546
- Methoden,542
- MobiLink-Performance,147
- Setup,537
- unterstützte Sprachen,536
- @data-Option
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - MobiLink Replay, Dienstprogramm (mlreplay),702
  - MobiLink, Stopp-Dienstprogramm (mlstop),698
  - MobiLink-Serveroption (mlsrv16),50
- @EmployeeID-Variable
  - mit MobiLink-Primärschlüsselpools verwenden,123
- A**
- a.
  - MobiLink, Präfix benannter Parameter,237
  - MobiLink, Präfix benutzerdefinierter Parameter,253
- Abbrechen
  - MobiLink-Server,32
- Abhängigkeiten von Diensten
  - MobiLink,39
- Abrufintervall
  - MobiLink-Agent,166
- active-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistiken,209
- Adaptive Server Enterprise
  - begin\_connection\_autocommit, Ereignis,301
  - DDL in MobiLink verwenden,301
  - konsolidierte MobiLink-Datenbank,10
  - MobiLink-Datenzuordnung,714
  - MobiLink-Isolationsstufen,140
  - MobiLink-Synchronisation,11
  - StaticCursorLongColBuffLen,11
- Add-Methode
  - DBParameterCollection-Klasse [MobiLink-Server-API für .NET],577
- addErrorListening-Methode
  - ServerContext-Schnittstelle [MobiLink-Server-Java-API],507
- addInfoListener-Methode
  - ServerContext-Schnittstelle [MobiLink-Server-Java-API],507
- addShutdownListener-Methode
  - ServerContext-Schnittstelle [MobiLink-Server-Java-API],508
- addWarningListener-Methode
  - ServerContext-Schnittstelle [MobiLink-Server-Java-API],509
- Administrations-Abrufintervall
  - MobiLink-Agent,166
- AdventureWorks
  - Synchronisationsprobleme,16
- Agenten-DB-Datei
  - entfernte Aufgaben,187
- Agenten-ID
  - entfernte Aufgaben,187
  - Info,155
- Agentendatenbank
  - Definition,156
- Agentenlog-Datei
  - entfernte Aufgaben,187
- Akkustand
  - entfernte Aufgaben,187
- Aktivieren, verteilte Microsoft-Transaktionen
  - Oracle-Treiberoption,768
- Alle Wege, Synchronisation (*Siehe* Zwei-Wege-Synchronisation)
- Alle Zeilen in einer Tabelle löschen
  - MobiLink,269
- Alte Zeilenparameter
  - MobiLink-Skripten,237
- Ändern, letzte Downloadzeit
  - MobiLink,110
- Anführungszeichen
  - konsolidierte MobiLink-Datenbanken in IBM DB2,14
- Angepasste Validierung
  - MobiLink, dateibasierte Downloads,221
- Anpassen von Statistiken
  - MobiLink-Profiler,205
- Antialiasing
  - MobiLink-Profiler, Option,201
- Anweisungsbasierte Skripten
  - Zeilen auslesen,262
- Anweisungsbasierte Uploads
  - Konflikterkennung,125
- Anwendungen

---

- Deployment von MobiLink,771
- Anwendungsserver
  - mit MobiLink synchronisieren,627
- Anwendungsübergreifendes Messaging
  - (*Siehe auch* Messaging)
- Anywhere verwalten
  - MobiLink, dateibasierte Downloads,213
- APIs
  - MobiLink-Server, API für Java,477
- Arraygröße
  - Oracle-Treiberoption,768
- ASE
  - (*Siehe auch* Adaptive Server Enterprise)
- Assembly
  - in MobiLink .NET-Synchronisationslogik suchen,537
  - in MobiLink implementieren,548
- Assistent zum Erstellen eines Dienstes
  - MobiLink,36
- Assistent zum Erstellen von Verbindungsskripten verwenden,259
- Assistent zur Skriptversionserstellung verwenden,257
- Aufgaben
  - (*Siehe auch* entfernte Aufgaben)
- Aufrechterhalten
  - Eindeutige Primärschlüssel,117
- Aufrechterhalten, eindeutige Primärschlüssel
  - UUIDs,118
  - zusammengesetzte Schlüssel,118
- Ausführen
  - MobiLink-Server,30
- Ausführen, .NET-Synchronisationslogik
  - Info,537
- Ausführen, Java-Synchronisationslogik
  - Info,463
- Ausführen, MobiLink
  - als Daemon,35
  - außerhalb der aktuellen Sitzung,34
  - Info,30
- Ausführen, MobiLink-Server
  - als Dienst,35
  - in einer Serverfarm,41
  - Info,30
- Ausführlichkeit
  - MobiLink-Performance,147
- Ausführlichkeitsstufe
  - MobiLink-Serveroption (mlsrv16) -v,86

- Ausgeben, Informationen von .NET
  - MobiLink, .NET-Synchronisationslogik,544
- Auslastungsdiagramm, Fensterausschnitt
  - MobiLink-Profiler,200
- Auswahl, Standardwerte
  - MobiLink, global autoincrement,121
- Auswahlrechteck
  - MobiLink-Profiler, Übersicht, Fensterausschnitt,203
- authenticate\_file\_transfer
  - Verbindungsereignis,284
- authenticate\_file\_upload
  - Verbindungsereignis,285
- authenticate\_parameters
  - Verbindungsereignis,287
- authenticate\_user
  - Verbindungsereignis,291
- authenticate\_user-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- authenticate\_user\_hashed
  - Verbindungsereignis,296
- authentication\_status, Synchronisationsparameter
  - Info,291
- Authentifizierung
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
- Authentifizierungsparameter
  - MobiLink,253
  - MobiLink-Skripten,237
- AUTOINCREMENT
  - (*Siehe auch* GLOBAL AUTOINCREMENT)
- Automatische Validierung
  - MobiLink, dateibasierte Downloads,219
- AvantGo (*Siehe* UltraLite für M-Business Anywhere)

## B

- Beenden
  - MobiLink-Server,32
- Befehle
  - entfernten Aufgaben hinzufügen,186
  - MobiLink, entfernte Aufgaben,181
- Befehlszeile
  - mlsrv16 starten,44
  - MobiLink-Server (mlsrv16),44
- Befehlszeilen-Dienstprogramme
  - MobiLink Replay (mlreplay), Syntax ,702

- MobiLink-Benutzerauthentifizierung (mluser),  
Syntax,699
- MobiLink-generierte Replay-API  
(mlgenreplayapi), Syntax,710
- Syntax des MobiLink-Stopp-Dienstprogramms  
(mlstop),698
- Befehlszeilenprogramme
  - MobiLink-Synchronisation,697
- begin\_connection
  - Verbindungsereignis,300
- begin\_connection\_autocommit
  - Verbindungsereignis,301
- begin\_download
  - Tabellenereignis,304
  - Verbindungsereignis,302
- begin\_download\_deletes
  - Tabellenereignis,306
- begin\_download\_rows
  - Tabellenereignis,308
- begin\_publication
  - Verbindungsereignis,310
- begin\_sync-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- begin\_synchronization
  - Tabellenereignis,316
  - Verbindungsereignis,313
- begin\_upload
  - Tabellenereignis,321
  - Verbindungsereignis,319
- begin\_upload\_deletes
  - Tabellenereignis,324
- begin\_upload\_rows
  - Tabellenereignis,326
- Behandeln mehrerer Fehler in einer einzelnen SQL-  
Anweisung
  - MobiLink,270
- Behandeln, MobiLink-Serverfehler in .NET
  - MobiLink, Java-Synchronisationslogik,544
- Behandeln, MobiLink-Serverfehler in Java
  - MobiLink, Java-Synchronisationslogik,470
- Beispiel-Domänen-Konfigurationsdatei
  - MobiLink,549
- Beispiele
  - .NET-Synchronisationslogik,550
  - Java-Synchronisationslogik,473
- Beispieleigenschaften
  - MobiLink-Profiler,204
- Benannte Parameter
  - Info über MobiLink,236
  - last\_download,109
  - last\_table\_download,109
  - Synchronisation von räumlichen Daten,7
- Benannte Skriptparameter
  - Info über MobiLink,236
  - ml\_add\_column-Systemprozedur,650
- Benannte Zeilenparameter
  - Info über MobiLink-Skripten,237
  - Spalteninformationen zur konsolidierten  
Datenbank hinzufügen,650
- Benutzerauthentifizierung, Dienstprogramm (mluser)  
Syntax,699
- Benutzerdefinierte Parameter
  - MobiLink,252
- Benutzerdefinierte Prozeduren
  - konsolidierte MobiLink-Datenbanken in IBM  
DB2,14
- Benutzerdefinierte Startklassen
  - MobiLink, .NET-Synchronisationslogik,542
  - MobiLink, Java-Synchronisationslogik,471
- Benutzernamen
  - MobiLink-Benutzerauthentifizierung,  
Dienstprogramm (mluser),699
- Benutzerparameter
  - MobiLink,252
- Bereitgestellte Aufgabe (*Siehe* bereitgestellte entfernte  
Aufgabe)
- Bereitgestellte entfernte Aufgabe
  - abbrechen,176
  - Definition,155
  - einleiten,177
  - Empfänger hinzufügen,179
  - reaktivieren,178
  - verwenden,176
- Berichten, Fehler
  - MobiLink-Synchronisation,271
- Bezeichner
  - maximale Länge in IBM DB2 LUW,5
- BLOBs
  - von ASE herunterladen,11
- Blockierende Downloadbestätigung
  - Info,137
- Broadcast-Download
  - MobiLink, dateibasierte Downloads,213
- buffer\_size, Option
  - MobiLink-Server (mlsrv16), Option -x für OE,99
- buffer\_size, Protokolloption

---

MobiLink-Server (mlsrv16), Option -x für HTTP,96  
MobiLink-Server (mlsrv16), Option -x für HTTPS,98

## C

C#, Programmiersprache  
    MobiLink-Optionen,78  
    MobiLink-Synchronisationsskripten,536  
    Unterstützung für MobiLink .NET,536  
C++, Programmiersprache  
    Unterstützung für MobiLink .NET,536  
CHAR, Datentyp  
    MobiLink und andere DBMS-Systeme,6  
CHAR, Spalten  
    konsolidierte MobiLink-Datenbanken in ASE,11  
    konsolidierte MobiLink-Datenbanken in Oracle,21  
    konsolidierte MobiLink-Datenbanken in SQL Server,16  
    MobiLink-Fragen,6  
CHAR-Spalten  
    konsolidierte MobiLink-Datenbanken in IBM DB2,14  
    MobiLink-Serveroption (mlsrv16) -b,51  
CLASSPATH, Umgebungsvariable  
    MobiLink Java-Synchronisationslogik,463  
Clear-Methode  
    DBParameterCollection-Klasse [MobiLink-Server-API für .NET],577  
Client-Hookprozeduren  
    *(Siehe auch Ereignis-Hooks)*  
ClientCertificates-Eigenschaft  
    NetworkData-Schnittstelle [MobiLink-Server-API für .NET],600  
Close-Methode  
    DBCommand-Schnittstelle [MobiLink-Server-API für .NET],562  
    DBConnection-Schnittstelle [MobiLink-Server-API für .NET],564  
    DBRowReader-Schnittstelle [MobiLink-Server-API für .NET],585  
CLR  
    MobiLink-Optionen,78  
collect\_network\_data-Protokolloption  
    MobiLink-Server (mlsrv16), Option -x für HTTP,96  
MobiLink-Server (mlsrv16), Option -x für HTTPS,98  
MobiLink-Server (mlsrv16), Option -x für TCP/IP,94  
MobiLink-Server (mlsrv16), Option -x für TLS über TCP/IP,95  
ColumnNames-Eigenschaft  
    DBRowReader-Schnittstelle [MobiLink-Server-API für .NET],586  
ColumnTypes-Eigenschaft  
    DBRowReader-Schnittstelle [MobiLink-Server-API für .NET],586  
CommandText-Eigenschaft  
    DBCommand-Schnittstelle [MobiLink-Server-API für .NET],563  
Commit-Methode  
    DBConnection-Schnittstelle [MobiLink-Server-API für .NET],564  
Common Language Runtime  
    MobiLink-Optionen,78  
completed-Eigenschaft  
    MobiLink-Profiler, Synchronisationsstatistik,209  
config, Option  
    MobiLink-Server (mlsrv16), Option -x für OE,99  
conflicted\_updates-Eigenschaft  
    MobiLink-Profiler, Synchronisationsstatistik,209  
connection\_retries-Eigenschaft  
    MobiLink-Profiler, Synchronisationsstatistik,209  
Contains-Methode  
    DBParameterCollection-Klasse [MobiLink-Server-API für .NET],577  
CopyTo-Methode  
    DBParameterCollection-Klasse [MobiLink-Server-API für .NET],578  
Count-Eigenschaft  
    DBParameterCollection-Klasse [MobiLink-Server-API für .NET],582  
CreateAndInitMLReplayUploadTransaction-Callback-Funktion [mlreplaycallbacks.cpp]  
    Beschreibung,639  
CreateCommand-Methode  
    DBConnection-Schnittstelle [MobiLink-Server-API für .NET],565  
CreateSpatialValue-Methode  
    SpatialUtilities-Klasse [MobiLink-Server-API für .NET],611  
createSpatialValue-Methode

- SpatialUtilities-Klasse [MobiLink-Server-Java-API],517
- Cursor Skripten
  - Definition,235

## D

- Daemon
  - MobiLink betreiben als,35
- Dateibasierte Downloads
  - Beispiele,221
  - Info,213
- Dateidefinitionsdatenbank
  - erstellen,215
  - Info,215
- Dateien
  - MobiLink, dateibasierte Downloads,213
- Daten abgleichen (*Siehe* Synchronisation)
- Daten senden, aus selbstreferenzierenden Tabellen
  - Info,140
- Datenaustausch (*Siehe* Synchronisation)
- Datenbank-Worker-Threads
  - automatische Anpassung,145
  - MobiLink,149
  - MobiLink-Performance,144
- Datenbanken
  - konsolidierte MobiLink-Datenbanken,1
- Datenbankschemata
  - konsolidierte Tabellen in Beziehung zu entfernten MobiLink-Tabellen setzen,2
- Datenbanktools, Schnittstelle
  - (*Siehe auch* DBTools, Schnittstelle)
- Datenbankverbindungen
  - MobiLink-Performance,152
  - MobiLink-Performance, maximale Anzahl festlegen,146
- Datenfluss (MobiLink) (*Siehe* Direkte Zeilenbehandlung)
- Dateninkonsistenz
  - MobiLink, Konfliktbehandlung,124
- Datenkonsistenz
  - (*Siehe auch* Synchronisation)
- Datenquellenname
  - Oracle-Treiberoption,768
- Datenskripten
  - download\_cursor,329
  - download\_delete\_cursor,331
  - handle\_DownloadData,371

- handle\_UploadData,383
- Info,283
- upload\_delete,435
- upload\_fetch,437
- upload\_fetch\_column\_conflict,439
- upload\_insert,441
- upload\_new\_row\_insert,443
- upload\_old\_row\_insert,445
- upload\_update,458
- Datenstromparameter
  - (*Siehe auch* Protokolloptionen)
- Datentypen
  - MobiLink, .NET und SQL,541
  - MobiLink, IBM DB2 LUW ,723
  - MobiLink, Java und SQL,466
  - MobiLink, Microsoft SQL Server,733
  - MobiLink, MySQL,740
  - MobiLink, Oracle,746
  - MobiLink, Zuordnungen der konsolidierten Datenbank,714
  - MobiLink-Zuordnung für ASE,714
  - MobiLink-Zuordnung für SAP HANA,757
  - MobiLink-Zuordnung für SAP Sybase IQ,761
- Datentypzuordnung
  - MobiLink, konsolidierte Datenbanken,714
- Datenzuordnungen
  - Info,714
- DateTime-Eigenschaft
  - DateTimeWithTimeZone-Klasse [MobiLink-Server-API für .NET],558
- DateTimeWithTimeZone-Klasse [MobiLink-Server-API für .NET]
  - Beschreibung,553
  - DateTime-Eigenschaft,558
  - DateTimeWithTimeZone-Konstruktor,554
  - Day-Eigenschaft,558
  - Hour-Eigenschaft,558
  - Millisecond-Eigenschaft,559
  - Minute-Eigenschaft,559
  - Month-Eigenschaft,559
  - Parse-Methode,556
  - Second-Eigenschaft,559
  - TimeZoneHour-Eigenschaft,559
  - TimeZoneMinute-Eigenschaft,560
  - ToString-Methode,556
  - Year-Eigenschaft,560
- DateTimeWithTimeZone-Konstruktor

---

DateTimeWithTimeZone-Klasse [MobiLink-Server-API für .NET],554  
 Day-Eigenschaft  
     DateTimeWithTimeZone-Klasse [MobiLink-Server-API für .NET],558  
 DB-Speicherort  
     entfernte Aufgaben,187  
 DBCommand-Schnittstelle [MobiLink-Server-API für .NET]  
     Beschreibung,560  
     Close-Methode,562  
     CommandText-Eigenschaft,563  
     ExecuteNonQuery-Methode,562  
     ExecuteReader-Methode,562  
     Parameters-Eigenschaft,563  
     Prepare-Methode,563  
 DBConnection-Schnittstelle [MobiLink-Server-API für .NET]  
     Beschreibung,563  
     Close-Methode,564  
     Commit-Methode,564  
     CreateCommand-Methode,565  
     Rollback-Methode,565  
 DBConnectionContext  
     Konstruktoren,541  
 DBConnectionContext-Schnittstelle [MobiLink-Server-API für .NET]  
     Beschreibung,565  
     GetConnection-Methode,567  
     GetDownloadData-Methode,567  
     GetProperties-Methode,568  
     GetRemoteID-Methode,568  
     GetServerContext-Methode,569  
     GetVersion-Methode,570  
     NetworkData-Eigenschaft,570  
 DBConnectionContext-Schnittstelle [MobiLink-Server-Java-API]  
     Beschreibung,477  
     getConnection-Methode,479  
     getDownloadData-Methode,479  
     getNetworkData-Methode,480  
     getProperties-Methode,480  
     getRemoteID-Methode,481  
     getServerContext-Methode,481  
     getVersion-Methode,482  
 dblsn-Dienstprogramm  
     MobiLink-Client-Deployment unter Windows,783  
 dbmlsync (*Siehe* dbmlsync-Dienstprogramm)  
 dbmlsync, Dienstprogramm  
     Deployment,782  
     Deployment unter Unix,784  
     Deployment unter Windows,783  
 DBParameter-Klasse [MobiLink-Server-API für .NET]  
     Beschreibung,571  
     DbType-Eigenschaft,572  
     Direction-Eigenschaft,572  
     HasChanged-Feld,575  
     IsNullable-Eigenschaft,573  
     ParameterName-Eigenschaft,573  
     Precision-Eigenschaft,573  
     Scale-Eigenschaft,574  
     Size-Eigenschaft,574  
     Value-Eigenschaft,574  
 DBParameterCollection-Klasse [MobiLink-Server-API für .NET]  
     Add-Methode,577  
     Beschreibung,575  
     Clear-Methode,577  
     Contains-Methode,577  
     CopyTo-Methode,578  
     Count-Eigenschaft,582  
     DBParameterCollection-Konstruktor,576  
     GetEnumerator-Methode,579  
     IndexOf-Methode,579  
     Insert-Methode,580  
     IsFixedSize-Eigenschaft,582  
     IsReadOnly-Eigenschaft,582  
     IsSynchronized-Eigenschaft,582  
     Remove-Methode,580  
     RemoveAt-Methode,581  
     SyncRoot-Eigenschaft,583  
     this-Eigenschaft,583  
 DBParameterCollection-Konstruktor  
     DBParameterCollection-Klasse [MobiLink-Server-API für .NET],576  
 DBRowReader-Schnittstelle [MobiLink-Server-API für .NET]  
     Beschreibung,584  
     Close-Methode,585  
     ColumnNames-Eigenschaft,586  
     ColumnTypes-Eigenschaft,586  
     NextRow-Methode,585  
 DBTools-Schnittstelle  
     (*Siehe auch* Datenbanktools, Schnittstelle)  
 DbType-Eigenschaft

- DBParameter-Klasse [MobiLink-Server-API für .NET],572
- Debuggen
  - MobiLink-Verbindungen,44
- Debuggen, Java-Methoden
  - MobiLink, Java-Synchronisationslogik,468
- DelayCreationOfSimulatedClient-Callback-Funktion [mlreplaycallbacks.cpp]
  - Beschreibung,640
- DelayDestructionOfSimulatedClient-Callback-Funktion [mlreplaycallbacks.cpp]
  - Beschreibung,640
- DelayStartOfReplay-Callback-Funktion [mlreplaycallbacks.cpp]
  - Beschreibung,641
- Deployment (*Siehe* Deployment durchführen)
  - MobiLink, Anwendungen und Datenbanken,771
  - MobiLink-Anwendungen,771
  - MobiLink-Performance,143
  - MobiLink-Server,771
  - SQL Anywhere MobiLink-Clients,782
  - Überblick über MobiLink,771
  - UltraLite MobiLink-Clients,785
- Deployment entfernter Datenbanken
  - Info,771
- Deployment von MobiLink-Anwendungen
  - Info,771
- Deployment von SQL Anywhere MobiLink-Clients
  - Info,782
- Deployment von UltraLite MobiLink-Clients
  - Info,785
- Deployment, MobiLink-Server
  - Info,771
- Deployment, Überblick
  - MobiLink,771
- DestroyMLReplayUploadTransaction-Callback-Funktion [mlreplaycallbacks.cpp]
  - Beschreibung,641
- Detailtabelle, Fensterausschnitt
  - MobiLink-Profiler,198
- Diagramm, Fensterausschnitt
  - MobiLink-Profiler,200,201
- Dienste
  - in MobiLink hinzufügen,36
  - konfigurieren,36
  - löschen,36
  - MobiLink,36
  - MobiLink als Dienst betreiben,35
  - MobiLink, Abhängigkeiten,39
  - MobiLink, mehrere ausführen,39
  - MobiLink-Server,34
- Dienstprogramme
  - MobiLink,697
  - MobiLink-Benutzerauthentifizierung (mluser), Syntax,699
  - Syntax des MobiLink-Stopp-Dienstprogramms (mlstop),698
- Direction-Eigenschaft
  - DBParameter-Klasse [MobiLink-Server-API für .NET],572
- Direkt einfügen, Skripten
  - MobiLink,261
- Direkte Synchronisationsereignisse
  - Info,628
- Direkte Zeilenbehandlung
  - Downloads,637
  - Entwicklungstipps,630
  - handle\_DownloadData, Verbindungsereignis,371
  - handle\_UploadData, Verbindungsereignis,383
  - Info,627
  - Schnellstart,629
  - Uploads,631
- Direkter Zeilenzugriff (*Siehe* direkte Zeilenbehandlung)
- Domänen-Konfigurationsdateien
  - MobiLink,549
- Download von Daten
  - dateibasierte Downloads in MobiLink,213
- Download, Ergebnismenge aus einem Aufruf einer gespeicherten Prozedur
  - Synchronisationsmethoden,138
- Download, Zeilen
  - Synchronisationsskripten,265
- Download-Datei
  - für dateibasierte Downloads in MobiLink erstellen,216
- download-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- Download-Fehler
  - neu startbare MobiLink-Downloads,135
- Download-Zeitstempel
  - MobiLink, Generierung,110
  - MobiLink, Info,109
- download\_bytes-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- download\_cursor

---

- Aufruf einer gespeicherten Prozedur verwenden,138
- Beispiel mit Aufruf einer gespeicherten Prozedur,138
- Fremdschlüsseltabellen partitionieren,116
- Info,266
- MobiLink, unabhängiges Verteilen,114
- Performance,147
- Skripten in Downloadzeilen schreiben,265
- Tabellenereignis,329
- Verteilung mit Überlappungen,115
- zeitstempelbasierte Synchronisation,109
- download\_delete\_cursor
  - Aufruf einer gespeicherten Prozedur verwenden,138
  - Beispiel mit Aufruf einer gespeicherten Prozedur,138
  - Fremdschlüsseltabellen partitionieren,116
  - Info,267
  - Performance,147
  - Skripten in Downloadzeilen schreiben,265
  - Tabellenereignis,331
  - unabhängiges Verteilen,114
  - Verteilung mit Überlappungen,115
- download\_deleted\_rows-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- download\_errors-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- download\_fetched\_rows-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- download\_filtered\_rows-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- download\_statistics
  - Tabellenereignis,337
  - Verbindungsereignis,333
- download\_timestamp
  - MobiLink, Generierung,110
- download\_warnings-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- Downloadbestätigung
  - Info,137
- DownloadData-Schnittstelle [MobiLink-Server-API für .NET]
  - Beschreibung,586
  - GetDownloadTableByName-Methode,587
  - GetDownloadTables-Methode,588
- DownloadData-Schnittstelle [MobiLink-Server-Java-API]
  - Beschreibung,483
  - getDownloadTableByName-Methode,484
  - getDownloadTables-Methode,484
- Downloadereignisse
  - MobiLink-Synchronisation,281
- Downloads
  - dateibasiert, MobiLink,213
  - MobiLink, fehlgeschlagene Downloads,134
  - MobiLink-Performance,148
  - MobiLink-Skripten zum Download von Zeilen,265
  - MobiLink-Transaktion,274
  - zeitstempelbasiert,107
- DownloadTableData-Schnittstelle [MobiLink-Server-API für .NET]
  - Beschreibung,588
  - GetDeleteCommand-Methode,590
  - GetLastDownloadTime-Methode,591
  - GetName-Methode,591
  - GetSchemaTable-Methode,592
  - GetUpsertCommand-Methode,592
- DownloadTableData-Schnittstelle [MobiLink-Server-Java-API]
  - Beschreibung,485
  - getDeletePreparedStatement-Methode,488
  - getLastDownloadTime-Methode,489
  - getMetaData-Methode,490
  - getName-Methode,490
  - getUpsertPreparedStatement-Methode,491
- Downloadtransaktion
  - MobiLink,274
- duration-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- Dynamische Cachedimensionierung
  - MobiLink-Server,41

## E

- e2ee\_private\_key-Protokolloption
  - MobiLink-Server (mlsrv16), Option -x für HTTPS,98
  - MobiLink-Server (mlsrv16), Option -x für TCP/IP,95
- e2ee\_private\_key\_password- Protokolloption
  - MobiLink-Server (mlsrv16), Option -x für HTTPS,98
- e2ee\_private\_key\_password-Protokolloption
  - MobiLink-Server (mlsrv16), Option -x für TCP/IP,95

- Eindeutig
  - Primärschlüssel in MobiLink,117
- Eindeutige Primärschlüssel
  - global autoincrement für MobiLink,119
  - mit Schlüsselpools für MobiLink generieren,122
  - mit UUIDs für MobiLink generieren,118
  - mit zusammengesetzten Schlüsseln für MobiLink generieren,118
  - MobiLink,117
- Eindeutige Primärschlüssel aufrechterhalten
  - global autoincrement,119
  - Primärschlüsselpools,122
- Eindeutige Schlüssel
  - MobiLink,117
- Einfügen
  - Skripten in MobiLink,261
- Einrichten
  - konsolidierte MobiLink-Datenbanken,1
  - MobiLink, dateibasierte Downloads,214
  - MobiLink, Java-Synchronisationslogik,463
  - MobiLink, konsolidierte Datenbanken,3
- Einrichten einer konsolidierten Oracle-Datenbank
  - MobiLink,20
- Einrichten einer konsolidierten SAP HANA-Datenbank
  - MobiLink,25
- Einrichten einer konsolidierten SAP Sybase IQ-Datenbank
  - MobiLink,28
- Einrichten einer konsolidierten SQL Anywhere-Datenbank
  - MobiLink,28
- Einrichten einer konsolidierten Sybase ASE-Datenbank
  - MobiLink,10
- Einrichten, .NET-Synchronisationslogik
  - Info,537
- Einrichten, direkte Zeilenbehandlung
  - Info,629
- Einrichten, konsolidierte IBM DB2 LUW-Datenbank
  - MobiLink,12
- Einrichten, konsolidierte MySQL-Datenbank
  - MobiLink,17
- Einwege-Synchronisation
  - Info,116
- Empfänger
  - zu einer bereitgestellten entfernten Aufgabe hinzufügen,179
- end\_connection
  - Verbindungsereignis,340
- end\_download
  - Tabellenereignis,344
  - Verbindungsereignis,342
- end\_download\_deletes
  - Tabellenereignis,346
- end\_download\_rows
  - Tabellenereignis,348
- end\_publication
  - Verbindungsereignis,350
- end\_sync-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- end\_synchronization
  - Tabellenereignis,355
  - Verbindungsereignis,353
- end\_upload
  - Tabellenereignis,361
  - Verbindungsereignis,358
- end\_upload\_deletes
  - Tabellenereignis,363
- end\_upload\_rows
  - Tabellenereignis,366
- Engpässe
  - MobiLink-Performance,148
- Entfernte Aufgabe
  - Befehle,181
  - Befehle hinzufügen,186
  - Datei herunterladen, Befehl,182
  - Datei hochladen, Befehl,185
  - Datei kopieren, Befehl,181
  - Datei löschen, Befehl,182
  - Datei umbenennen, Befehl,184
  - Datenbank erstellen, Befehl,181
  - Datenbank löschen, Befehl,183
  - Definition,155
  - Eingabeaufforderung, Befehl,184
  - erstellen,173
  - exportieren,175
  - Info,169
  - Logik,172
  - Programm ausführen, Befehl,185
  - SQL ausführen, Befehl,183
  - Status,189
  - Synchronisieren, Befehl,185
  - Variablen,187
- Entfernte Aufgabe, Attribut
  - Auslösemechanismen,170

---

- Bedingungen,170
- Befehle,170
- Beschreibung,169
- hohe Priorität,171
- maximale Anzahl von Neuversuchen,170
- maximale Laufzeit,171
- Name,169
- Name des entfernten Schemas,170
- Schemaänderung,171
- Statusberichterstellung,171
- Zeitspanne zwischen Neuversuchen,170
- zufälliges Verzögerungsintervall,171
- Entfernte Aufgaben
  - bearbeiten,174
  - bereitstellen,174
- Entfernte Datenbanken
  - konsolidierte Tabellen in Beziehung zu entfernten MobiLink-Tabellen setzen,2
  - MobiLink-Zuordnung von Datentypen,714
- Entfernte Tabellen
  - Zeilen in MobiLink löschen,267
- Entfernte\_ID
  - entfernte Aufgaben,187
- Entwicklungstipps
  - MobiLink, direkte Zeilenbehandlung,630
  - MobiLink-Synchronisation,107
- equals-Methode
  - TimestampWithTimeZone-Klasse [MobiLink-Server-Java-API],522
- Ereignismodell
  - MobiLink-Pseudocode,276
- Ereignisse
  - Info über MobiLink-Ereignisse,232
  - Info zur MobiLink-Synchronisation,272
  - MobiLink,272
  - MobiLink, direkte Zeilenbehandlung,628
- Ereignisse beim Download
  - Info,281
  - Skripten für Zeilen-Download erstellen,265
- Ereignisse beim Upload
  - Info,279
  - Skripten für Zeilen-Upload erstellen,262
- Erforderliche Skripten
  - MobiLink,258
- Erkennen von Konflikten
  - MobiLink, mit upload\_fetch-Skripten,126
- Erkennen, Konflikte
  - MobiLink mit upload\_update-Skripten,128

- ERROR-Variable
  - LogMessage-Klasse [MobiLink-Server-Java-API],498
- ErrorListener-Ereignis
  - ServerContext-Schnittstelle [MobiLink-Server-API für .NET],607
- Erstellen
  - .NET-Synchronisationslogik,536
  - Dateidefinitionsdatenbank,215
  - Download-Datei für dateibasierte Downloads in MobiLink,216
  - Java-Synchronisationslogik,463
  - MobiLink, konsolidierte Datenbank,3
- Erstellen, dateibasierte Datenbanken
  - MobiLink,215
- Erstellen, Datenbanken
  - konsolidierte,3
- Erstellen, Download-Dateien
  - MobiLink, dateibasierter Download,216
- Erstellen, Java-Synchronisationsskript
  - MobiLink, Beispiel für Java-Synchronisationslogik,474
- Erstellen, konsolidierte Datenbanken
  - MobiLink, Info,3
- Excel
  - mit MobiLink synchronisieren,627
- ExecuteNonQuery-Methode
  - DBCommand-Schnittstelle [MobiLink-Server-API für .NET],562
- ExecuteReader-Methode
  - DBCommand-Schnittstelle [MobiLink-Server-API für .NET],562

## F

- Failover
  - MobiLink-Serverfarm,41
- Farm (*Siehe* MobiLink-Serverfarm)
- FatalException-Klasse [MobiLink-Server-API für .NET]
  - Beschreibung,593
  - FatalException-Konstruktor,593
- FatalException-Konstruktor
  - FatalException-Klasse [MobiLink-Server-API für .NET],593
- Fehler
  - aufzeichnen,271

- Behandlung während der MobiLink-Synchronisation,270
- MobiLink, modify\_error\_message-Verbindungsereignis,389
- Fehlerbehandlung
  - fehlgeschlagene Downloads handhaben,134
  - MobiLink, Ferndatenverlust,110
  - MobiLink-Server,375
  - MobiLink-Serverlog,33
  - MobiLink-Serverstart,43
  - neu startbare MobiLink-Downloads,135
  - während der MobiLink-Synchronisation,270
- Fehlerlogs
  - MobiLink-Serveroption (mlsrv16),61
- Fehlersuche
  - .NET synchronization logic,546
  - MobiLink-Serverlog,33
  - MobiLink-Synchronisation mit Java-Klassen,468
  - Synchronisationsfehler,61
- Fehlersuche, .NET-Synchronisationslogik Info,546
- Fehlgeschlagene Downloads
  - MobiLink,135
  - Synchronisationsmethoden,134
- Fehlgeschlagene Downloads wieder aufnehmen
  - MobiLink,135
- file\_authentication\_code
  - authenticate\_file\_transfer-Parameter,284
- FiniIdentifySimulatedClient-Callback-Funktion [mlreplaycallbacks.cpp]
  - Beschreibung,642
- FIPS, Option
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - MobiLink-Serveroption (mlsrv16),63
- FIPS, Protokolloption
  - MobiLink-Server (mlsrv16), Option -x für HTTPS,98
  - MobiLink-Server (mlsrv16), Option -x mit TCP/IP,95
- Fragezeichen
  - MobiLink, Skriptparameter,236
- Fragmentierung
  - (*Siehe auch* Partitionierung)
- Fremdschlüsseltabellen partitionieren
  - MobiLink,116
- FTP
  - MobiLink, datebasierte Downloads,213

## G

- Gehosteter Relay Server
  - (*Siehe auch* Sybase Hosted Relay Service)
- Gemeinsam genutzt
  - MobiLink-Serverfarm,41
- Gemeinsam genutzte Assemblys
  - in MobiLink implementieren,548
- Gemeinsamer Serverstatus
  - MobiLink-Server (mlsrv16), Option -zs,103
- Gemeinsames Nutzen von Regeln (*Siehe* Partitionierung)
- generate\_next\_last\_download\_timestamp
  - Verbindungsereignis,368
- Generierungsnummern
  - MobiLink, datebasierte Downloads,220
- Gespeicherte Prozeduren
  - für Datendownload verwenden,138
  - MobiLink,648
  - MobiLink, gespeicherte Prozeduren, Quellcode,261
  - zum Hinzufügen oder Löschen von Synchronisationsskripten verwenden,258
- GetBytes-Methode
  - SpatialUtilities-Klasse [MobiLink-Server-API für .NET],611
- getBytes-Methode
  - SpatialUtilities-Klasse [MobiLink-Server-Java-API],518
- getCertificateChain-Methode
  - NetworkData-Schnittstelle [MobiLink-Server-Java-API],503
- GetConnection-Methode
  - DBConnectionContext-Schnittstelle [MobiLink-Server-API für .NET],567
- getConnection-Methode
  - DBConnectionContext-Schnittstelle [MobiLink-Server-Java-API],479
- GetDeleteCommand-Methode
  - DownloadTableData-Schnittstelle [MobiLink-Server-API für .NET],590
- getDeletePreparedStatement-Methode
  - DownloadTableData-Schnittstelle [MobiLink-Server-Java-API],488
- GetDeletes-Methode
  - UploadedTableData-Schnittstelle [MobiLink-Server-API für .NET],619
- getDeletes-Methode

---

UploadedTableData-Schnittstelle [MobiLink-Server-Java-API],532	DownloadTableData-Schnittstelle [MobiLink-Server-API für .NET],591
GetDownloadApplyTime-Callback-Funktion [mlreplaycallbacks.cpp] Beschreibung,643	getLastDownloadTime-Methode
GetDownloadData-Methode	DownloadTableData-Schnittstelle [MobiLink-Server-Java-API],489
DBConnectionContext-Schnittstelle [MobiLink-Server-API für .NET],567	getMetaData-Methode
getDownloadData-Methode	DownloadTableData-Schnittstelle [MobiLink-Server-Java-API],490
DBConnectionContext-Schnittstelle [MobiLink-Server-Java-API],479	UploadedTableData-Schnittstelle [MobiLink-Server-Java-API],533
GetDownloadTableByName-Methode	GetMLReplayAPIVersion-Callback-Funktion [mlreplaycallbacks.cpp] Beschreibung,644
DownloadData-Schnittstelle [MobiLink-Server-API für .NET],587	GetName-Methode
getDownloadTableByName-Methode	DownloadTableData-Schnittstelle [MobiLink-Server-API für .NET],591
DownloadData-Schnittstelle [MobiLink-Server-Java-API],484	UploadedTableData-Schnittstelle [MobiLink-Server-API für .NET],621
GetDownloadTables-Methode	getName-Methode
DownloadData-Schnittstelle [MobiLink-Server-API für .NET],588	DownloadTableData-Schnittstelle [MobiLink-Server-Java-API],490
getDownloadTables-Methode	UploadedTableData-Schnittstelle [MobiLink-Server-Java-API],534
DownloadData-Schnittstelle [MobiLink-Server-Java-API],484	getNetworkData-Methode
GetEnumerator-Methode	DBConnectionContext-Schnittstelle [MobiLink-Server-Java-API],480
DBParameterCollection-Klasse [MobiLink-Server-API für .NET],579	GetProperties-Methode
getHTTPHeaders-Methode	DBConnectionContext-Schnittstelle [MobiLink-Server-API für .NET],568
NetworkData-Schnittstelle [MobiLink-Server-Java-API],503	getProperties-Methode
GetHTTPHeaderValue-Methode	DBConnectionContext-Schnittstelle [MobiLink-Server-Java-API],480
NetworkData-Schnittstelle [MobiLink-Server-API für .NET],599	ServerContext-Schnittstelle [MobiLink-Server-API für .NET],604
getHTTPHeaderValue-Methode	ServerContext-Schnittstelle [MobiLink-Server-Java-API],509
NetworkData-Schnittstelle [MobiLink-Server-Java-API],504	getPropertiesByVersion-Methode
GetHTTPHeaderValues-Methode	ServerContext-Schnittstelle [MobiLink-Server-API für .NET],605
NetworkData-Schnittstelle [MobiLink-Server-API für .NET],599	ServerContext-Schnittstelle [MobiLink-Server-Java-API],510
getHTTPHeaderValues-Methode	getPropertySetNames-Methode
NetworkData-Schnittstelle [MobiLink-Server-Java-API],504	ServerContext-Schnittstelle [MobiLink-Server-API für .NET],605
GetInserts-Methode	ServerContext-Schnittstelle [MobiLink-Server-Java-API],510
UploadedTableData-Schnittstelle [MobiLink-Server-API für .NET],620	GetRemoteID-Methode
getInserts-Methode	
UploadedTableData-Schnittstelle [MobiLink-Server-Java-API],532	
GetLastDownloadTime-Methode	

- DBConnectionContext-Schnittstelle [MobiLink-Server-API für .NET],568
- getRemoteID-Methode
  - DBConnectionContext-Schnittstelle [MobiLink-Server-Java-API],481
- GetSchemaTable-Methode
  - DownloadTableData-Schnittstelle [MobiLink-Server-API für .NET],592
  - UploadedTableData-Schnittstelle [MobiLink-Server-API für .NET],621
- GetServerContext-Methode
  - DBConnectionContext-Schnittstelle [MobiLink-Server-API für .NET],569
- getServerContext-Methode
  - DBConnectionContext-Schnittstelle [MobiLink-Server-Java-API],481
- GetSRID-Methode
  - SpatialUtilities-Klasse [MobiLink-Server-API für .NET],611
- getSRID-Methode
  - SpatialUtilities-Klasse [MobiLink-Server-Java-API],518
- GetStartClassInstances-Methode
  - ServerContext-Schnittstelle [MobiLink-Server-API für .NET],606
- getStartClassInstances-Methode
  - ServerContext-Schnittstelle [MobiLink-Server-Java-API],511
- getText-Methode
  - LogMessage-Klasse [MobiLink-Server-Java-API],497
- getTimeZoneOffsetHours-Methode
  - TimestampWithTimeZone-Klasse [MobiLink-Server-Java-API],524
- getTimeZoneOffsetMinutes-Methode
  - TimestampWithTimeZone-Klasse [MobiLink-Server-Java-API],524
- getType-Methode
  - LogMessage-Klasse [MobiLink-Server-Java-API],497
- GetUpdates-Methode
  - UploadedTableData-Schnittstelle [MobiLink-Server-API für .NET],622
- getUpdates-Methode
  - UploadedTableData-Schnittstelle [MobiLink-Server-Java-API],535
- GetUploadedTableByName-Methode
  - UploadData-Schnittstelle [MobiLink-Server-API für .NET],617
- getUploadedTableByName-Methode
  - UploadData-Schnittstelle [MobiLink-Server-Java-API],529
- GetUploadedTables-Methode
  - UploadData-Schnittstelle [MobiLink-Server-API für .NET],617
- getUploadedTables-Methode
  - UploadData-Schnittstelle [MobiLink-Server-Java-API],529
- GetUploadTransaction-Callback-Funktion [mlreplaycallbacks.cpp]
  - Beschreibung,644
- GetUpsertCommand-Methode
  - DownloadTableData-Schnittstelle [MobiLink-Server-API für .NET],592
- getUpsertPreparedStatement-Methode
  - DownloadTableData-Schnittstelle [MobiLink-Server-Java-API],491
- getUser-Methode
  - LogMessage-Klasse [MobiLink-Server-Java-API],498
- getValue-Methode
  - InOutInteger-Schnittstelle [MobiLink-Server-Java-API],493
  - InOutString-Schnittstelle [MobiLink-Server-Java-API],495
- GetVersion-Methode
  - DBConnectionContext-Schnittstelle [MobiLink-Server-API für .NET],570
- getVersion-Methode
  - DBConnectionContext-Schnittstelle [MobiLink-Server-Java-API],482
- Global
  - Skriptversionen in MobiLink,256
- GLOBAL AUTOINCREMENT
  - (*Siehe auch* AUTOINCREMENT)
- global autoincrement
  - Algorithmus,121
  - global\_database\_id für MobiLink festlegen,120
  - MobiLink, deklarieren,120
  - MobiLink, eindeutige Primärschlüssel,119
- global\_database\_id, Option
  - in MobiLink festlegen,120
- Globale Skriptversionen
  - MobiLink,256
- Globaler Assembly-Cache

---

- in MobiLink implementieren,548
- GlobalFini-Callback-Funktion
  - [mlreplaycallbacks.cpp]
  - Beschreibung,645
- GlobalInit-Callback-Funktion [mlreplaycallbacks.cpp]
  - Beschreibung,645
- Grundlegende Regeln
  - MobiLink,107
- Gruppen
  - MobiLink,167
- GUIDs
  - (*Siehe auch* UUIDs)

## H

- handle\_DownloadData
  - Verbindungsereignis,371
- handle\_error
  - Synchronisationsskripten,270
  - Verbindungsereignis,375
- handle\_odbc\_error
  - Verbindungsereignis,379
- handle\_UploadData
  - Verbindungsereignis,383
- HasChanged-Feld
  - DBParameter-Klasse [MobiLink-Server-API für .NET],575
- header\_limit-Protokolloption
  - MobiLink, mlsrv16-Option -x für HTTP,96
  - MobiLink, mlsrv16-Option -x für HTTPS,98
  - MobiLink, mlsrv16-Option -x für OE,99
- Herunterfahren
  - MobiLink, Stopp-Dienstprogramm (mlstop),698
  - MobiLink-Server,32
- Herunterladen von Löschungen
  - MobiLink, download\_delete\_cursor-Skripten,267
- Hinzufügen
  - Benutzernamen in MobiLink,699
  - MobiLink .NET-Tabellenskripten,654
  - MobiLink .NET-Verbindungsskripten,653
  - MobiLink Java-Tabellenskripten,656
  - MobiLink Java-Verbindungsskripten,655
  - MobiLink SQL-Tabellenskripten,668
  - MobiLink SQL-Verbindungsskripten,651
  - MobiLink-Eigenschaften,664
  - Synchronisationsskripten mit Sybase Central,259
- Hinzufügen und Löschen, Skripten
  - MobiLink,259

- Hinzufügen von Skripten
  - MobiLink, Info,258
- Hinzufügen von Skriptversionen
  - MobiLink,256
- Hinzufügen, Synchronisationsskripten
  - gespeicherte Prozeduren verwenden,258
- Hooks
  - (*Siehe auch* Ereignis-Hooks)
- host, Protokolloption
  - MobiLink-Server (mlsrv16), Option -x für HTTP,96
  - MobiLink-Server (mlsrv16), Option -x für HTTPS,98
  - MobiLink-Server (mlsrv16), Option -x für TCP/IP,94
  - MobiLink-Server (mlsrv16), Option -x für TLS über TCP/IP,95
- Hour-Eigenschaft
  - DateTimeWithTimeZone-Klasse [MobiLink-Server-API für .NET],558
- HTTP
  - MobiLink-Server (mlsrv16), Option -x,96
- HTTPHeaders-Eigenschaft
  - NetworkData-Schnittstelle [MobiLink-Server-API für .NET],600
- HTTPS
  - MobiLink-Server (mlsrv16), Option -x,97

## I

- ianywhere.ml.script-Paket
  - MobiLink-Server-API für Java, Referenz,477
- iAnywhere.MobiLink.Script-Namespace
  - MobiLink-Server-API für .NET, Referenz,552
- IBM BD2
  - MobiLink, Datenzuordnung für LUW,723
- IBM DB2
  - IBM DB2 LUW als konsolidierte MobiLink-Datenbank,12
  - MobiLink-Isolationsstufen,140
- IBM DB2 LUW
  - konsolidierte MobiLink-Datenbank,12
- IBM DB2 LUW, konsolidierte Datenbank
  - MobiLink,12
- IdentifySimulatedClient-Callback-Funktion [mlreplaycallbacks.cpp]
  - Beschreibung,646
- identity, Option

- MobiLink-Server (mlsrv16), Option -x für HTTPS,98
  - identity, Protokolloption
    - MobiLink-Server (mlsrv16), Option -x für HTTPS,98
  - identity-Option
    - MobiLink, mlsrv16-Option -x für TLS über TCP/IP,95
  - identity-Protokolloption
    - MobiLink, mlsrv16-Option -x für TLS über TCP/IP,95
  - identity\_password, Protokolloption
    - MobiLink-Server (mlsrv16), Option -x für HTTPS,98
  - identity\_password-Protokolloption
    - MobiLink, mlsrv16-Option -x für TLS über TCP/IP,95
  - ignore, Protokolloption
    - MobiLink-Server (mlsrv16), Option -x für TCP/IP,94
    - MobiLink-Server (mlsrv16), Option -x für TLS über TCP/IP,95
  - ignored\_deletes-Eigenschaft
    - MobiLink-Profiler, Synchronisationsstatistik,209
  - ignored\_inserts-Eigenschaft
    - MobiLink-Profiler, Synchronisationsstatistik,209
  - ignored\_updates-Eigenschaft
    - MobiLink-Profiler, Synchronisationsstatistik,209
  - IndexOf-Methode
    - DBParameterCollection-Klasse [MobiLink-Server-API für .NET],579
  - Indizes
    - MobiLink-Performance,147
  - INFO-Variable
    - LogMessage-Klasse [MobiLink-Server-Java-API],499
  - InfoListener-Ereignis
    - ServerContext-Schnittstelle [MobiLink-Server-API für .NET],607
  - Inkonsistenz
    - MobiLink, Konfliktbehandlung,124
  - Inkrementelle Uploads
    - MobiLink-Performance,145
  - InOutInteger-Schnittstelle [MobiLink-Server-Java-API]
    - Beschreibung,492
    - getValue-Methode,493
    - setValue-Methode,494
  - InOutString-Schnittstelle [MobiLink-Server-Java-API]
    - Beschreibung,494
    - getValue-Methode,495
    - setValue-Methode,495
  - Insert-Methode
    - DBParameterCollection-Klasse [MobiLink-Server-API für .NET],580
  - Integritätsregelfehler (*Siehe* Konflikte)
  - IsEndToEndEncrypted-Eigenschaft
    - NetworkData-Schnittstelle [MobiLink-Server-API für .NET],601
  - isEndToEndEncrypted-Methode
    - NetworkData-Schnittstelle [MobiLink-Server-Java-API],505
  - IsFixedSize-Eigenschaft
    - DBParameterCollection-Klasse [MobiLink-Server-API für .NET],582
  - IsHTTP-Eigenschaft
    - NetworkData-Schnittstelle [MobiLink-Server-API für .NET],601
  - isHTTP-Methode
    - NetworkData-Schnittstelle [MobiLink-Server-Java-API],505
  - IsNullable-Eigenschaft
    - DBParameter-Klasse [MobiLink-Server-API für .NET],573
  - Isolationsstufen
    - MobiLink,140
  - IsReadOnly-Eigenschaft
    - DBParameterCollection-Klasse [MobiLink-Server-API für .NET],582
  - IsSynchronized-Eigenschaft
    - DBParameterCollection-Klasse [MobiLink-Server-API für .NET],582
  - IsTLS-Eigenschaft
    - NetworkData-Schnittstelle [MobiLink-Server-API für .NET],601
  - isTLS-Methode
    - NetworkData-Schnittstelle [MobiLink-Server-Java-API],505
- ## J
- Java
    - MobiLink, objektbasierter Datenfluss,627
    - MobiLink-Datentypen,466
    - MobiLink-Server, API-Referenz,477

- 
- MobiLink-Synchronisationsskripten,463
  - Java MobiLink-Server-API (*Siehe* MobiLink-Server, API für Java)
  - Java VM
    - MobiLink-Optionen,79
  - Java vs. SQL, Synchronisationslogik
    - MobiLink-Performance,147
  - Java-Klassen
    - Instanziierung für Java-Synchronisationslogik,466
  - Java-Synchronisation
    - MobiLink, Java-Synchronisationslogik,473
  - Java-Synchronisationslogik
    - Beispiel,473
    - Deployment unter 32-Bit-Unix,778
    - Deployment unter 32-Bit-Windows,772
    - Deployment unter 64-Bit-Unix,780
    - Deployment unter 64-Bit-Windows,775
    - in der Befehlszeile von MobiLink-Server festlegen,463
    - Instanziierungen von Java-Klassen,466
    - Methoden,468
    - MobiLink-Performance,147
    - MobiLink-Server, API,477
    - Setup,463
  - Javadoc
    - MobiLink,477
  - K**
  - Keep Partial Download, Synchronisationsparameter
    - neu startbare Downloads,136
  - Kennwörter
    - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
    - verschlüsseln, im Oracle-Treiber,768
  - Klasseninstanzen
    - Java-Synchronisationslogik,466
    - MobiLink .NET-Synchronisationslogik,540
  - Kollationssequenzen
    - MobiLink-Synchronisation,765
  - Kollision
    - MobiLink-Konfliktlösung,124
  - Kommunikation
    - MobiLink-Server (mlsrv16), Option -c,53
    - MobiLink-Server (mlsrv16), Option -x,93
  - Komplettes Ereignismodell
    - MobiLink,272
    - MobiLink-Pseudocode,276
  - Konfigurieren
    - MobiLink, konsolidierte Datenbanken,3
  - Konfliktbehandlung
    - MobiLink,124
  - Konflikte
    - MobiLink,124
    - MobiLink, direkte Zeilenbehandlung,632
    - MobiLink, Erkennung,125
    - MobiLink-Performance,143
    - MobiLink-Performance, Erklärung,149
    - MobiLink-Standardverhalten,124
  - Konflikte lösen
    - resolve\_conflict-Skript,128
    - upload\_update-Skript,130
  - Konflikte, erkennen
    - MobiLink,125
  - Konflikte, lösen
    - MobiLink, Überblick,128
  - Konflikterkennung
    - MobiLink,125
    - MobiLink, anweisungsbasierte Uploads,125
  - Konfliktlösung
    - MobiLink,124
    - MobiLink, Erkennung,125
    - MobiLink-Konflikterkennung,125
    - MobiLink-Standardverhalten,124
  - Konfliktverarbeitung für direkte Uploads
    - MobiLink, direkte Zeilenbehandlung,632
  - Konsistenz
    - (*Siehe auch* Synchronisation)
  - Konsolidierte Datenbanken
    - ASE als MobiLink,10
    - Datenbanken, die keine SQL Anywhere-Datenbanken sind,5
    - DBMS-Abhängigkeiten,5
    - IBM DB2 LUW als MobiLink,12
    - MobiLink erstellen,3
    - MobiLink-Isolationsstufen,140
    - MobiLink-Zuordnung von Datentypen,714
    - MySQL als MobiLink,17
    - Oracle als MobiLink,20
    - SAP HANA als MobiLink,25
    - SAP Sybase IQ als MobiLink,28
    - SQL Anywhere als MobiLink,28
    - SQL Server als MobiLink,15
    - Synchronisationsskripten hinzufügen,258
    - Tabellen in Beziehung zu entfernten MobiLink-Tabellen setzen,2

- Konsolidierte Microsoft SQL Server-Datenbank
  - MobiLink,15
- Konsolidierte MobiLink-Datenbanken
  - Oracle,20
  - SQL Anywhere,28
  - SQL Server,15
- Konsolidierte Oracle-Datenbank
  - MobiLink,20
- Konsolidierte SAP Sybase IQ-Datenbank
  - MobiLink,28
- Konsolidierte SQL Anywhere-Datenbank
  - MobiLink,28
- Konstruktoren
  - MobiLink, .NET-Synchronisationslogik,541
  - MobiLink, Java-Synchronisationslogik,467
- Konvertierung
  - Zeichensatz durch ODBC-Treiber,767
- Konvertierung zwischen Zeichensätzen
  - MobiLink-Synchronisation,766
- L**
- Laden, Assemblies
  - MobiLink, .NET-Synchronisationslogik,548
- last\_download
  - MobiLink, benannte Parameter,109
  - modify\_last\_download\_timestamp, Verbindungsereignis,392
- last\_download\_timestamp
  - MobiLink, benannte Parameter,109
  - MobiLink, Generierung,110
- last\_table\_download
  - MobiLink, benannte Parameter,109
  - modify\_last\_download\_timestamp, Verbindungsereignis,392
- Lastverteilung
  - MobiLink-Serverfarm,41
- Läuft\_mit\_Netzstrom
  - entfernte Aufgaben,187
- Leere Zeichenfolgen
  - in Oracle nicht unterstützt,22
  - konsolidierte MobiLink-Datenbanken in Oracle,21
- Letzte geänderte Spalte
  - MobiLink,107
- Linux
  - Deployment von dbmlsync,784
- Listener-Dienstprogramm (dblsn)
  - MobiLink-Client-Deployment unter Windows,783
- log\_bad\_request Option
  - MobiLink-Server (mlsrv16), Option -x für HTTPS,98
  - MobiLink-Server (mlsrv16), Option -x für OE,99
- log\_bad\_request, Option
  - MobiLink-Server (mlsrv16), Option -x für HTTP,96
- LogCallback-Delegat [MobiLink-Server-API für .NET]
  - Beschreibung,623
- Logdatei-Anzeige
  - MobiLink-Serverlogs,34
- Logdateien
  - MobiLink-Server,33
  - MobiLink-Server anzeigen,34
- Logische Löschungen
  - download\_delete\_cursor-Skripten schreiben,267
- LogListener-Schnittstelle [MobiLink-Server-Java-API]
  - Beschreibung,495
  - messageLogged-Methode,496
- LogMessage-Klasse [MobiLink-Server-API für .NET]
  - Beschreibung,595
  - LogMessage-Konstruktor,595
  - MessageType-Enumeration,596
  - Text-Eigenschaft,596
  - Type-Eigenschaft,596
  - User-Eigenschaft,596
- LogMessage-Klasse [MobiLink-Server-Java-API]
  - Beschreibung,496
  - ERROR-Variable,498
  - getText-Methode,497
  - getType-Methode,497
  - getUser-Method,498
  - INFO-Variable,499
  - WARNING-Variable,500
- LogMessage-Konstruktor
  - LogMessage-Klasse [MobiLink-Server-API für .NET],595
- Logs
  - (*Siehe auch* Logdateien)
- LONG-Datentyp
  - Oracle-Synchronisation,753
- Löschen
  - MobiLink .NET-Tabellenskripten,654
  - MobiLink .NET-Verbindungsskripten,653
  - MobiLink Java-Tabellenskripten,656
  - MobiLink Java-Verbindungsskripten,655

- 
- MobiLink SQL-Tabellenskripten,668
  - MobiLink SQL-Verbindungsskripten,651
  - MobiLink-Eigenschaften,664
  - Zeilen in entfernten MobiLink-Datenbanken,267
  - Löschen von Skripten
    - MobiLink, Info,258
  - Löschen, Zeilen
    - entfernte MobiLink-Datenbanken,267
  - Löschungen
    - MobiLink-Downloads,267
    - Upload der Löschungen für SQL Anywhere-Clients,133
  - Löschungen, handhaben
    - MobiLink,132
  - Lösen
    - MobiLink-Konflikte,124
  - Lösen, Konflikte
    - MobiLink mit resolve\_conflict-Skripten,128
    - MobiLink mit upload\_update-Skripten,130
    - resolve\_conflict-Skript,128
    - upload\_update-Skript,130
  - Lösung
    - MobiLink-Konfliktlösung,124
  - LUW
    - IMB DB2 LUW als konsolidierte MobiLink-Datenbank,12
  - M**
  - Mac OS X
    - Deployment von dbmlsync,784
  - MakeConnection-Methode
    - ServerContext-Schnittstelle [MobiLink-Server-API für .NET],607
  - makeConnection-Methode
    - ServerContext-Schnittstelle [MobiLink-Server-Java-API],511
  - Mehrere Threads, Netzwerkaufgaben
    - MobiLink-Performance,144
    - MobiLink-Serveroption (mlsrv16),82
  - messageLogged-Methode
    - LogListener-Schnittstelle [MobiLink-Server-Java-API],496
  - MessageType-Enumeration
    - LogMessage-Klasse [MobiLink-Server-API für .NET],596
  - Methoden
    - MobiLink, .NET-Synchronisationslogik,542
    - MobiLink, Java-Synchronisationslogik,468
  - Methoden für automatisches autoincrement
    - konsolidierte MobiLink-Datenbanken in Oracle,21
  - Methoden registrieren
    - MobiLink-Server, API für Java,468
  - Microsoft ActiveSync
    - MobiLink-Client-Deployment unter Windows,783
  - Microsoft Distributed Transaction Coordinator
    - Oracle-Treiberoption,768
  - Microsoft Excel
    - mit MobiLink synchronisieren,627
  - Microsoft SQL Server
    - als konsolidierte MobiLink-Datenbank,15
    - MobiLink-Datenzuordnung,733
    - MobiLink-Isolationsstufen,140
  - Millisecond-Eigenschaft
    - DateTimeWithTimeZone-Klasse [MobiLink-Server-API für .NET],559
  - Minute-Eigenschaft
    - DateTimeWithTimeZone-Klasse [MobiLink-Server-API für .NET],559
  - ML-Benutzername
    - entfernte Aufgaben,187
  - ML-Datenstrom
    - entfernte Aufgaben,187
  - ML-Kennwort
    - entfernte Aufgaben,187
  - ml\_add\_certificates\_file-Systemprozedur
    - Syntax,650
  - ml\_add\_column-Systemprozedur
    - Syntax,650
  - ml\_add\_connection\_script-Systemprozedur
    - Syntax,651
  - ml\_add\_dnet\_connection\_script-Systemprozedur
    - Syntax,653
  - ml\_add\_dnet\_table\_script-Systemprozedur
    - Syntax,654
  - ml\_add\_java\_connection\_script-Systemprozedur
    - Syntax,655
  - ml\_add\_java\_table\_script-Systemprozedur
    - Syntax,656
  - ml\_add\_lang\_connection\_script-Systemprozedur
    - Syntax,657
  - ml\_add\_lang\_connection\_script\_chk-Systemprozedur
    - Syntax,658
  - ml\_add\_lang\_table\_script-Systemprozedur
    - Syntax,658
  - ml\_add\_lang\_table\_script\_chk-Systemprozedur

Syntax,658	Syntax,675
ml_add_lcs-Systemprozedur	ml_ra_add_agent_id-Systemprozedur
Syntax,657	Syntax,679
ml_add_lcs_chk-Systemprozedur	ml_ra_assign_task-Systemprozedur
Syntax,658	Syntax,679
ml_add_ldap_server-Systemprozedur	ml_ra_cancel_notification-Systemprozedur
Syntax,658	Syntax,680
ml_add_lts-Systemprozedur	ml_ra_cancel_task_instance-Systemprozedur
Syntax,658	Syntax,681
ml_add_lts_chk-Systemprozedur	ml_ra_clone_agent_properties-Systemprozedur
Syntax,658	Syntax,681
ml_add_missing_dnld_scripts-Systemprozedur	ml_ra_delete_agent_id-Systemprozedur
Syntax,660	Syntax,682
ml_add_passthrough-Systemprozedur	ml_ra_delete_events_before-Systemprozedur
Syntax,660	Syntax,682
ml_add_passthrough_repair-Systemprozedur	ml_ra_delete_remote_id-Systemprozedur
Syntax,661	Syntax,683
ml_add_passthrough_script-Systemprozedur	ml_ra_get_agent_events-Systemprozedur
Syntax,662	Syntax,684
ml_add_property-Systemprozedur	ml_ra_get_agent_ids-Systemprozedur
Syntax,664	Syntax,687
ml_add_table_script-Systemprozedur	ml_ra_get_agent_properties-Systemprozedur
Syntax,668	Syntax,687
ml_add_user-Systemprozedur	ml_ra_get_latest_event_id-Systemprozedur
Syntax,670	Syntax,688
ml_add_user_auth_policy-Systemprozedur	ml_ra_get_orphan_taskdbs-Systemprozedur
Syntax,670	Syntax,688
ml_delete_passthrough-Systemprozedur	ml_ra_get_remote_ids-Systemprozedur
Syntax,671	Syntax,689
ml_delete_passthrough_repair-Systemprozedur	ml_ra_get_task_results-Systemprozedur
Syntax,672	Syntax,690
ml_delete_passthrough_script-Systemprozedur	ml_ra_get_task_status-Systemprozedur
Syntax,672	Syntax,691
ml_delete_sync_state-Systemprozedur	ml_ra_manage_remote_db-Systemprozedur
Syntax,673	Syntax,693
ml_delete_sync_state_before-Systemprozedur	ml_ra_notify_agent_sync-Systemprozedur
Syntax,674	Syntax,694
ml_delete_task-Systemprozedur	ml_ra_notify_task-Systemprozedur
Syntax,683	Syntax,694
ml_delete_user-Systemprozedur	ml_ra_reassign_taskdb-Systemprozedur
Syntax,675	Syntax,694
ml_global, Skriptversion	ml_ra_set_agent_property-Systemprozedur
Info,256	Syntax,695
ml_model_check_all_schema-Systemprozedur	ml_ra_unmanage_remote_db-Systemprozedur
Syntax,676	Syntax,696
ml_model_check_version_schema-Systemprozedur	ml_reset_sync_state-Systemprozedur
Syntax,677	Syntax,696
ml_model_drop-Systemprozedur	ml_server_delete-Systemprozedur

- 
- Syntax,697
  - ml\_server\_update-Systemprozedur
    - Syntax,697
  - ml\_user
    - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - mlagent
    - Syntax,159
  - mlagent-Befehl
    - ausführen,158
    - Info,158
  - mlarbiter, Dienstprogramm
    - Unix,712
    - Windows,711
  - mlarbstop
    - Dienstprogramm zum MobiLink-Arbitr-Stopp,713
  - mlastop-Befehl
    - Info,163
  - mlDomConfig.xml
    - Info,549
  - mlgenreplayapi, Dienstprogramm
    - Optionen,710
    - Syntax,710
  - mlprof
    - Info zum MobiLink-Profiler,192
    - Optionen,194
    - starten,194
  - mlreplay-Dienstprogramm
    - Optionen,702
    - Syntax,702
  - mlreplaycallbacks.cpp-Referenz
    - CreateAndInitMLReplayUploadTransaction-Callback-Funktion,639
    - DelayCreationOfSimulatedClient-Callback-Funktion,640
    - DelayDestructionOfSimulatedClient-Callback-Funktion,640
    - DelayStartOfReplay-Callback-Funktion,641
    - DestroyMLReplayUploadTransaction-Callback-Funktion,641
    - FiniIdentifySimulatedClient-Callback-Funktion,642
    - GetDownloadApplyTime-Callback-Funktion,643
    - GetMLReplayAPIVersion-Callback-Funktion,644
    - GetUploadTransaction-Callback-Funktion,644
    - GlobalFini-Callback-Funktion,645
    - GlobalInit-Callback-Funktion,645
    - IdentifySimulatedClient-Callback-Funktion,646
    - MobiLink Replay-C++-Callback-Funktionen,639
    - ReportEndOfReplay-Callback-Funktion,647
  - mlscript.jar
    - MobiLink Java-Synchronisationslogik,463
  - mlsrv16
    - (*Siehe auch* MobiLink-Server)
    - alphabetische Liste der Optionen,44
    - Fehlerkontext in Meldungslog berichten,67
    - Info,44
    - Notifier,66
    - Optionen,44
    - Protokollierung,33
    - starten,30
    - stoppen,32
    - Syntax,44
    - Verbindungszeichenfolge,53
  - mlstop, Dienstprogramm
    - Deployment unter 32-Bit-Unix,778
    - Deployment unter 32-Bit-Windows,772
    - Deployment unter 64-Bit-Unix,780
    - Deployment unter 64-Bit-Windows,775
    - Methoden zum Stoppen von MobiLink-Server,32
    - Optionen,698
    - Syntax,698
  - mluser, Dienstprogramm
    - Deployment unter 32-Bit-Unix,778
    - Deployment unter 32-Bit-Windows,772
    - Deployment unter 64-Bit-Unix,780
    - Deployment unter 64-Bit-Windows,775
    - Optionen,699
    - Syntax,699
  - Mobile Device Center (*Siehe* Microsoft ActiveSync)
  - MobiLink
    - .NET-Synchronisationslogik,536
    - alphabetische Liste,272
    - außerhalb der aktuellen Sitzung betreiben,35
    - dateibasierte Downloads,213
    - Datentypen,714
    - Deployment von Anwendungen,771
    - Deployment von UltraLite-Clients,785
    - Entwicklungstipps,107
    - Ereignisüberblick,272
    - Hinweise zum Zeichensatz,765
    - Java-Synchronisationslogik,463
    - Konfliktbehandlung,124
    - konsolidierte Datenbanken,1
    - mlsrv16-Optionen,44
    - MobiLink-Server stoppen,32

- ODBC-Treiber-Unterstützung,767
- Performance,143
- Profiler,192
- Skripten,231
- starten,30
- Synchronisationsmethoden,107
- Synchronisationsserver ausführen,30
- Systemprozeduren,648
- Systemtabellen,4
- Verbindungsparameter für den Profiler,194
- Verbindungsparameter für mlsrv16,93
- MobiLink Listener-Dienstprogramm (dblsn)
  - MobiLink-Client-Deployment unter Windows,783
- MobiLink Replay, Dienstprogramm (mlreplay)
  - Syntax,702
- MobiLink Replay-C++-Callback-Funktionen
  - mlreplaycallbacks.cpp-Referenz,639
- MobiLink, direkte Konfliktverarbeitung
  - .NET-Beispiel,635
  - Java-Beispiel,634
- MobiLink, Generierungsnummern
  - dateibasierte Downloads,220
- MobiLink, gespeicherte Prozeduren (*Siehe* MobiLink-Systemprozeduren)
- MobiLink, konsolidierte Datenbanken
  - ASE,10
  - IBM DB2 LUW,12
  - Info,1
  - MySQL,17
  - SAP HANA,25
  - SAP Sybase IQ,28
- MobiLink, Logdatei-Anzeige
  - MobiLink-Serverlogs,34
- MobiLink, objektbasierter Datenfluss für Java und .NET
  - Info,627
- MobiLink, Server
  - Deployment,771
- MobiLink, Stopp-Dienstprogramm (mlstop)
  - Syntax,698
- MobiLink, Systemdatenbank
  - Info,4
- MobiLink-Agent
  - ausführen,158
  - Authentifizierung,168
  - Definition,155
  - Deployment unter Windows für SQL Anywhere-Clients,783
  - Deployment unter Windows für UltraLite-Clients,785
  - Eigenschaften,166
  - entfernte Datenbanken hinzufügen,167
  - Info,157
  - Konfigurationsfenster,161
  - konfigurieren,158
  - stoppen,163
  - Sybase Central,164
- MobiLink-Agenten-ID
  - Definition,155
- MobiLink-Agenteneigenschaften
  - Abrufintervall,166
  - anzeigen und ändern,166
  - Netzwerkprotokolloptionen für den MobiLink-Client,166
  - Synchronisationsintervall,166
  - Verbindungszeichenfolgen,166
- MobiLink-Arbiterserver-Dienstprogramm
  - Unix,712
  - Windows,711
- MobiLink-Benutzer
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - Registrieren mit dem Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
- MobiLink-Benutzer registrieren
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
- MobiLink-Benutzerauthentifizierung, Dienstprogramm (mluser)
  - Syntax,699
- MobiLink-Clients
  - Deployment,782
- MobiLink-Datentypen
  - .NET und SQL,541
  - Java und SQL,466
- MobiLink-Datenzuordnungen
  - Info,714
- MobiLink-Datenzuordnungen zwischen entfernten und konsolidierten Datenbanken
  - Info,714
- MobiLink-Dienstprogramm für die Dateiübertragung (mlfiletransfer)
  - MobiLink-Server (mlsrv16), Option -ftr,63
  - MobiLink-Server (mlsrv16), Option -ftru,64
- MobiLink-Dienstprogramme
  - Info,697

---

- MobiLink Replay (mlreplay), Syntax ,702
- MobiLink-Arbitr-Dienstprogrammssyntax, Unix,712
- MobiLink-Arbitr-Stopp, Syntax, Windows,713
- MobiLink-Arbitr-Syntax, Windows,711
- MobiLink-Benutzerauthentifizierung (mluser), Syntax,699
- MobiLink-generierte Replay-API (mlgenreplayapi), Syntax,710
- Server,697
- Syntax des MobiLink-Stopp-Dienstprogramms (mlstop),698
- MobiLink-Ereignisse
  - Liste,272
- MobiLink-Farm (*Siehe* MobiLink-Serverfarm)
- MobiLink-generiertes Replay-API-Dienstprogramm (mlgenreplayapi)
  - Syntax,710
- MobiLink-Gruppen
  - hinzufügen,167
- MobiLink-Konnektivität
  - mit HTTP und HTTPS überprüfen,32,44
- MobiLink-Logs anzeigen
  - Info,34
- MobiLink-Performance
  - Info,143
  - Schlüsselfaktoren,148
  - überwachen,152
- MobiLink-Profiler
  - Auswahlrechteck,203
  - Beispieleigenschaften,204
  - Benutzeroberfläche,198
  - Datenbank,197
  - Datenbankansichten,207
  - Deployment unter 32-Bit-Unix,778
  - Deployment unter 32-Bit-Windows,772
  - Deployment unter 64-Bit-Unix,780
  - Deployment unter 64-Bit-Windows,775
  - Detaillabelle, Fensterausschnitt,198
  - Diagramm, Fensterausschnitt,200,201
  - Diagramm-Fensterausschnitt verwenden,201
  - Info,192
  - neue Überwachung,206
  - Optionen,194,204
  - Sitzungseigenschaften,204
  - starten,193,194
  - Statistikeigenschaften,209
  - stoppen,196
  - Übersicht, Fensterausschnitt,203
  - Überwachungen festlegen,205
  - Überwachungsmanager,205
  - verwenden,197
  - vorherige Sitzung löschen,196
  - vorherige Sitzung öffnen,196
  - zoomen,201
- MobiLink-Profiler starten
  - Info,193
- MobiLink-Projekt
  - Definition,154
- MobiLink-Server
  - (*Siehe auch* mlsrv16)
  - ausführen,30
  - MobiLink Replay, Dienstprogramm (mlreplay),702
  - MobiLink, Stopp-Dienstprogramm (mlstop),698
  - MobiLink-generiertes Replay-API-Dienstprogramm (mlgenreplayapi),710
  - Optionen,44
  - starten,30
  - Syntax,44
- MobiLink-Server, gemeinsam genutzter Zustand
  - Serverfarm,41
- MobiLink-Server, Logdatei-Anzeige
  - MobiLink-Serverlogs,34
- MobiLink-Server, Optionen
  - Info,44
- MobiLink-Server-API für .NET
  - DateTimeWithTimeZone-Klasse,553
  - DBCommand-Schnittstelle,560
  - DBConnection-Schnittstelle,563
  - DBConnectionContext-Schnittstelle,565
  - DBParameter-Klasse,571
  - DBParameterCollection-Klasse,575
  - DBRowReader-Schnittstelle,584
  - DownloadData-Schnittstelle,586
  - DownloadTableData-Schnittstelle,588
  - FatalException-Klasse,593
  - LogCallback-Delegat,623
  - LogMessage-Klasse,595
  - NetworkData-Schnittstelle,597
  - ScriptExecutionException-Klasse,601
  - ServerContext-Schnittstelle,603
  - ServerException-Klasse,608
  - ShutdownCallback-Delegat,623
  - SpatialUtilities-Klasse,610
  - SQLType-Enumeration,624
  - SynchronizationException-Klasse,612

- UpdateDataReader-Schnittstelle,614
- UploadData-Schnittstelle,616
- UploadedTableData-Schnittstelle,618
- MobiLink-Server-API für .NET, Referenz
  - iAnywhere.MobiLink.Script-Namespace,552
- MobiLink-Server-API für Java, Referenz
  - ianywhere.ml.script-Paket,477
- MobiLink-Server-Java-API
  - DBConnectionContext-Schnittstelle,477
  - DownloadData-Schnittstelle,483
  - DownloadTableData-Schnittstelle,485
  - InOutInteger-Schnittstelle,492
  - InOutString-Schnittstelle,494
  - LogListener-Schnittstelle,495
  - LogMessage-Klasse,496
  - NetworkData-Schnittstelle,501
  - ServerContext-Schnittstelle,505
  - ServerException-Klasse,514
  - ShutdownListener-Schnittstelle,516
  - SpatialUtilities-Klasse,517
  - SynchronizationException-Klasse,519
  - TimestampWithTimeZone-Klasse,520
  - UpdateResultSet-Schnittstelle,526
  - UploadData-Schnittstelle,528
  - UploadedTableData-Schnittstelle,530
- MobiLink-Serverfarm
  - Failover,41
  - Lastverteilung,41
  - MobiLink mlsrv16 -Option -lsc,65
- MobiLink-Skripten
  - Liste,272
- MobiLink-Statistikeigenschaften
  - MobiLink-Profiler,209
- MobiLink-Synchronisation
  - .NET-Klassen erstellen,542
  - .NET-Synchronisationslogik,536
  - datebasierte Downloads,213
  - Java-Klassen erstellen,468
  - Java-Synchronisationslogik,463
  - konsolidierte Datenbanken,1
  - neu startbare Downloads,135
  - Performance,143
  - Überblick über Ereignisse,272
- MobiLink-Synchronisationslogik
  - .NET,536
  - alphabetische Liste der Skripten,272
  - Datentypen für .NET und SQL,541
  - Datentypen für Java und SQL,466
- Java,463
  - Skripten erstellen,231
  - Synchronisationsmethoden,107
- MobiLink-Synchronisationsserver (*Siehe* MobiLink-Server)
- MobiLink-Synchronisationsskripten
  - .NET-Klassen erstellen,541,542
  - alphabetische Liste der Skripten,272
  - Datenbanktransaktionen in .NET bewahren,540
  - Datenbanktransaktionen in Java bewahren,466
  - Datenbanktransaktionen und .NET-Klassen,540
  - Datenbanktransaktionen und Java-Klassen,466
  - Fehlersuche in Java-Klassen,468
  - Info,231
  - Java-Klassen erstellen,467,468
- MobiLink-Systemdatenbank (MLSD)
  - Option -cs für mlsrv16,57
- MobiLink-Systemprozeduren
  - Info,648
- MobiLink-Systemtabellen
  - in konsolidierter Datenbank erstellen,3
  - Info,4
- MobiLink-Verbindungen
  - debuggen,44
- Mobility (*Siehe* MobiLink)
- modify\_error\_message
  - Verbindungsereignis,389
- modify\_last\_download\_timestamp
  - Verbindungsereignis,392
- modify\_next\_last\_download\_timestamp
  - Verbindungsereignis,395
- modify\_user
  - Verbindungsereignis,398
- Month-Eigenschaft
  - DateTimeWithTimeZone-Klasse [MobiLink-Server-API für .NET],559
- MySQL
  - konsolidierte MobiLink-Datenbank,17
  - MobiLink-Datenzuordnung,740
- MySQL, konsolidierte Datenbank
  - MobiLink,17

## N

- Name des entfernten Schemas
  - Definition,156
  - hinzufügen,164
  - importieren,164

## NetworkData-Eigenschaft:

DBConnectionContext-Schnittstelle [MobiLink-Server-API für .NET],570

## NetworkData-Schnittstelle [MobiLink-Server-API für .NET]

Beschreibung,597

ClientCertificates-Eigenschaft,600

GetHTTPHeaderValue-Methode,599

GetHTTPHeaderValues-Methode,599

HTTPHeaders-Eigenschaft,600

IsEndToEndEncrypted-Eigenschaft,601

IsHTTP-Eigenschaft,601

IsTLS-Eigenschaft,601

## NetworkData-Schnittstelle [MobiLink-Server-Java-API]

Beschreibung,501

getCertificateChain-Methode,503

getHTTPHeaders-Methode,503

getHTTPHeaderValue-Methode,504

getHTTPHeaderValues-Methode,504

isEndToEndEncrypted-Methode,505

isHTTP-Methode,505

isTLS-Methode,505

## Netzwerkparameter

MobiLink-Server (mlsrv16), Option -x,93

## Netzwerkprotokolle

MobiLink-Server,94

MobiLink-Server (mlsrv16) mit HTTP,96

MobiLink-Server (mlsrv16) mit HTTPS,97

MobiLink-Server (mlsrv16), Option -x mit TCP/IP,94

MobiLink-Server (mlsrv16), Option -x mit TLS über TCP/IP,95

## Neu startbare Downloads

MobiLink,135

## NextRow-Methode

DBRowReader-Schnittstelle [MobiLink-Server-API für .NET],585

## Nicht blockierende Downloadbestätigung

Info,137

nonblocking\_download\_ack,

Verbindungsereignis,401

publication\_nonblocking\_download\_ack,

Verbindungsereignis,407

## Nicht-relationale Datenbanken

mit MobiLink synchronisieren,627

## nonblocking\_download\_ack

Verbindungsereignis,401

## Notifier

Deployment unter 32-Bit-Unix,778

Deployment unter 32-Bit-Windows,772

Deployment unter 64-Bit-Unix,780

Deployment unter 64-Bit-Windows,775

starten,66

## O

o.

MobiLink, Präfix benannter Parameter,237

## Objektbasierter Datenfluss (*Siehe* Direkte Zeilenbehandlung)

## Objekte

MobiLink-Server, API für Java,477

## ODBC

mehrere Fehler in MobiLink,270

MobiLink-Treiber,767

Oracle-Treiber,768

## ODBC-Treiber

MobiLink Unterstützung,767

MobiLink-Zeichensatzkonvertierung,767

Oracle,768

## Optimierung, Performance

MobiLink,149

## Option, -ct

MobiLink-Serveroption (mlsrv16),58

## Option, -zu

MobiLink-Serveroption (mlsrv16),104

## Optionen

Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699

mlsrv16,44

MobiLink Replay, Dienstprogramm (mlreplay),702

MobiLink, Stopp-Dienstprogramm (mlstop),698

MobiLink-generiertes Replay-API-Dienstprogramm (mlgenreplayapi),710

MobiLink-Server (mlsrv16),44

## Optionen, Fenster

MobiLink-Profiler,204

## Oracle

als konsolidierte MobiLink-Datenbank,20

LONG-Daten synchronisieren,753

MobiLink-Datenzuordnung,746

MobiLink-Isolationsstufen,140

Oracle-Treiber,768

räumliche Daten synchronisieren,7

Sequenzen bei der MobiLink-Synchronisation,22

- XMLTYPE-Datentyp,23
- Oracle, VARRAY
  - Beispiel,24
  - Einschränkungen,25
  - in gespeicherten Prozeduren,24
- Oracle-Treiber
  - ODBC,768
  - verschlüsseln, Kennwörter,768
- Outbound Enabler
  - MobiLink-Server (mlsrv16), Option -x,99
- P**
- Paketierter Download
  - MobiLink, dateibasierte Downloads,213
- Parallelität
  - MobiLink-Performance,143
- Parameter
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
  - MobiLink-Server (mlsrv16),44
  - Synchronisationsskripten,236
- ParameterName-Eigenschaft
  - DBParameter-Klasse [MobiLink-Server-API für .NET],573
- Parameters-Eigenschaft
  - DBCommand-Schnittstelle [MobiLink-Server-API für .NET],563
- Parse-Methode
  - DateTimeWithTimeZone-Klasse [MobiLink-Server-API für .NET],556
- Partial Download Retained, Synchronisationsparameter
  - neu startbare Downloads,136
- Passthrough-Modus
  - (*Siehe auch* SQL Passthrough)
- Performance
  - automatische Anpassung von Datenbank-Worker-Threads,145
  - inkrementelle MobiLink-Uploads,145
  - MobiLink,143
  - MobiLink, Ausführlichkeit der Protokollierung,147
  - MobiLink, BLOBs synchronisieren,146
  - MobiLink, maximale Datenbankverbindungen,146
  - MobiLink-Datenbank-Worker-Threads,144
  - MobiLink-Datenbankverbindungen,152
  - MobiLink-Downloads,148
  - MobiLink-Konflikt,143
  - MobiLink-Mehrbenutzerbetrieb,143
  - MobiLink-Optimierung,149
  - MobiLink-Server (mlsrv16), Option -sm,81
  - MobiLink-Skriptausführung,147
  - MobiLink-Speicher,146
  - MobiLink-Synchronisationslogik,147
  - MobiLink-Synchronisationspriorität,148
  - MobiLink-Threads,144
  - MobiLink-Zeilen-Uploads,148
  - transaktionale MobiLink-Uploads,145
- port, Protokolloption
  - MobiLink-Server (mlsrv16), Option -x für HTTP,96
  - MobiLink-Server (mlsrv16), Option -x für HTTPS,98
  - MobiLink-Server (mlsrv16), Option -x für TCP/IP,94
  - MobiLink-Server (mlsrv16), Option -x für TLS über TCP/IP,95
- Präfixe
  - MobiLink, benannte Parameter,237
- Precision-Eigenschaft
  - DBParameter-Klasse [MobiLink-Server-API für .NET],573
- Prepare-Methode
  - DBCommand-Schnittstelle [MobiLink-Server-API für .NET],563
- prepare\_for\_download
  - Verbindungsereignis,404
- prepare\_for\_download-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- Primärschlüssel
  - MobiLink, Eindeigkeitstechniken,117
  - MobiLink, Info,107
  - Oracle-Sequenzen,22
- Primärschlüsselpools
  - eindeutige Werte mit GLOBAL AUTOINCREMENT für MobiLink generieren,119
  - MobiLink, eindeutige Primärschlüssel,122
  - MobiLink-Beispiel,124
- Priorität, Synchronisation
  - MobiLink-Performance,148
- Private Assembly
  - in MobiLink implementieren,548
- Privilegien
  - MobiLink-Server,31
- ProcResults
  - Oracle-Treiberoption,768

- 
- Profiler
    - MobiLink-Profiler,192
  - Profilerstellung
    - Synchronisationen in MobiLink,192
  - Profilerstellungsdatenbank
    - anzeigen,207
    - MobiLink,197
  - Profilerstellungssitzung
    - beenden,196
    - starten,194
    - Überwachung erstellen,206
    - vorherige Sitzung löschen,196
    - vorherige Sitzung öffnen,196
  - Projekte
    - Gruppen hinzufügen,167
  - Protokolle
    - (*Siehe auch* Netzwerkprotokolle)
    - MobiLink-Server,94
    - MobiLink-Server (mlsrv16) mit HTTP,96
    - MobiLink-Server (mlsrv16), Option -x mit TCP/IP,94
    - MobiLink-Server (mlsrv16), Option -x mit TLS über TCP/IP,95
    - MobiLink-Server (mlsrv16) mit HTTPS,97
  - Protokollieren
    - Aktionen des MobiLink-Servers,33
    - MobiLink-Performance,147
  - Protokollieren, Aktionen des MobiLink-Servers
    - Info,33
  - Prozeduraufrufe
    - konsolidierte MobiLink-Datenbanken in SQL Server,16
  - Prozeduren
    - MobiLink,648
  - Prozeduren, Ergebnisse zurückgeben
    - Oracle-Treiberoption,768
  - Pseudocode
    - MobiLink-Ereignisse,272
  - publication\_nonblocking\_download\_ack
    - Verbindungsereignis,407
  - Punkt-zu-Punkt-Verschlüsselungstyp
    - MobiLink-Serveroption (mlsrv16),93
  - R**
  - r.
    - MobiLink, Präfix benannter Parameter,237
  - Räumliche Daten
    - Synchronisation,6
  - Räumliche Daten synchronisieren
    - Anforderungen an die SRID,7
    - benannte Parametern,7
    - Einschränkungen,7
    - Hinweise zu Oracle,7
    - Upload- und Download-Skripten,8
  - READPAST, Tabellen-Hint
    - download\_cursor-Probleme,330
    - download\_delete\_cursor-Probleme,333
    - upload\_fetch-Probleme,438
  - Regeln
    - (*Siehe auch* Übertragungsregeln)
  - Registrieren
    - Methoden als MobiLink-Skripten,648
  - Reine Download-Synchronisation
    - erforderliche Skripten,258
    - Info,116
  - Reine Upload- und reine Download-Synchronisation
    - Info,116
  - Reine Upload-Synchronisation
    - Info,116
  - Reiner Download
    - (*Siehe auch* Reiner Download)
  - Relay Server Outbound Enabler
    - (*Siehe auch* Outbound Enabler)
  - Relay Server-Hostingdienst
    - (*Siehe auch* Sybase Hosted Relay Service)
  - relayserver (*Siehe* Relay Server )
  - Remoteadministration (*Siehe* zentrale Administration entfernter Datenbanken)
  - Remove-Methode
    - DBParameterCollection-Klasse [MobiLink-Server-API für .NET],580
  - RemoveAt-Methode
    - DBParameterCollection-Klasse [MobiLink-Server-API für .NET],581
  - removeErrorListener-Methode
    - ServerContext-Schnittstelle [MobiLink-Server-Java-API],512
  - removeShutdownListener-Methode
    - ServerContext-Schnittstelle [MobiLink-Server-Java-API],513
  - removeWarningListener-Methode
    - ServerContext-Schnittstelle [MobiLink-Server-Java-API],513
  - Replikation
    - (*Siehe auch* MobiLink)
-

- report\_error
  - Syntax, 271
  - Verbindungsereignis, 410
- report\_odbc\_error
  - Verbindungsereignis, 414
- ReportEndOfReplay-Callback-Funktion [mlreplaycallbacks.cpp]
  - Beschreibung, 647
- resolve\_conflict
  - Tabellenereignis, 417
  - verwenden, 128
- Resume Partial Download, Synchronisationsparameter
  - neu startbare Downloads, 136
- Rollback-Methode
  - DBConnection-Schnittstelle [MobiLink-Server-API für .NET], 565
- rsa-Protokolloption
  - MobiLink, mlsrv16-Option -x für HTTPS, 98
  - MobiLink, mlsrv16-Option -x für TCP/IP, 95
- rshost (*Siehe* Relay Server Status-Manager)
- RSOE (*Siehe* Outbound Enabler)
- rtnotifier
  - Syntax, 180
- Rückgabewerte
  - .NET-Synchronisation, 542
  - Java-Synchronisation, 468
- S**
- s.
  - MobiLink, Präfix benannter Parameter, 237
- Sanftes Herunterfahren
  - MobiLink, Stopp-Dienstprogramm (mlstop), 698
- SAP HANA
  - konsolidierte MobiLink-Datenbank, 25
  - MobiLink-Datenzuordnung, 757
  - MobiLink-Synchronisation, 26
- SAP Sybase IQ
  - als konsolidierte MobiLink-Datenbank, 28
  - MobiLink-Datenzuordnung, 761
- Scale-Eigenschaft
  - DBParameter-Klasse [MobiLink-Server-API für .NET], 574
- Schattentabellen
  - download\_delete\_cursor-Skripten schreiben, 267
- Schemata
  - konsolidierte Tabellen in Beziehung zu entfernten MobiLink-Tabellen setzen, 2
- Schlüsselpools
  - MobiLink-Synchronisationsanwendung, 122
- Schnellstart
  - MobiLink, direkte Zeilenbehandlung, 629
- ScriptExecutionException-Klasse [MobiLink-Server-API für .NET]
  - Beschreibung, 601
  - ScriptExecutionException-Konstruktor, 602
- ScriptExecutionException-Konstruktor
  - ScriptExecutionException-Klasse [MobiLink-Server-API für .NET], 602
- Second-Eigenschaft
  - DateTimeWithTimeZone-Klasse [MobiLink-Server-API für .NET], 559
- Selbstreferenzierende Tabellen
  - MobiLink, 140
- Selektives und gemeinsames Nutzen (*Siehe* Partitionierung)
- Sequenz von MobiLink-Ereignissen
  - Pseudocode, 276
- Sequenzen
  - Primärschlüssel-Eindeutigkeit bei der MobiLink-Synchronisation, 22
- Server
  - MobiLink-Serverdienstprogramm (mlsrv16), 30
- ServerContext-Schnittstelle [MobiLink-Server-API für .NET]
  - Beschreibung, 603
  - ErrorListener-Ereignis, 607
  - getProperties-Methode, 604
  - getPropertiesByVersion-Methode, 605
  - getPropertySetNames-Methode, 605
  - GetStartClassInstances-Methode, 606
  - InfoListener-Ereignis, 607
  - MakeConnection-Methode, 607
  - Shutdown-Methode, 607
  - ShutdownListener-Ereignis, 608
  - WarningListener-Ereignis, 608
- ServerContext-Schnittstelle [MobiLink-Server-Java-API]
  - addErrorListening-Methode, 507
  - addInfoListener-Methode, 507
  - addShutdownListener-Methode, 508
  - addWarningListener-Methode, 509
  - Beschreibung, 505
  - getProperties-Methode, 509
  - getPropertiesByVersion-Methode, 510
  - getPropertySetNames-Methode, 510

---

- getStartClassInstances-Methode,511
- makeConnection-Methode,511
- removeErrorListener-Methode,512
- removeInfoListener-Methode,512
- removeShutdownListener-Methode,513
- removeWarningListener-Methode,513
- Shutdown-Methode,514
- ServerException-Klasse [MobiLink-Server-API für .NET]
  - Beschreibung,608
  - ServerException-Konstruktor,609
- ServerException-Klasse [MobiLink-Server-Java-API]
  - Beschreibung,514
  - ServerException-Konstruktor,515
- ServerException-Konstruktor
  - ServerException-Klasse [MobiLink-Server-API für .NET],609
  - ServerException-Klasse [MobiLink-Server-Java-API],515
- Serverfarm
  - Lastverteilung,41
  - MobiLink,41
  - MobiLink-Server (mlsrv16), Option -zs,103
- Serverinitiierte entfernte Aufgabe
  - Definition,156
  - SIRT,179
- Serversystemprozeduren
  - MobiLink,648
- ServiceName
  - Oracle-Treiberoption,768
- SET NOCOUNT
  - konsolidierte MobiLink-Datenbanken in SQL Server,16
- SetNewRowValues-Methode
  - UpdateDataReader-Schnittstelle [MobiLink-Server-API für .NET],615
- setNewRowValues-Methode
  - UpdateResultSet-Schnittstelle [MobiLink-Server-Java-API],527
- SetOldRowValues-Methode
  - UpdateDataReader-Schnittstelle [MobiLink-Server-API für .NET],615
- setOldRowValues-Methode
  - UpdateResultSet-Schnittstelle [MobiLink-Server-Java-API],527
- SetSRID-Methode
  - SpatialUtilities-Klasse [MobiLink-Server-API für .NET],612
- setSRID-Methode
  - SpatialUtilities-Klasse [MobiLink-Server-Java-API],518
- setTimeZoneOffsetHours-Methode
  - TimestampWithTimeZone-Klasse [MobiLink-Server-Java-API],524
- setTimeZoneOffsetMinutes-Methode
  - TimestampWithTimeZone-Klasse [MobiLink-Server-Java-API],524
- Setup
  - MobiLink .NET-Synchronisationslogik,537
  - MobiLink Java-Synchronisationslogik,463
- Setup, Skripten
  - MobiLink-Systemdatenbank,4
- Setupskripten
  - konsolidierte MobiLink-Datenbanken,3
- setValue-Methode
  - InOutInteger-Schnittstelle [MobiLink-Server-Java-API],494
  - InOutString-Schnittstelle [MobiLink-Server-Java-API],495
- Shutdown-Methode
  - ServerContext-Schnittstelle [MobiLink-Server-API für .NET],607
  - ServerContext-Schnittstelle [MobiLink-Server-Java-API],512,514
- ShutdownCallback-Delegat [MobiLink-Server-API für .NET]
  - Beschreibung,623
- ShutdownListener-Ereignis
  - ServerContext-Schnittstelle [MobiLink-Server-API für .NET],608
- ShutdownListener-Schnittstelle [MobiLink-Server-Java-API]
  - Beschreibung,516
  - shutdownPerformed-Methode,517
- shutdownPerformed-Methode
  - ShutdownListener-Schnittstelle [MobiLink-Server-Java-API],517
- Sicherheit
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
- SIRT (*Siehe* serverinitiierte entfernte Aufgabe)
  - bereitgestellte entfernte Aufgabe einleiten,177
  - serverinitiierte entfernte Aufgabe,179
- sis (*Siehe* serverinitiierte Synchronisation)
- Sitzungseigenschaften
  - MobiLink-Profiler,204

---

- Sitzungsweite Variablen
  - konsolidierte MobiLink-Datenbanken in IBM DB2,14
- Size-Eigenschaft
  - DBParameter-Klasse [MobiLink-Server-API für .NET],574
- Skripten
  - .NET-Tabellenskripten hinzufügen und löschen,654
  - .NET-Verbindungsskripten hinzufügen und löschen,653
  - erforderlich in MobiLink,258
  - globale Skriptversionen,256
  - Info zu MobiLink,231
  - Java-Tabellenskripten hinzufügen und löschen,656
  - Java-Verbindungsskripten hinzufügen und löschen,655
  - MobiLink-Ereignisse,272
  - Räumliche Daten synchronisieren,8
  - Skripten für Zeilen-Download erstellen,265
  - Skripten für Zeilen-Upload erstellen,262
  - SQL-Tabellenskripten hinzufügen und löschen,668
  - SQL-Verbindungsskripten hinzufügen und löschen,651
  - Tabellenskripten,236
  - Überblick über MobiLink-Ereignisse,272
  - unterstützte DBMS-Skriptstrategien,5
  - Verbindungsskripten,235
  - Versionen,254
  - zur konsolidierten Datenbank in MobiLink hinzufügen,258
- Skripten für die Synchronisation von räumlichen Daten
  - IBM DB2-Beispiel,10
  - Microsoft SQL Server-Beispiel,9
  - MySQL-Beispiel,10
  - Oracle-Beispiel,9
  - SQL Anywhere-Beispiel,9
  - Upload- und Download-Skripten,8
- Skriptparameter
  - Info über MobiLink,236
  - last\_download,109
  - last\_table\_download,109
- Skripttypen
  - MobiLink,235
- Skriptversion löschen (*Siehe* Skriptversion entfernen)
- Skriptversionen
  - entfernen,257
  - global,256
  - hinzufügen,256
  - reservierte Namen,255
  - über MobiLink-Synchronisation,254
- Skriptversionen entfernen
  - MobiLink,257
- Snapshot-Isolation
  - MobiLink,140
  - MobiLink-Server (mlsrv16), Option -dsd zum Deaktivieren,60
  - MobiLink-Server (mlsrv16), Option -dt für SQL Server,60
  - MobiLink-Server (mlsrv16), Option -esu zum Aktivieren für Uploads,62
- Snapshot-Synchronisation
  - Info,111
- Sofortiges Herunterfahren
  - MobiLink, Stopp-Dienstprogramm (mlstop),698
- Sommerzeit
  - MobiLink,111
- Sortierreihenfolge
  - Zeichen und MobiLink-Synchronisation,765
- Spaltengrößen
  - konsolidierte MobiLink-Datenbanken in ASE,11
- SpatialUtilities-Klasse [MobiLink-Server-API für .NET]
  - Beschreibung,610
  - CreateSpatialValue-Methode,611
  - GetBytes-Methode,611
  - GetSRID-Methode,611
  - SetSRID-Methode,612
- SpatialUtilities-Klasse [MobiLink-Server-Java-API]
  - Beschreibung,517
  - createSpatialValue-Methode,517
  - getBytes-Methode,518
  - getSRID-Methode,518
  - setSRID-Methode,518
- Speichernutzung
  - MobiLink-Server,41
- Sperrenerweiterung
  - konsolidierte MobiLink-Datenbanken in IBM DB2,14
- Sprachenbibliotheken
  - MobiLink-Server-Deployment unter 32-Bit-Windows,772,778
  - MobiLink-Server-Deployment unter 64-Bit-Windows,775,780
  - SQL Anywhere

- als konsolidierte MobiLink-Datenbank,28
- MobiLink-Isolationsstufen,140
- SQL Anywhere 16 - Oracle-ODBC-Treiber
  - Info,768
  - Unix-Konfiguration,770
  - Windows-Konfiguration,770
- SQL Anywhere 16, Oracle-ODBC-Treiber
  - MobiLink-Unterstützung,767
- SQL Server
  - (*Siehe auch* Microsoft SQL Server)
  - als konsolidierte MobiLink-Datenbank,15
  - MobiLink-Datenzuordnung,733
- SQL-.NET, Datentypen
  - MobiLink, .NET-Synchronisationslogik,541
- SQL-Java, Datentypen
  - Info,466
- SQL-Synchronisationslogik
  - MobiLink,231
  - MobiLink-Performance,147
- SQL-Syntax
  - MobiLink-Server (mlsrv16),44
- SQL\_TXN\_READ\_COMMITTED
  - MobiLink-Isolationsstufen,140
- SQLType-Enumeration [MobiLink-Server-API für .NET]
  - Beschreibung,624
- SRIDs
  - Synchronisation von räumlichen Daten,7
- Standard, global autoincrement
  - MobiLink, deklarieren,120
- Standardisierungsstufen
  - MobiLink,140
- start\_time-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- Starten
  - MobiLink-Profiler (mlprof),194
  - MobiLink-Server,30
- Startklassen
  - DMLStartClasses, Option für Java,79
  - MLStartClasses, Option für .NET,78
  - MobiLink, .NET-Synchronisationslogik,542
  - MobiLink, Java-Synchronisationslogik,471
- StaticCursorLongColBuffLen
  - ASE,11
- Statistikeigenschaften
  - MobiLink,209
- Statistiken
  - MobiLink,209

- STOP SYNCHRONIZATION DELETE-Anweisung
  - SQL Anywhere-Clients,133
  - Syntax,267
- Stopp-Dienstprogramm (mlstop)
  - Syntax,698
- Stoppen
  - MobiLink, Stopp-Dienstprogramm (mlstop),698
  - MobiLink-Server,32
  - Upload der Löschungen für SQL Anywhere-Clients,133
- Sybase Adaptive Server Enterprise (*Siehe* Adaptive Server Enterprise)
- sync-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- sync\_deadlocks-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- sync\_errors-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- sync\_requests-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- sync\_tables-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- sync\_warnings-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- syncase.sql
  - Info,10
- syncdb2long.sql
  - Info,12
- synchrona.sql
  - Info,25
- Synchronisation
  - alphabetische Liste der Skripten,272
  - Datentypzuordnungen in MobiLink,714
  - Ereignisüberblick,272
  - Konfliktlösung,124
  - konsolidierte Datenbanken,1
  - Methoden,107
  - MobiLink für ASE-Datentypen,714
  - MobiLink, IBM DB2 LUW-Datentypen,723
  - MobiLink, Microsoft SQL Server-Datentypen,733
  - MobiLink, MySQL-Datentypen,740
  - MobiLink, Oracle-Datentypen,746
  - MobiLink, SAP HANA-Datentypen,757
  - MobiLink, SAP Sybase IQ-Datentypen,761
  - MobiLink-Dienstprogramme,697
  - MobiLink-Server ausführen,30
  - MobiLink-Skripten in .NET erstellen,536
  - MobiLink-Skripten in Java erstellen,463

- MobiLink-Systemprozeduren,648
- MobiLink-Zeichensätze,765
- MobiLink-Zeichensatzkonvertierung,766
- neu startbare Downloads,135
- Performance-Tipps,143
- Protokolloptionen für mlsrv16,93
- räumliche Daten,6
- Skripten erstellen,231
- Snapshot,111
- Verbindungsparameter für den Profiler,193
- Viele-zu-Viele-Beziehungen,115
- Vorgang,234
- Zeilen einlesen,265
- Zeilen löschen,267
- Synchronisation durch reinen Download
  - (*Siehe auch* reine Download-Synchronisation)
- Synchronisation durch reinen Upload
  - (*Siehe auch* reine Upload-Synchronisation)
- Synchronisation von räumlichen Daten
  - Anforderungen an die SRID,7
  - benannte Parametern,7
  - Einschränkungen,7
  - Hinweise zu Oracle,7
  - Upload- und Download-Skripten,8
- Synchronisation, neue entfernte Datenbanken
  - MobiLink, dateibasierte Downloads,217
- Synchronisation, serverinitiiert (*Siehe* serverinitiierte Synchronisation)
- Synchronisationsbenutzer
  - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
- Synchronisationsdatenstrom-Bibliotheken
  - MobiLink-Server-Deployment unter 32-Bit-Untix,778
  - MobiLink-Server-Deployment unter 32-Bit-Windows,772
  - MobiLink-Server-Deployment unter 64-Bit-Untix,780
  - MobiLink-Server-Deployment unter 64-Bit-Windows,775
- Synchronisationseigenschaften
  - MobiLink-Profiler,205
- Synchronisationsereignisse
  - alphabetische Liste der Ereignisskripten,272
  - begin\_connection\_autocommit, ASE-Verbindungsereignis,301
  - Info,272
  - Info zur MobiLink-Synchronisation,272
  - MobiLink, Download,281
  - MobiLink, Upload,279
- Synchronisationsfehler
  - Behandlung in MobiLink,270
  - Fehlersuche,61
- Synchronisationsintervall
  - MobiLink-Agent,166
- Synchronisationslogik
  - MobiLink,231
- Synchronisationsmethoden
  - fehlgeschlagene Downloads,134
  - gespeicherte Prozeduren für Download,138
  - Info,107
  - Primärschlüsselpools,122
  - Snapshot-basierte Synchronisation,111
  - verteilen,113
  - Zeilen löschen,132
  - zeitstempelbasierte Synchronisation,107
- Synchronisationsparameter
  - HTTP-Synchronisation,94
  - HTTPS-Synchronisation,94
  - TCP/IP-Synchronisation,94
- Synchronisationsserver (*Siehe* MobiLink-Server)
- Synchronisationsskripten
  - .NET,536
  - .NET-Methoden,542
  - Ausführung,234
  - Beispiel,233
  - DBMS-Abhängigkeiten,5
  - download\_cursor,266
  - für .NET implementieren,537
  - für Java implementieren,463
  - handle\_error, Ereignis,270
  - hinzufügen und löschen,258
  - Info,231
  - Java,463
  - Java-Methoden,468
  - mit gespeicherten Prozeduren hinzufügen oder löschen,258
  - mit Sybase Central hinzufügen,259
  - MobiLink-Ereignisse,272
  - Parameter,236
  - report\_error,271
  - Skripten für Zeilen-Download erstellen,265
  - Skripten für Zeilen-Upload erstellen,262
  - Skriptversionen,254
  - Tabellenskripten,236
  - Typen,235

- 
- unterstützte DBMS-Skriptstrategien,5
  - Verbindungsskripten,235
  - Synchronisationssubskriptionen
    - (*Siehe auch* Subskriptionen)
  - Synchronisationsverfahren
    - Zeilen auslesen,262
  - Synchronisieren
    - (*Siehe auch* Synchronisation)
    - räumliche Daten,6
  - Synchronisieren, Datenquellen, die keine konsolidierten Datenbanken sind
    - Info,627
  - Synchronisieren, selbstreferenzierende Tabellen
    - MobiLink,140
  - synchronization\_statistics
    - Tabellenereignis,425
    - Verbindungsereignis,421
  - SynchronizationException-Klasse [MobiLink-Server-API für .NET]
    - Beschreibung,612
    - SynchronizationException-Konstruktor,613
  - SynchronizationException-Klasse [MobiLink-Server-Java-API]
    - Beschreibung,519
    - SynchronizationException-Konstruktor,520
  - SynchronizationException-Konstruktor
    - SynchronizationException-Klasse [MobiLink-Server-API für .NET],613
    - SynchronizationException-Klasse [MobiLink-Server-Java-API],520
  - syncmss.sql
    - Info,15
  - syncora.sql
    - Info,20
  - SyncRoot-Eigenschaft
    - DBParameterCollection-Klasse [MobiLink-Server-API für .NET],583
  - syncsa.sql
    - Info,28
  - Syntax
    - Dienstprogramm für die MobiLink-Benutzerauthentifizierung (mluser),699
    - MobiLink, Stopp-Dienstprogramm (mlstop),698
    - MobiLink-Server (mlsrv16),44
    - MobiLink-Skripten,272
    - MobiLink-Synchronisation, Dienstprogramme,697
    - MobiLink-Systemprozeduren,648
  - Systemdatenbank
    - MobiLink,4
  - Systemparameter
    - MobiLink-Skripten,237
  - Systemprozeduren
    - alphabetische Liste der MobiLink-Systemprozeduren,648
    - ml\_add\_cs,651
    - ml\_add\_dcs,653
    - ml\_add\_dts,654
    - ml\_add\_jcs,655
    - ml\_add\_lcs,657
    - ml\_add\_lcs\_chk,658
    - ml\_add\_lts,658
    - MobiLink ,648
  - Systemprozeduren für LDAP
    - MobiLink-Server,649
  - Systemprozeduren, Eigenschaften hinzufügen oder löschen
    - MobiLink-Server,649
  - Systemprozeduren, Skripten hinzufügen oder löschen
    - MobiLink-Server,648
  - Systemtabellen
    - in konsolidierter Datenbank in MobiLink erstellen,3
- ## T
- Tabellen
    - konsolidierte Tabellen in Beziehung zu entfernten MobiLink-Tabellen setzen,2
    - verteilen,113
  - Tabellenkalkulation
    - mit MobiLink synchronisieren,627
  - Tabellenskript hinzufügen, Assistent verwenden,260
  - Tabellenskripten
    - alphabetische Liste der MobiLink-Skripten,272
    - Definition,235
    - Hinzufügen von .NET-Skripten,654
    - Hinzufügen von Java-Skripten,656
    - Hinzufügen von SQL-Skripten,668
    - Info,236
    - Löschen von .NET-Skripten,654
    - Löschen von Java-Skripten,656
    - Löschen von SQL-Skripten,668
    - mit Sybase Central hinzufügen,260
    - MobiLink,232
  - Tablespace-Kapazität

- konsolidierte MobiLink-Datenbanken in IBM DB2,14
  - TCP/IP
    - MobiLink-Server (mlsrv16), Option -x,94
  - Teilmengen
    - Herunterladen, Datenteilmengen an entfernte,113
  - Text-Eigenschaft
    - LogMessage-Klasse [MobiLink-Server-API für .NET],596
  - Textdateien
    - mit MobiLink synchronisieren,627
  - this-Eigenschaft
    - DBParameterCollection-Klasse [MobiLink-Server-API für .NET],583
  - Threading
    - (*Siehe auch* Threads)
    - MobiLink-Performance,144
  - Threads
    - automatische Anpassung,145
    - MobiLink, Worker-Threads und Performance,144
  - time\_statistics
    - Tabellenereignis,432
    - Verbindungsereignis,428
  - Timeout
    - Option -tc (mlsrv16),82
    - Option -tf (mlsrv16),83
  - TimestampWithTimeZone-Klasse [MobiLink-Server-Java-API]
    - Beschreibung,520
    - equals-Methode,522
    - getTimeZoneOffsetHours-Methode,524
    - getTimeZoneOffsetMinutes-Methode,524
    - setTimeZoneOffsetHours-Methode,524
    - setTimeZoneOffsetMinutes-Methode,524
    - TimestampWithTimeZone-Konstruktor,521
    - toString-Methode,525
    - toTimestampWithTimeZone-Methode,525
    - valueOf-Methode,525
  - TimestampWithTimeZone-Konstruktor
    - TimestampWithTimeZone-Klasse [MobiLink-Server-Java-API],521
  - TimeZoneHour-Eigenschaft
    - DateTimeWithTimeZone-Klasse [MobiLink-Server-API für .NET],559
  - TimeZoneMinute-Eigenschaft
    - DateTimeWithTimeZone-Klasse [MobiLink-Server-API für .NET],560
  - Tipps
    - Performance von MobiLink,143
    - Synchronisationsmethoden,107
  - TLS
    - (*Siehe auch* Transportschichtssicherheit)
    - MobiLink-Client-Deployment unter Unix,784
    - MobiLink-Client-Deployment unter Windows,783
    - MobiLink-Server (mlsrv16), Option -x,95
    - MobiLink-Server-Deployment unter 32-Bit-Unix,778
    - MobiLink-Server-Deployment unter 32-Bit-Windows,772
    - MobiLink-Server-Deployment unter 64-Bit-Unix,780
    - MobiLink-Server-Deployment unter 64-Bit-Windows,775
  - Tools
    - MobiLink-Profiler, Auswahlrechteck,203
  - ToString-Methode
    - DateTimeWithTimeZone-Klasse [MobiLink-Server-API für .NET],556
  - toString-Methode
    - TimestampWithTimeZone-Klasse [MobiLink-Server-Java-API],525
  - toTimestampWithTimeZone-Methode
    - TimestampWithTimeZone-Klasse [MobiLink-Server-Java-API],525
  - Transaktionale Uploads (*Siehe* Uploads auf Transaktionsebene)
    - MobiLink-Performance,145
  - Transaktionen
    - MobiLink,272
    - MobiLink, .NET-Synchronisationslogik,540
    - MobiLink, Java-Synchronisationslogik,466
  - Treiber
    - unterstützt von MobiLink,767
  - trusted\_certificates
    - MobiLink, mlsrv16-Option -x für HTTPS,98
    - MobiLink, mlsrv16-Option -x für TLS über TCP/IP,95
  - Type-Eigenschaft
    - LogMessage-Klasse [MobiLink-Server-API für .NET],596
- ## U
- u.
    - MobiLink, Präfix benutzerdefinierter Parameter,252

---

- Überlappungen
  - verteilen,113
- Übersicht, Fensterausschnitt
  - MobiLink-Profiler,203
- Überwachen
  - MobiLink-Performance,152
  - Synchronisationen in MobiLink,192
- Überwachung, MobiLink-Performance
  - Überblick,152
- ULRollbackPartialDownload-Funktion
  - neu startbare Downloads,136
- UltraLite
  - Deployment,785
- Umwandlung zwischen Zeichensätzen
  - MobiLink-Synchronisation,766
- Unabhängiges Teilen
  - Definition,113
- Unabhängiges Verteilen
  - MobiLink,114
- Unix
  - Deployment von dbmlsync,784
  - MobiLink-Server als Daemon,35
- Unternehmensdatenbanken
  - mit MobiLink synchronisieren,627
- UPDATE, Konflikte
  - MobiLink,124
- UpdateDataReader-Schnittstelle [MobiLink-Server-API für .NET]
  - Beschreibung,614
  - SetNewRowValues-Methode,615
  - SetOldRowValues-Methode,615
- UpdateResultSet-Schnittstelle [MobiLink-Server Java-API]
  - Beschreibung,526
  - setNewRowValues-Methode,527
  - setOldRowValues-Methode,527
- Upgrade von Anwendungen
  - mehrere MobiLink-Skriptversionen verwenden,254
- Upload, Zeilen
  - Skripten erstellen,262
- upload-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- Upload-Ereignisse
  - Info,262
  - MobiLink-Synchronisation,279
- Upload-only-Synchronisation
  - erforderliche Skripten,258
- Upload. Zeilen
  - MobiLink-Performance,148
- upload\_bytes-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- upload\_deadlocks-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- upload\_delete
  - Tabellenereignis,435
- upload\_deleted\_rows-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- upload\_errors-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- upload\_fetch
  - Konflikterkennung,126
  - Konflikterkennung, Überblick,125
  - Tabellenereignis,437
- upload\_fetch\_column\_conflict
  - Konflikterkennung,126
  - Konflikterkennung, Überblick,125
  - Tabellenereignis,439
- upload\_insert
  - Tabellenereignis,441
- upload\_inserted\_rows-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- upload\_new\_row\_insert
  - Tabellenereignis,443
- upload\_old\_row\_insert
  - Tabellenereignis,445
- upload\_statistics
  - Tabellenereignis,452
  - Verbindungsereignis,448
- upload\_update
  - Konflikterkennung,128
  - Konflikterkennung, Überblick,125
  - Tabellenereignis,458
  - verwenden,130
- upload\_updated\_rows-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- upload\_warnings-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- UploadData-Schnittstelle [MobiLink-Server-API für .NET]
  - Beschreibung,616
  - GetUploadedTableByName-Methode,617
  - GetUploadedTables-Methode,617
- UploadData-Schnittstelle [MobiLink-Server-Java-API]
  - Beschreibung,528
  - getUploadedTableByName-Methode,529

- getUploadedTables-Methode,529
- UploadedTableData-Schnittstelle [MobiLink-Server-API für .NET]
  - Beschreibung,618
  - GetDeletes-Methode,619
  - GetInserts-Methode,620
  - GetName-Methode,621
  - GetSchemaTable-Methode,621
  - GetUpdates-Methode,622
- UploadedTableData-Schnittstelle [MobiLink-Server-Java-API]
  - Beschreibung,530
  - getDeletes-Methode,532
  - getInserts-Methode,532
  - getMetaData method,533
  - getName-Methode,534
  - getUpdates-Methode,535
- Uploads
  - MobiLink-Skripten für den Upload von Zeilen,262
  - MobiLink-Transaktion,273
  - vorübergehend stoppen in MobiLink,133
- Uploadtransaktion
  - MobiLink,273
- User-Eigenschaft
  - LogMessage-Klasse [MobiLink-Server-API für .NET],596
- user-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- UTC TIMESTAMP
  - MobiLink,111
- UUIDs
  - MobiLink-Synchronisationsanwendung,118

## V

- Validieren
  - MobiLink, automatisch,219
  - MobiLink, benutzerdefiniert,221
- Validierung
  - MobiLink, dateibasierte Downloads,218
- Validierungsprüfungen
  - MobiLink, dateibasierte Downloads,218
- Value-Eigenschaft
  - DBParameter-Klasse [MobiLink-Server-API für .NET],574
- valueOf-Methode
  - TimestampWithTimeZone-Klasse [MobiLink-Server-Java-API],525

- VARBIT, Datentyp
  - Einschränkungen in konsolidierten ASE MobiLink-Datenbanken,11
- VARCHAR, Datentyp
  - MobiLink und andere DBMS-Systeme,6
- Variablen für die gesamte Sitzung
  - konsolidierte MobiLink-Datenbanken in Oracle,21
- VARRAY (Oracle)
  - Beispiel,24
  - Einschränkungen,25
  - in gespeicherten Prozeduren,24
- VB (*Siehe* Visual Basic)
- Verarbeiten, Konflikte
  - MobiLink, direkte Zeilenbehandlung,632
- Verarbeitung direkter Downloads
  - MobiLink, direkte Zeilenbehandlung,637
- Verarbeitung direkter Uploads
  - MobiLink, direkte Zeilenbehandlung,631
- Verbindung herstellen
  - MobiLink-Server (mlsrv16), Option -c,53
  - MobiLink-Server (mlsrv16), Option -x,93
- Verbindungen
  - MobiLink-Server (mlsrv16), Option -c,53
  - MobiLink-Server (mlsrv16), Option -x,93
- Verbindungseigenschaften
  - MobiLink-Server (mlsrv16), Option -x,93
- Verbindungsparameter
  - MobiLink-Server (mlsrv16), Option -x,93
- Verbindungsskripten
  - alphabetische Liste der MobiLink-Skripten,272
  - Definition,235
  - Hinzufügen von .NET-Skripten,653
  - Hinzufügen von Java-Skripten,655
  - Hinzufügen von SQL-Skripten,651
  - Info,235
  - Löschen von .NET-Skripten,653
  - Löschen von Java-Skripten,655
  - Löschen von SQL-Skripten,651
  - mit Sybase Central hinzufügen,259
  - ml\_global,256
- Verbindungszeichenfolgen
  - MobiLink-Server (mlsrv16),53
- Verschlüsseln, Kennwörter
  - Oracle-Treiberoption,768
- version, Option
  - MobiLink-Server (mlsrv16), Option -x für OE,99
- version, Protokolloption

---

- MobiLink-Server (mlsrv16), Option -x für HTTP,96
- MobiLink-Server (mlsrv16), Option -x für HTTPS,98
- version-Eigenschaft
  - MobiLink-Profiler, Synchronisationsstatistik,209
- Versionen
  - MobiLink-Synchronisationsskripten,254
  - Skriptversionen entfernen,257
  - Skriptversionen hinzufügen,256
- Verteilen
  - Definition,113
  - MobiLink, Info,113
  - MobiLink, unabhängig,114
- Verteilen mit Überlappungen
  - MobiLink,115
- Verteilen, Tabellen
  - Beispiel,113
- Verteilen, Zeilen
  - MobiLink, auf entfernte Datenbanken,113
- Verteilter Download
  - MobiLink, dateibasierte Downloads,213
- Verwaltete entfernte Datenbank
  - hinzufügen,167
- Viele-zu-Viele-Beziehungen
  - Synchronisation,115
  - verteilen,115
- Visual Basic
  - MobiLink-Synchronisationsskripten,536
  - Unterstützung in MobiLink .NET,536
- Visual Studio
  - MobiLink-Synchronisationsskripten,536
- Volle Synchronisation (*Siehe* Zwei-Wege-Synchronisation)

## W

- WARNING-Variable
  - LogMessage-Klasse [MobiLink-Server-Java-API],500
- WarningListener-Ereignis
  - ServerContext-Schnittstelle [MobiLink-Server-API für .NET],608
- Webdienste
  - mit MobiLink synchronisieren,627
- WebLogic
  - MobiLink und,627
- Webserver

- mit MobiLink synchronisieren,627
- Windows CE (*Siehe* Windows Mobile)
- Windows Mobile Device Center (*Siehe* Microsoft ActiveSync)
- Worker-Threads
  - automatische Anpassung,145
  - MobiLink,149
  - MobiLink-Performance,144

## X

- Xusage.txt
  - Standort,79

## Y

- Year-Eigenschaft
  - DateTimeWithTimeZone-Klasse [MobiLink-Server-API für .NET],560

## Z

- Zeichensatz, Hinweise
  - MobiLink,765
- Zeichensätze
  - MobiLink-Synchronisation,765
- Zeichensatzkonvertierung
  - durch ODBC-Treiber,767
  - während der MobiLink-Synchronisation,766
- Zeilen
  - in entfernten MobiLink-Datenbanken löschen,267
  - verteilen,113
- Zeilen auslesen
  - Synchronisationsmethoden,548
- Zeilen löschen
  - MobiLink, Methoden,132
- Zeilen mit dem Skript download\_delete\_cursor löschen
  - MobiLink,267
- Zeilenbehandlung in MobiLink (*Siehe* Direkte Zeilenbehandlung)
- Zeilenparameter
  - MobiLink-Skripten,237
- Zeitpunkt des letzten Downloads
  - MobiLink, Info,109
- Zeitstempel
  - MobiLink, Download,110
- Zeitstempel des letzten Downloads
  - handle\_DownloadData-Verbindungsereignis,368
  - MobiLink, Generierung,110

- MobiLink, Info,109
- modify\_last\_download\_timestamp,  
Verbindungsereignis,392
- modify\_next\_last\_download\_timestamp,  
Verbindungsereignis,395
- Zeitstempelbasierte Downloads
  - Info,107
- Zeitstempelbasierte Synchronisation
  - download\_cursor, Skript,109
  - Info,107
- Zeitumstellung
  - MobiLink,111
- Zentraldatenbanken
  - konsolidierte MobiLink-Datenbanken,1
- Zentrale Administration entfernter Datenbanken
  - bereitgestellte entfernte Aufgabe abbrechen,176
  - entfernte Aufgabe einleiten,177
  - entfernte Aufgabe reaktivieren,178
- Zentrale Administration von entfernten Aufgaben  
(*Siehe* zentrale Administration von entfernten  
Datenbanken)
- Zentrale Administration von entfernten Datenbanken
  - Agent stoppen,163
  - Agentendatenbank, Definition,156
  - Befehle,181
  - Befehlsdefinition,155
  - bereitgestellte Aufgabe, Definition,155
  - Datei herunterladen, Befehl,182
  - Datei hochladen, Befehl,185
  - Datei kopieren, Befehl,181
  - Datei löschen, Befehl,182
  - Datei umbenennen, Befehl,184
  - Datenbank erstellen, Befehl,181
  - Datenbank löschen, Befehl,183
  - Eingabeaufforderung, Befehl,184
  - Empfänger zu einer bereitgestellten Aufgabe  
hinzufügen,179
  - entfernte Aufgabe bearbeiten,174
  - entfernte Aufgabe bereitstellen,174
  - entfernte Aufgabe erstellen,173
  - entfernte Aufgabe exportieren,175
  - entfernte Aufgabe, Definition,155
  - entfernte Aufgaben,169
  - entfernte Datenbank, Definition,156
  - entferntes Schema, Namensdefinition,156
  - Info,153
  - Konzepte,154
  - mit bereitgestellten Aufgaben arbeiten,176
  - MobiLink-Agenten-ID, Definition,155
  - MobiLink-Agentendefinition,155
  - MobiLink-Projektdefinition,154
  - Programm ausführen, Befehl,185
  - serverinitiierte entfernte Aufgabe, Definition,156
  - Setup,156
  - SIRT,179
  - SQL ausführen, Befehl,183
  - Status,189
  - Statusinformationen, Definition,155
  - Sybase Central,164
  - Synchronisieren, Befehl,185
  - Systemprozeduren,189
- Zuordnung
  - MobiLink,Datentypen der konsolidierten  
Datenbank,714
- Zurücksetzen
  - MobiLink, letzte Downloadzeit,110
- Zusammengesetzte Schlüssel
  - MobiLink, eindeutige Primärschlüssel,118
- Zwei-Wege-Synchronisation
  - erforderliche Skripten,258
  - Info,116