Relay Server

Version 16.0

Februar 2013

Version 16.0
Februar 2013

Inhalt

Über diese Dokumentation

In diesem Handbuch wird beschrieben, wie Sie den Relay Server einrichten und verwenden und damit eine sichere Kommunikation zwischen mobilen Geräten sowie Afaria-, Mobile Office-, MobiLink-, SQL Anywhere-, Unwired Server- und Sybase Unwired Platform-Servern ermöglichen, die ihrerseits über einen Webserver kommunizieren.

Einführung in den Relay Server

Der Relay Server ermöglicht eine sichere Kommunikation mit Lastverteilung zwischen mobilen Geräten und Backend-Servern über einen Webserver. Zu den unterstützten Backend-Servern gehören Afaria, Mobile Office, MobiLink, SQL Anywhere, Unwired Server und Sybase Unwired Platform. Der Relay Server stellt Folgendes zur Verfügung:

- Eine gemeinsame Kommunikationsarchitektur 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 unterstützen, dass sie problemlos in vorhandene Firewallkonfigurationen und Unternehmensrichtlinien integriert werden kann.

Relay Server-Architektur

Ein Relay Server-Deployment umfasst Folgendes:

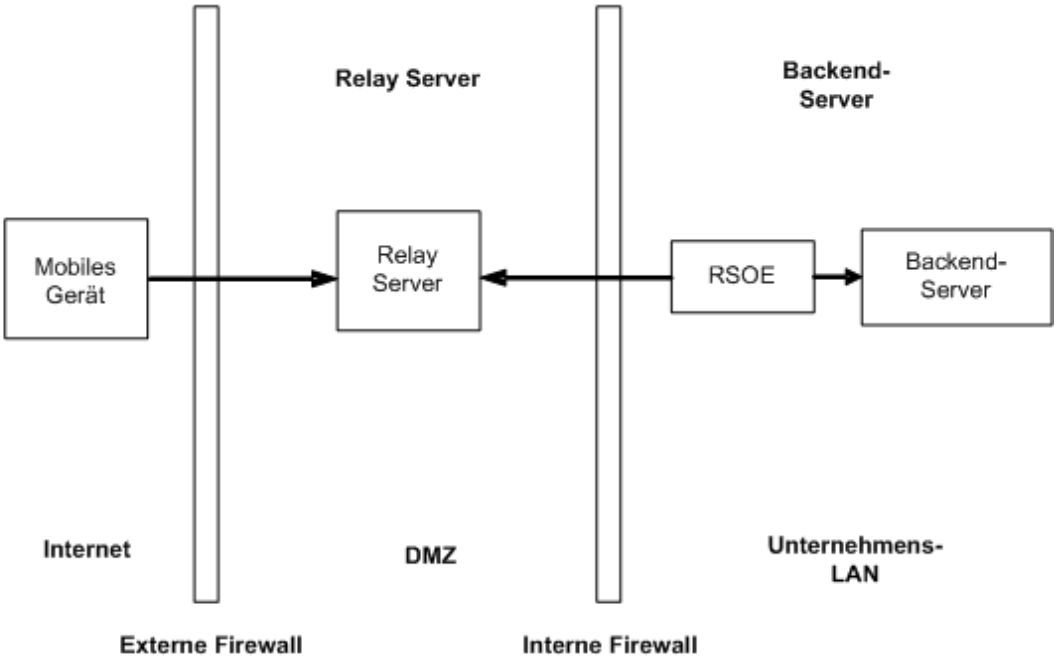
- Mobile Geräte, auf denen Clientanwendungen und Dienste ausgeführt werden, die mit Backend-Servern in einem Unternehmens-LAN kommunizieren müssen.
- Einen optionalen Lastverteiler, der Anforderungen von den mobilen Geräten an eine Gruppe von Relay Servern sendet.
- Einen oder mehrere Relay Server, die in der DMZ eines Unternehmens ausgeführt werden.
- Mindestens ein Backend-Server, der in einem Unternehmens-LAN ausgeführt wird und für die Bearbeitung von Clientanforderungen zuständig ist. Die folgenden Backend-Server werden für die Verwendung mit dem Relay Server unterstützt:
  - Afaria
  - Mobile Office
  - MobiLink
  - SQL Anywhere
  - Unwired Server
  - Sybase Unwired Platform

Hinweis
Der Relay Server wurde mit bestimmten Backend-Servern und Clients getestet, die für die Kommunikation über wohldefinierte HTTP-Anforderungen und -Antworten verwenden. In Deployments mit benutzerdefiniertem HTTP-Datenverkehr, einschließlich der Verwendung von SQL Anywhere als Webserver, muss der Datenverkehr gründlich getestet werden, um sicherzustellen, dass er mit dem Relay Server funktioniert.

Weitere Hinweise dazu, welche Backend-Server unterstützt werden, finden Sie in der Lizenzvereinbarung oder auf der Seite mit den SQL Anywhere-Komponenten nach Plattform. Siehe http://www.sybase.com/detail?id=1061806.

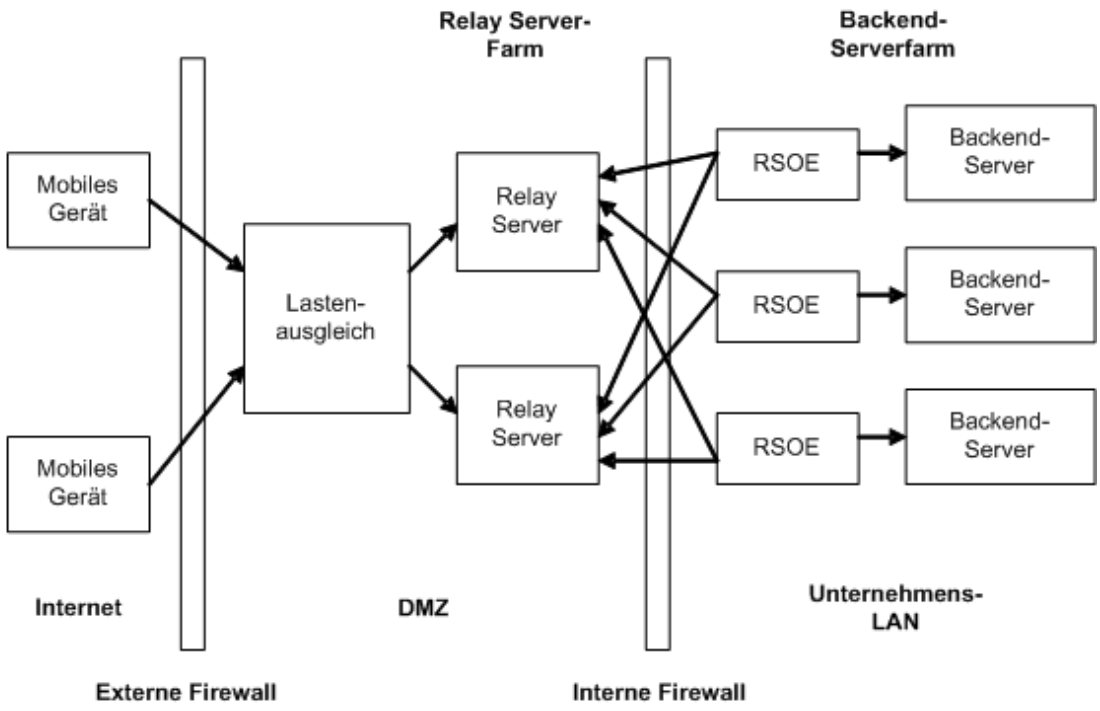
- Es gibt in der Regel nur einen Relay Server Outbound Enabler (RSOE) pro Backend-Server, aber es kann auch mehrere geben. Der Outbound Enabler verwaltet die gesamte Kommunikation zwischen einem Backend-Server und der Relay Server-Farm.

Das folgende Diagramm zeigt die Relay Server-Architektur mit einem Relay Server.



Hinweis
Weitere Hinweise dazu, welche Backend-Server unterstützt werden, finden Sie in der Lizenzvereinbarung oder auf der Seite mit den SQL Anywhere-Komponenten nach Plattform. Siehe http://www.sybase.com/detail?id=1061806.

Das folgende Diagramm zeigt die Relay Server-Architektur für ein komplexeres System mit einer Relay Server-Farm und einer Backend-Serverfarm.



Hinweis
Weitere Hinweise dazu, welche Backend-Server unterstützt werden, finden Sie in der Lizenzvereinbarung oder auf der Seite mit den SQL Anywhere-Komponenten nach Plattform. Siehe http://www.sybase.com/detail?id=1061806.

Der Relay Server verfügt über eine Reihe von Weberweiterungen, einen Hintergrundprozess für die Verwaltung von Zustandsinformationen und einen Webserver.

Da der Relay Server eine auf einem Webserver ausgeführte Weberweiterung ist, findet die gesamte Kommunikation über HTTP oder HTTPS statt. Die Verwendung von HTTP kann problemlos in vorhandene Firewallkonfigurationen und Richtlinien von Unternehmen integriert werden. Der Relay Server setzt voraus, dass die Verbindung vom Unternehmens-LAN zum Relay Server vom Unternehmens-LAN initiiert wird. Dadurch wird eine sicherere Deployment-Umgebung bereitgestellt, da keine eingehenden Verbindungen vom DMZ in das Unternehmens-LAN erforderlich sind.

Der Relay Server enthält zwei Weberweiterungen: eine Clienterweiterung und eine Servererweiterung. Die Clienterweiterung verarbeitet Clientanforderungen von Anwendungen, die auf mobilen Geräten ausgeführt werden. Die Servererweiterung verarbeitet Anforderungen, die vom Outbound Enabler im Auftrag eines Backend-Servers gesendet werden.

Shared Memory und Sicherheit

Der Relay Server verwendet Shared Memory zum Übertragen von HTTP-Anforderungen und -Antworten zwischen Client- und Server-Plug-In. In sicheren Deployments wird zwischen dem Client und dem Relay Server sowie zwischen dem Outbound Enabler und dem Relay Server HTTPS verwendet. In diesem Szenario entschlüsselt der Webserver das HTTPS in HTTP und anschließend verschlüsselt der Relay Server das HTTP auf dem Weg zum Outbound Enabler erneut. Das kurze Intervall, während dessen die Daten im Relay Server nicht verschlüsselt sind, wird manchmal als "Wireless Application Protocol-Gap" oder "WAP-Gap" bezeichnet.

Es gibt zwei Möglichkeiten, diese Daten gegen schädliche Prozesse auf demselben Computer zu sichern. Die primäre Methode besteht darin, Clients und Backend-Server zu verwenden, die Ende-zu-Ende-Verschlüsselung unterstützen. Die meisten MobiLink-Clients unterstützen Ende-zu-Ende-Verschlüsselung. Die zweite Methode, die für alle Relay Server minimal empfohlen wird, ist das Verstärken von Webserver und Betriebssystem (BS) für das Deployment in der DMZ mit den Standardverfahren, die für die einzelnen unterstützten Webserver und BS dokumentiert sind. Dieses Verstärken muss Schritte umfassen, mit denen die Anzahl der BS-Konten auf dem Webserver verringert wird. Im Idealfall wird beim Verstärken außerdem der Computer/die VM darauf beschränkt, nur den Relay Server und den Webserver auszuführen und sonst nichts. Ziel ist es, die Anzahl der Prozesse auf dem Computer auf ein Minimum zu beschränken und gleichzeitig durch Verstärken zu verhindern, dass schädliche Agenten schädliche Programme hinzufügen.

Die Relay Server-Farm

Eine Relay Server-Farm besteht aus einer Reihe von Relay Servern mit einem Frontend-Lastverteiler. Es ist möglich, eine Relay Server-Farm mit einem einzigen Relay Server einzurichten. In diesem Fall ist kein Lastverteiler erforderlich. Mobile Geräte können dann eine direkte Verbindung mit dem Relay Server herstellen.

Backend-Serverfarm

Eine Backend-Serverfarm ist eine Gruppe von homogenen Backend-Servern. Ein Client, der eine Anforderung über die Relay Server-Farm stellt, muss die Backend-Serverfarm angeben, an die die Anforderung gerichtet ist.

Lastverteiler

Der Lastverteiler richtet Anforderungen von den mobilen Geräten an einen in der Relay Server-Farm ausgeführten Relay Server. Der Lastverteiler ist nicht erforderlich, wenn nur ein Relay Server vorhanden ist.

Relay Server Outbound Enabler

Der Relay Server Outbound Enabler wird auf demselben Computer ausgeführt wie der Backend-Server. Seine vorrangige Funktion besteht darin, im Auftrag des Backend-Servers eine abgehenden Verbindung mit allen Relay Servern in der Relay Server-Farm zu initiieren. Es gibt in der Regel nur einen Relay Server Outbound Enabler (RSOE) pro Backend-Server, aber es kann auch mehrere geben.

Siehe auch

- Siehe „Outbound Enabler“ auf Seite 39.

Relay Server-Sicherheit

Der Relay Server verfügt über integrierte Sicherheitsfunktionen, nutzt aber auch die vom Webserver bereitgestellten Sicherheitsfunktionen. In Kombination mit dem Webserver bietet der Relay Server die folgenden Funktionen für eine sichere Kommunikation:

- Serverseitige Zertifikate
- Clientseitige Zertifikate
- Konfiguration von Backend-Servern und -Farm
- RSOE-MAC-Adressenfilterung und Token-Authentifizierung
- Clientseitige Verschlüsselungstechnologien (Verschlüsselung auf Protokollebene)

Serverseitige Zertifikate

Mithilfe eines serverseitigen Zertifikats kann ein Client bei der Kommunikation mit dem Relay Server überprüfen, ob es sich bei dem Webserver, auf dem der Relay Server ausgeführt wird, um einen vertrauenswürdigen Server handelt. Der Client überprüft das öffentliche Zertifikat des Webservers anhand des Stammzertifikats, das auf dem Client gespeichert ist. Wenn die Zertifikate überprüft werden, erfolgt ein Schlüsselaustausch zur Herstellung der verschlüsselten Verbindung.

Clientseitige Zertifikate

Mithilfe eines clientseitigen Zertifikats kann der Webserver überprüfen, ob es sich bei einem Client, der mit dem Relay Server kommuniziert, um einen vertrauenswürdigen Client handelt. Der Webserver überprüft das öffentliche Zertifikat des Clients anhand des Stammzertifikats, das im Zertifikat-Manager auf dem Webserver-Computer gespeichert ist. Wenn die Zertifikate überprüft werden, erfolgt ein Schlüsselaustausch zur Herstellung der verschlüsselten Verbindung.

Konfiguration von Backend-Servern und -Farm

Die Datei *rs.config* wird vom Relay Server zum Definieren der Peer-Liste von Relay Servern verwendet, wenn er in einer Farmumgebung ausgeführt wird, einschließlich der Konfigurationen von Backend-Farm und Backend-Servern. Jeder Relay Server in der Umgebung muss über eine Kopie der Datei *rs.config* verfügen.

Die Konfiguration von Backend-Farm und Backend-Servern stellt sicher, dass der Relay Server nur mit Computern kommuniziert, mit denen er konfiguriert wurde. Jeder Versuch einer Kommunikation mit Computern, für die der Relay Server nicht konfiguriert wurde, wird abgelehnt.

Die Backend-Farm kann so konfiguriert werden, dass sie die Stufe der Kommunikationssicherheit angibt, wenn sie Anforderungen von den Clients und vom RSOE akzeptiert. Es gibt die Optionen client_security und backend_security, mit denen die Backend-Farm angeben kann, welcher Kommunikationstyp zulässig ist. Diese Option wird wie folgt festgelegt:

client_security=on|off **On** bedeutet, dass der Client sich über HTTPS verbinden muss. **Off** bedeutet, dass der Client sich über HTTP verbinden muss. Diese Einstellung ist optional. Wenn kein Wert angegeben ist, kann sich der Client über HTTP oder HTTPS verbinden.

backend_security=on|off **On** bedeutet, dass der RSOE sich über HTTPS verbinden muss. **Off** bedeutet, dass der RSOE sich über HTTP verbinden muss. Diese Einstellung ist optional. Wenn kein Wert angegeben ist, kann sich der RSOE über HTTP oder HTTPS verbinden.

RSOE-MAC-Adressenfilterung und Token-Authentifizierung

Der RSOE stellt die Verbindung zwischen dem Backend-Server und dem Relay Server in drei Phasen her: 1) Startphase, 2) Bereitschaftsphase, 3) Arbeitsphase.

Im Backend-Server-Abschnitt der Datei *rs.config* wird jeder Server in der Backend-Farm mit einer ID und dem zugeordneten Farmnamen konfiguriert. Die ID entspricht dem Servernamen. Der Relay Server hat die Möglichkeit, die MAC-Adresse des Computers zu überprüfen, auf dem der RSOE ausgeführt wird, um zu gewährleisten, dass der Server, der eine Kommunikation von innerhalb der internen Firewall aus versucht, vertrauenswürdig ist und eine Verbindung mit dem Relay Server herstellen darf. Die MAC-Eigenschaft ist die MAC-Adresse des Netzwerkadapters, der vom RSOE verwendet wird. Die Adresse wird im Format IEEE 802 MAC-48 angegeben.

Der Backend-Server-Abschnitt ermöglicht außerdem die Konfiguration eines Sicherheitstokens, der vom Relay Server zum Authentifizieren der Backend-Serververbindung verwendet wird. Der Token muss beim Start des RSOE bereitgestellt werden, wenn die Verbindung mit dem Relay Server hergestellt wird.

MobiLink-Sicherheit

Der MobiLink Client verwendet HTTP oder HTTPS, um mit dem Relay Server zu kommunizieren. Bei HTTPS-Kommunikation werden Daten während des Austauschs zwischen dem Client und dem Backend-Server vorübergehend entschlüsselt und wieder verschlüsselt. Dies wird als "WAP-Gap" bezeichnet. Um eine vollständig sichere Kommunikation über das WAP-Gap sicherzustellen, sollten Sie die Ende-zu-Ende-Verschlüsselungsfunktion von MobiLink verwenden, um die über den Relay Server übertragenen Daten zusätzlich zu schützen. Die Ende-zu-Ende-Verschlüsselungsfunktion von MobiLink bietet eine Verschlüsselung auf Protokollebene zwischen den MobiLink-, und UltraLite-Clients und dem MobiLink-Server. MobiLink und UltraLite verwenden RSA-Verschlüsselung. TLS-Sicherheit kann in Kombination mit Ende-zu-Ende-Verschlüsselung verwendet werden.

HTTP-Listening-Port des Outbound Enablers

Aus Sicherheitsgründen sollte bei Verwendung eines Standalone-Outbound Enablers der HTTP-Listening-Port des Backend-Servers explizit nur an die Loopback-IP-Adresse (127.0.0.1) gebunden werden.

Siehe auch

- „Relay Server-Konfigurationsdatei“ auf Seite 29
- Microsoft IIS http://www.sybase.com/detail?id=1059277
- Apache http://www.sybase.com/detail?id=1065869
- „Ende-zu-Ende-Verschlüsselung“ [*SQL Anywhere Server - Datenbankadministration*]

Affinität

Einige Clients und Backend-Server erfordern, dass bestimmte Sequenzen multipler HTTP-Anforderungen an denselben Backend-Server gehen oder sogar auf denselben TCP Socket, der mit diesem Backend-

Server verbunden ist. In der Relay Server-Terminologie ist "Affinität" die Verknüpfung zwischen einem Client und einem Backend-Server über mehrere HTTP-Anforderungen. Wenn ein Client NICHT mehrere Anforderungen für denselben Backend-Server verlangt, ist keine Affinität erforderlich und Sie benötigen keine weiteren Kenntnisse darüber.

Der Relay Server fügt Affinitätsinformationen zu HTTP-Anforderungen hinzu und respektiert Affinitätsinformationen, die in HTTP-Anforderungen gesendet wurden. Die Affinitätsinformationen werden über HTTP-Cookies und/oder Header gesendet. Clients, die mehrere Anforderungen über den Relay Server an denselben Backend-Server senden wollen, müssen die Affinitätsinformationen zurücksenden, die sie mit jeder HTTP-Antwort erhalten, indem sie sie in die nächste HTTP-Anforderung einfügen. Backend-Server müssen nichts tun, um an der Relay Server-Affinität teilzunehmen. Wenn eine nicht beständige HTTP-Verbindung für eine Sequenz von mehreren Anforderungen für denselben Backend-Server verwendet wird, erstellt jede Anforderung eine neue TCP Socket-Verbindung. Wenn es mehrere Backend-Server in einer Farm gibt, muss der Client die Affinitätsinformationen zwischen den Anforderungen aufrechterhalten. Die Affinitätsinformationen teilen dem Relay Server mit, dass die Anforderungen verbunden sind und an den gleichen Backend-Server gehen.

Wenn eine beständige HTTP-Verbindung verwendet wird, muss jede Anforderung auf dem gleichen TCP-Socket erfolgen. Eine beständige HTTP-Verbindung ist eine häufiger verwendete Möglichkeit, weil sie den Performance-Vorteil bietet, den Overhead durch den Socket (und möglicherweise TLS-Verbindungen) zu reduzieren. Wenn eine beständige HTTP-Verbindung verwendet wird, muss der Client jedoch weiterhin die Affinitätsinformationen zwischen Anforderungen aufrechterhalten, weil der Relay Server weiterhin verlangt, dass die Beständigkeit zwischen dem Outbound Enabler und dem Backend-Server erhalten bleibt.

Clientseitige Probleme mit der Verwaltung von Affinität können zu fehlgeschlagenen Anforderungen und anderem ungewöhnlichem Verhalten führen. Wenn Sie ein Affinitätsproblem mit Ihrer Clientanwendung vermuten, wenden Sie sich an den Lieferanten Ihrer Clientanwendung. Wenn der Relay Server-Administrator mit der Clientanwendung vertraut ist und gründliche Kenntnisse des gewählten Mechanismus für die Verwaltung der Affinität für alle betroffenen Clients der Backend-Farm hat, kann er nicht benutzte Affinitätseinfügungen mit der **active_cookie**- oder der **active_header**-Eigenschaft im Backend-Farm-Abschnitt der Relay Server-Konfigurationsdatei gezielt abschalten. Das explizite Abschalten der **renew_overlapped_cookie**-Eigenschaft wird NICHT empfohlen. Dies könnte zu Verbindungsproblemen führen, wenn der Client Standard HTTP-Cookie-Reflektion verwendet, um Affinitätsinformationen ohne Isolation zwischen gleichzeitigen Sitzungen aufrechtzuerhalten.

Siehe auch

Eigenschaften im Backend-Farm-Abschnitt auf Seite 30

Relay Server-Statusseite

Die Relay Server-Statusseite enthält die folgenden Informationen:

- Relay Server-Version und Buildnummer
- Hostname
- Servicestartzeit in UTC
- Statuserfassungszeit in UTC
- Statusaktualisierungsintervall oder Hinweis, dass manuelle Aktualisierung erwartet wird
- Gesamtverfügbarkeit
- Liste der nicht verfügbaren Backend-Farmen
- Liste der teilweise verfügbaren Backend-Farmen
- Liste der verfügbaren Backend-Farmen

Leere Statusseite
Eine leere Statusseite kann darauf hinweisen, dass es ein Konfigurationsproblem gibt, das die Erweiterung von der Generierung der Seite abhält. Stellen Sie sicher, dass der rshost-Prozess gestartet wurde und der Webserver Worker-Thread über die Berechtigung für den Zugriff auf den gemeinsam genutzten Speicher verfügt, der vom rshost-Prozess erstellt wurde. Um Konfigurationsfehler zu beheben, starten Sie rshost manuell und prüfen die Logdatei.

Die Frequenz der automatischen Aktualisierung des Backend-Servers wird mit dem ias-rs-status-refresh-sec-Abfrageparameter des Outbound Enablers eingestellt.

Aktiviert regelmäßige Backend-Server-Statusanforderungen. Der Parameter status_url wird im Format `status_url=/your-status-url ias-rs-status-refresh-sec=n` angegeben. Mit dem Wert 0 wird die automatische Aktualisierung ausgeschaltet.

Das folgende Beispiel zeigt, wie status_url mit -cs angegeben und auto-refresh auf 20 Sekunden gesetzt wird.

```
-cs "host=localhost;port=80;status_url=/getstatus/ ias-rs-status-refresh-
sec=20"
```

Verwenden Sie die Option -d, um die Frequenz der Backend-Serverstatusanforderungen anzugeben.

Die URL für eine typische Statusseite lautet *http://MyHost:80/ias_relay_server/server/rs_server.dll* für Microsoft IIS und *http://MyHost:80/srv/iarelayserver* für Apache.

Zusätzliche Statusseiten

Die folgenden Statusseiten sind ebenfalls verfügbar:

Statusseite	Benutzer oder Speicherort	URL-Formatbeispiele
Detaillierter Status	Relay Server-Administrator	*http://<host>/ias_relay_server/admin/rs_admin.dll*

Statusseite	Benutzer oder Speicherort	URL-Formatbeispiele
Gesamtstatus	Entfernter Benutzer	*http:// <host>/ias_relay_server/client/rs_client.dll*
Status der Backend-Farm	Entfernter Benutzer	*http:// <host>/ias_relay_server/client/rs_client.dll?ias-rs-farm=<App-farm>*
Status des Backend-Servers	Entfernter Benutzer	*http:// <host>/ias_relay_server/client/rs_client.dll?ias-rs-farm=<App-farm>&ias-rs-server=<App-server>*
Gesamtstatus	Backend-Server-Administrator	*http:// <host>/ias_relay_server/server/rs_server.dll*
Status der Backend-Farm	Backend-Server-Administrator	*http:// <host>/ias_relay_server/server/rs_server.dll?ias-rs-farm=<App-farm>*
Status des Backend-Servers	Backend-Server-Administrator	*http:// <host>/ias_relay_server/server/rs_server.dll?ias-rs-farm=<App-farm>&ias-rs-server=<App-server>*

Relay Server-Deployment

Deployment der Relay Server-Komponenten für Microsoft IIS 6.0 unter Windows Server 2003

Vor dem Ausführen des Relay Servers mit IIS 6.0 müssen Sie Relay Server-Dateien auf jedem Computer in der Relay Server-Farm bereitstellen.

Voraussetzungen

Die Relay Server-Komponenten werden als Teil von SQL Anywhere 16 installiert. Der Installationsprozess stellt automatisch alle erforderlichen Dateien auf dem Computer bereit, auf dem der Relay Server ausgeführt werden soll.

Standardmäßig werden alle Dateien im Verzeichnis *%SQLANY16%* installiert und basieren auf dem Bitwert des Computers:

- *%SQLANY16%\Bin64* wird für DLLs und ausführbare Dateien für Administrationszwecke benutzt.
- *%SQLANY16%\RelayServer\IIS\Bin64* wird für Relay Server-spezifische Dateien im entsprechenden Ordner verwendet (z.B. *Admin*, *Client*, *Monitor* oder *Server*). Der Ordner *Server* enthält die Dateien *rshost.exe* und *rs.config*.

Kontext und Bemerkungen

Interaktive Schnellsetup-Funktion
Eine interaktive Schnellsetup-Funktion *rs-setup.bat* wird als Alternative zu dieser Prozedur bereitgestellt. *rs-setup.bat* befindet sich im Verzeichnis *%SQLANY16%/RelayServer/IIS/quicksetup_iis6* und umfasst die folgenden Aufgaben:

1. Erstellt eine Demoanwendung.
2. Generiert eine Kurzreferenz.

Der Relay Server für Windows umfasst die folgenden Programmdateien:

- *rs_client.dll*
- *rs_server.dll*
- *rs_monitor.dll*
- *rshost.exe*
- *dblgen16.dll*
- *dbsvc.exe*
- *dbfhide.exe*
- *dbtool16.dll*
- *dblib16.dll*
- *dbicu16.dll*
- *dbicudt16.dll*
- *dbsupport.exe*
- *dbghelp.dll*

Hinweise dazu, welche Versionen von IIS unterstützt werden, finden Sie unter http://www.sybase.com/detail?id=1061806.

Setupskripten für Relay Server für IIS finden Sie im Verzeichnis *%SQLANY16%\RelayServer\IIS*.

Aufgabe

1. Erstellen Sie ein virtuelles Verzeichnis namens *rs* unter "Standardwebsite" in Microsoft IIS Manager für die Verwendung durch den Relay Server. Der physische Speicherort des virtuellen Verzeichnisses ist *% SQLANY16% \RelayServer\IIS\Bin64*.

2. Erstellen Sie die Relay Server-Konfigurationsdatei *rs.config* unter Beachtung folgender Richtlinien:

   - Die Datei sollte vier Abschnitte haben:
     - Optionen-Abschnitt
     - Relay Server-Abschnitt
     - Backend-Farm-Abschnitt
     - Backend-Server-Abschnitt
   - Jeder Abschnitt beginnt mit einem Abschnitts-Tag, das ein Schlüsselwort, das den Abschnittsnamen angibt, in eckigen Klammern enthält.
   - Fügen Sie den einzelnen Abschnitten die entsprechenden Eigenschaften hinzu. Eine Eigenschaft wird durch den Eigenschaftsnamen links von einem Gleichheitszeichen und den dazugehörigen Wert rechts vom Gleichheitszeichen festgelegt. Beispiel: *propertyname = value*.
   - Die Konfigurationsdatei darf nur 7-Bit-ASCII-Zeichen enthalten.

3. Erstellen Sie einen Anwendungspool:

   a. Starten Sie die Microsoft IIS Manager-Konsole.

   b. Klicken Sie mit der rechten Maustaste auf **Anwendungspools** und erstellen Sie einen neuen Anwendungspool, z.B. RS_POOL.

   c. Bearbeiten Sie die Eigenschaften für den Anwendungspool, den Sie erstellt haben.

i. Klicken Sie auf die Registerkarte **Wiederverwendung** und deaktivieren Sie alle Wiederverwendungsoptionen.

ii. Klicken Sie auf die Registerkarte **Performance** und führen Sie Folgendes durch:

A. Deaktivieren Sie **Arbeitsprozesse im Leerlauf herunterfahren nach (Minuten)**.

B. Setzen Sie die Anzahl der Worker-Prozesse auf die Gesamtzahl der Prozessorkerne. Sie können diese Anzahl weiter an Ihre Nutzungs- und Performancepräferenzen anpassen. Weitere Hinweise finden Sie in den Microsoft IIS-Performancehinweisen zur Webgartengröße.

4. Setzen Sie den Verbindungs-Timeout der Standardwebsite auf mindestens 60 Sekunden. Standardmäßig ist dieser Wert 120 Sekunden, was normalerweise ausreicht.

5. Bearbeiten Sie die Eigenschaften von "rs" und aktivieren Sie die Relay Server-Weberweiterungen in der IIS-Manager-Konsole:

a. Klicken Sie auf die Registerkarte **Verzeichnis** und führen Sie folgende Schritte aus:

i. Definieren Sie die Ausführungsberechtigungen für **Skripts und ausführbare Dateien**

ii. Klicken Sie unter **Anwendungseinstellungen** auf **Erstellen**. Wählen Sie den Anwendungspool, den Sie in Schritt 3 erstellt haben, als zugeordneten Anwendungspool aus.

b. Klicken Sie auf die Registerkarte **Verzeichnissicherheit** und führen Sie folgende Schritte aus:

i. Klicken Sie unter **Authentifizierung und Zugriffsteuerung** auf **Bearbeiten**.

ii. Aktivieren Sie den anonymen Zugriff und geben Sie den Benutzernamen und das Kennwort für ein Konto ein, das Mitglieds der Administratorengruppe ist.

Alternativ können Sie auch die Einstellung als integrierter Benutzer **IUSR_ %computername%** beibehalten und den folgenden Befehl ausführen, um Berechtigungen für den Zugriff auf die Microsoft IIS-Metabasis zu erteilen.

```
C:\Windows\Microsoft.Net\Framework\<Version>\aspnet_regiis.exe -ga
IUSR_%computername%
```

c. Fügen Sie unter **Webservererweiterungen** im Microsoft IIS Manager die Einträge *rs_server.dll*, *rs_client.dll* und *rs_monitor.dll* als neue Webdiensterweiterungen hinzu. Der Erweiterungsname sollte ISAPI sein und in den DLLs muss der Erweiterungsstatus "Zugelassen" festgelegt werden.

6. Nehmen Sie das Deployment der Relay Server-Konfigurationsdatei vor, indem Sie eine Relay Server-Konfigurationsdatei erstellen und in das Verzeichnis *%SQLANY16%\RelayServer\IIS\BinXX\server* kopieren.

7. Stellen Sie optimale Performance mithilfe der Performancetipps sicher.

8. Definieren Sie einen Dienst, der automatisch den Relay Server-Status-Manager startet, wenn der Computer hochgefahren wird. Benutzen Sie dazu eine Befehlszeile wie im folgenden Beispiel:

```
dbsvc -as -t rshost -w RelayServer "%SQLANY16%\RelayServer\IIS\BinXX
\Server\rshost.exe" -q -qc -f "%SQLANY16%\RelayServer\IIS\BinXX\Server
\rs.config" -o "c:\temp\ias_relay_server.log"
```

Hinweis
Es wird empfohlen, dass Sie den Status-Manager als Dienst starten. Er kann jedoch auch automatisch vom Relay Server gestartet werden.

9. Aktualisieren Sie die Relay Server-Konfiguration für Microsoft IIS 6.0 unter Windows:
   a. Kopieren Sie für jeden Computer, der zur aktualisierten Relay Server-Farm gehört, die aktualisierte Konfigurationsdatei in das Verzeichnis *%SQLANY16%\RelayServer\IIS\BinXX\Server*, das sich unter dem Stammverzeichnis der Relay Server-Website befindet.
   b. Führen Sie im Verzeichnis *%SQLANY16%\RelayServer\IIS\BinXX\Server* die folgende Anweisung in der Befehlszeile aus, um die Aktualisierung der Konfiguration zu übernehmen:

      ```
      rshost -u -f rs.config
      ```

   c. Wiederholen Sie die oben beschriebenen Schritte für jeden Computer in der zu aktualisierenden Relay Server-Farm.

Hinweis
Nachdem Sie den Relay Server mit IIS konfiguriert haben, sollten Sie entweder den IIS-Server oder den Computer neu starten.

Ergebnisse

Die Relay Server-Konfigurationsdatei wird auf allen Computern in der Relay Server-Farm bereitgestellt.

Siehe auch

- „Relay Server-Konfigurationsdatei“ auf Seite 29
- „Performancetipps“ auf Seite 14
- „Relay Server-Status-Manager“ auf Seite 25
- „Relay Server-Status-Manager als Dienst“ auf Seite 25
- „Befehlszeilensyntax des Relay Server Status-Managers (rshost)“ auf Seite 26
- Dienstprogramm zum Verschleiern von Dateien (dbfhide) auf Seite 47

Performancetipps

Beachten Sie folgende Punkte, wenn Sie das Deployment des Relay Servers für Microsoft IIS unter Windows durchführen:

- Die Relay Server-Weberweiterung basiert nicht auf ASP.NET. Durch Entfernen des ASP.NET-ISAPI-Filters können Sie eine bessere Performance erzielen. Der Filter wird in einer Standardinstallation von Microsoft IIS standardmäßig aktiviert. Um den Filter zu deaktivieren, führen Sie folgende Schritte aus:

  1. Starten Sie die Microsoft IIS Manager-Konsole.
  2. Bearbeiten Sie die Eigenschaften der **Standardwebsite**.
  3. Entfernen Sie auf der Registerkarte **ISAPI-Filter** den ASP.NET-Filter.

- Um eine bessere Performance zu erzielen, können Sie das Microsoft IIS-Zugriffslog deaktivieren. Um das Zugriffslog zu deaktivieren, führen Sie folgende Schritte aus:

  1. Starten Sie die Microsoft IIS Manager-Konsole.
  2. Bearbeiten Sie die Eigenschaften des Verzeichnisses *rs* unter **Standardwebsite**.
  3. Heben Sie auf der Registerkarte **Verzeichnis** die Auswahl der Option **Besuche protokollieren** auf.

- In einer Produktionsumgebung kann die Relay Server-Ausführlichkeit über die Relay Server-Konfigurationsdatei auf 0 gesetzt werden. Dies ergibt eine bessere Performance unter hoher Belastung.

- Der Relay Server erzwingt keine Einschränkungen der Webgartengröße. Ein Worker-Prozess kann Anforderungen von allen Outbound Enablern sowie von allen Clients bearbeiten. Die Anzahl der Threads, die dabei erstellt werden können, wird jedoch durch den Prozess-Heap-Speicherplatz begrenzt, der noch für die Thread-Erstellung zur Verfügung steht. Der von Microsoft IIS erstellte Thread hat eine Stack-Größe von 256 kB. Wenn Ihr Computer über ausreichend Ressourcen verfügt und Sie vermuten, dass der Mehrbenutzerbetrieb an eine Grenze stößt, wenn der Server mit Tausenden von gleichzeitigen Anforderungen belastet wird, experimentieren Sie mit einer höheren Anzahl von Prozessen.

Deployment der Relay Server-Komponenten für Microsoft IIS 7.0 unter Windows Server 7.0 oder 8.0

Vor dem Ausführen des Relay Servers mit IIS 7.0, 7,5 oder 8.0 müssen Sie die Relay Server-Dateien auf jedem Computer in der Relay Server-Farm bereitstellen.

Voraussetzungen

Die Microsoft IIS-Funktion "ISAPI-Erweiterungen" ist installiert.

Die Relay Server-Komponenten werden unter Verwendung der SQL Anywhere-Installation installiert. Standardmäßig werden alle Dateien in *% SQLANY16%* installiert:

- *%SQLANY16%\Bin64* wird für DLLs und ausführbare Dateien für Administrationszwecke benutzt.
- *%SQLANY16%\RelayServer\IIS\Bin64* wird für Relay Server-spezifische Dateien im entsprechenden Ordner verwendet (z.B. *Admin*, *Client*, *Monitor* oder *Server*). Der Ordner *Server* enthält die Dateien *rshost.exe* und *rs.config*.

Kontext und Bemerkungen

Interaktive Schnellsetup-Funktion
Eine interaktive Schnellsetup-Funktion *rs-setup.bat* wird als Alternative zu dieser Prozedur bereitgestellt. Die Datei *rs-setup.bat* befindet sich im Verzeichnis *%SQLANY16%/RelayServer/IIS/quicksetup_iis7or8* und führt die folgenden Aufgaben aus:

1. Installiert IIS7 oder IIS8 und aktiviert die erforderlichen IIS7- oder IIS8-Funktionen
2. Konfiguriert IIS7 oder IIS8 für den Relay Server.
3. Erstellt eine Demoanwendung.
4. Generiert eine Kurzreferenz.

Der Relay Server für Windows umfasst die folgenden Programmdateien:

- *rs_client.dll*
- *rs_server.dll*
- *rs_monitor.dll*
- *rshost.exe*
- *dblgen16.dll*
- *dbsvc.exe*
- *dbfhide.exe*
- *dbtool16.dll*
- *dblib16.dll*
- *dbicu16.dll*
- *dbicudt16.dll*
- *dbsupport.exe*
- *dbghelp.dll*

Hinweise dazu, welche Versionen von IIS unterstützt werden, finden Sie unter http://www.sybase.com/detail?id=1061806.

IIS7/8-Setupskripten für den Relay Server finden Sie im Verzeichnis *%SQLANY16%\RelayServer\IIS\quicksetup_iis7or8*.

Aufgabe

1. Sichern Sie die IIS-Konfigurationsdatei *applicationHost.config* aus dem Verzeichnis *c:\Windows\System32\inetsrv\config*.
2. Um einen Anwendungspool für den Relay Server hinzuzufügen, bearbeiten Sie die Datei *applicationHost.config* und fügen Sie im Abschnitt **<system.applicationHost>** » **<applicationPools>** den folgenden Code hinzu.

```
<add name="RelayServer" queueLength="65535" managedRuntimeVersion=""
managedPipelineMode="Integrated">
   <processModel identityType="LocalSystem" idleTimeout="00:00:00"
```

```
maxProcesses="20" pingingEnabled="false"
         pingInterval="00:00:30" pingResponseTime="00:01:30" />
    <recycling disallowOverlappingRotation="true">
      <periodicRestart time="00:00:00">
        <schedule>
          <clear />
        </schedule>
      </periodicRestart>
    </recycling>
    <failure rapidFailProtection="false" />
    <cpu resetInterval="00:00:00" />
</add>
```

Hinweis
Während der restlichen Beschreibung der durchzuführenden Schritte wird *%SQLANY16%\RelayServer\IIS\BinXX* als %rs_dir% in der Datei *applicationHost.config* bezeichnet. Die Erweiterung der Umgebungsvariablen wird aber nicht in jedem Abschnitt der IIS-Konfigurationsdatei unterstützt, sodass die Variable %rs_dir% vollständig erweitert werden muss, wenn Sie sie in die Datei applicationHost.config einfügen.

3. Um die Relay Server-Anwendung zur Standardwebsite hinzuzufügen, bearbeiten Sie die Datei *applicationHost.config* und fügen im Abschnitt **<system.applicationHost>** » **<sites>** » **<site name="Default Web Site">** den folgenden Code hinzu.

   ```
   <application path="/rs" applicationPool="RelayServer">
     <virtualDirectory path="/" physicalPath="%rs_dir%"/>
   </application>
   ```

4. Um die ISAPI-Erweiterungen für den Relay Server hinzuzufügen, bearbeiten Sie die Datei *applicationHost.config* und fügen Sie im Abschnitt **<system.webServer>** » **<security>** » **<isapiCgiRestriction>** den folgenden Code hinzu.

   ```
   <add path="%rs_dir%\Admin\rs_admin.dll" allowed="true" />
   <add path="%rs_dir%\Client\rs_client.dll" allowed="true" />
   <add path="%rs_dir%\Monitor\rs_monitor.dll" allowed="true" />
   <add path="%rs_dir%\Server\rs_server.dll" allowed="true" />
   ```

5. Um den Zugriff auf den Relay Server zu aktivieren, bearbeiten Sie die Datei *applicationHost.config* und fügen Sie im Abschnitt **<configuration>** den folgenden Code hinzu.

   ```
   <location path="Default Web Site/rs">
     <system.webServer>
       <security>
         <authentication>
           <anonymousAuthentication userName="" />
         </authentication>
         <requestFiltering>
           <requestLimits maxAllowedContentLength="2147483647" />
         </requestFiltering>
       </security>
       <handlers accessPolicy="Execute, Script" />
     </system.webServer>
   </location>
   ```

Hinweis
Der Relay Server wird basierend auf diesen Anweisungen für einen anonymen Zugriff eingerichtet. Basierend auf den Geschäftsanforderungen muss die Sicherheit für IIS und für den Relay Server ordnungsgemäß konfiguriert werden.

6. Um den HTTPS-Zugriff auf die Relay Server-Administrationserweiterung aus Gründen der Sicherheit zu erzwingen, bearbeiten Sie die Datei *applicationHost.config* und fügen Sie den folgenden Code dem Abschnitt <configuration> hinzu.

```
<location path="Default Web Site/rs/Admin">
  <system.webServer>
    <security>
      <access sslFlags="Ssl" />
    </security>
  </system.webServer>
</location>
```

7. Speichern Sie diese Änderungen in der Datei *applicationHost.config*.

8. Setzen Sie den Verbindungs-Timeout der Standardwebsite auf mindestens 60 Sekunden. Standardmäßig ist dieser Wert 120 Sekunden, was normalerweise ausreicht.

9. Erstellen Sie die Relay Server-Konfigurationsdatei *rs.config* unter Beachtung folgender Richtlinien:

   - Die Datei sollte vier Abschnitte haben:
     - Optionen-Abschnitt
     - Relay Server-Abschnitt
     - Backend-Farm-Abschnitt
     - Backend-Server-Abschnitt
   - Jeder Abschnitt beginnt mit einem Abschnitts-Tag, das ein Schlüsselwort, das den Abschnittsnamen angibt, in eckigen Klammern enthält.
   - Fügen Sie den einzelnen Abschnitten die entsprechenden Eigenschaften hinzu. Eine Eigenschaft wird durch den Eigenschaftsnamen links von einem Gleichheitszeichen und den dazugehörigen Wert rechts vom Gleichheitszeichen festgelegt. Beispiel: *propertyname* = *value*.
   - Die Konfigurationsdatei darf nur 7-Bit-ASCII-Zeichen enthalten.

10. Kopieren Sie die Datei *rs.config* in das Verzeichnis *%SQLANY16%\RelayServer\IIS\BinXX\Server*.

11. Stellen Sie optimale Performance mithilfe der Performancetipps sicher.

12. Definieren Sie einen Dienst, der automatisch den Relay Server-Status-Manager startet, wenn der Computer hochgefahren wird. Benutzen Sie dazu eine Befehlszeile wie im folgenden Beispiel:

```
dbsvc -a <administrator> -p <password> -t rshost -w RelayServer "%rs_dir%
\Server\rshost.exe" -q -qc -f "%rs_dir%\Server\rs.config" -o "c:\temp
\ias_relay_server.log"
```

Hinweis
Es wird empfohlen, dass Sie den Status-Manager als Dienst starten. Er kann jedoch auch automatisch vom Relay Server gestartet werden.

13. Aktualisieren Sie die Relay Server-Konfiguration für Microsoft IIS unter Windows:
    a. Kopieren Sie für jeden Computer, der zur aktualisierten Relay Server-Farm gehört, die aktualisierte Konfigurationsdatei in das Verzeichnis *%SQLANY16%\RelayServer\IIS\BinXX\Server*, das sich unter dem Stammverzeichnis der Relay Server-Website befindet.
    b. Führen Sie im Verzeichnis *%SQLANY16%\RelayServer\IIS\BinXX\Server* die folgende Anweisung in der Befehlszeile aus, um die Aktualisierung der Konfiguration zu übernehmen:

       ```
       rshost -u -f rs.config
       ```

    c. Wiederholen Sie die oben beschriebenen Schritte für jeden Computer in der zu aktualisierenden Relay Server-Farm.

Ergebnisse

Die Relay Server-Konfigurationsdatei wird auf allen Computern in der Relay Server-Farm bereitgestellt.

Siehe auch

- „Relay Server-Konfigurationsdatei“ auf Seite 29
- „Performancetipps“ auf Seite 14
- „Relay Server-Status-Manager“ auf Seite 25
- „Relay Server-Status-Manager als Dienst“ auf Seite 25
- „Befehlszeilensyntax des Relay Server Status-Managers (rshost)“ auf Seite 26
- Dienstprogramm zum Verschleiern von Dateien (dbfhide) auf Seite 47

Deployment der Relay Server-Komponenten auf Apache unter Linux

Vor dem Ausführen des Relay Servers mit Apache müssen Sie die Relay Server-Dateien auf jedem Computer in der Relay Server-Farm bereitstellen.

Voraussetzungen

Die Relay Server-Komponenten werden unter Verwendung der SQL Anywhere-Installation installiert. Unter Linux werden die Relay Server-Dateien als Teil der SQL Anywhere-Installation im Verzeichnis */opt/sqlanywhere16* installiert.

Kontext und Bemerkungen

Interaktive Schnellsetup-Funktion
Eine interaktive Schnellsetup-Funktion wird als Alternative zu dieser Prozedur bereitgestellt. Die interaktive Schnellsetup-Funktion umfasst folgende Schritte:

1. Konfiguriert den Webserver für den Relay Server.
2. Erstellt eine Demoanwendung.
3. Generiert eine Kurzreferenz.

Das Schnellsetup besteht aus zwei Hauptschritten:

1. Der Apache-Webserver wird für den Relay Server konfiguriert. Für diesen Schritt können Sie das Skript *ap-setup.sh* im Verzeichnis *install-dir/relayserver/quicksetup_apache* ausführen.
2. Relay Server-Testdienste werden erstellt und gestartet. Für diesen Schritt können Sie das Skript *rs-test-setup.sh* im Verzeichnis *install-dir/relayserver/quicksetup_apache* ausführen.

Der Relay Server für Apache umfasst die folgenden Programmdateien:

- *mod_rs_ap_client.so*
- *mod_rs_ap_server.so*
- *rshost*
- *dblgen16.res*
- *libdbtasks16.so*
- *libdbtasks16_r.so*
- *libdbicudt16.so*
- *libdbicu16_r.so*
- *libdblib16_r.so*
- *dbsupport*
- *dbfhide*
- *libdblib16.so*
- *mod_rs_ap_monitor.so*
- *mod_rs_ap_admin.so*

Aufgabe

1. Erstellen Sie die Relay Server-Konfigurationsdatei *rs.config*.
2. Kopieren Sie die Datei *rs.config* in das Verzeichnis *install-dir/relayserver/apache??/bin64*.
3. Bearbeiten Sie die Relay Server-Konfigurationsdatei *rs.config* unter Beachtung folgender Richtlinien.

- Die Datei sollte vier Abschnitte haben:
  - Relay Server-Abschnitt
  - Backend-Farm-Abschnitt
  - Backend-Server-Abschnitt
  - Optionen-Abschnitt
- Jeder Abschnitt beginnt mit einem Abschnitts-Tag, das ein Schlüsselwort, das den Abschnittsnamen angibt, in eckigen Klammern enthält.
- Fügen Sie den einzelnen Abschnitten die entsprechenden Eigenschaften hinzu. Eine Eigenschaft wird durch den Eigenschaftsnamen links von einem Gleichheitszeichen und den dazugehörigen Wert rechts vom Gleichheitszeichen festgelegt. Zum Beispiel: Eigenschaftsname = Wert.
- Die Konfigurationsdatei darf nur 7-Bit-ASCII-Zeichen enthalten.

4. Die Umgebungsvariable LD_LIBRARY_PATH muss die Apache-Verzeichnisse *install-dir/lib64* und *install-dir/relayserver/apache??/bin64* enthalten. Bearbeiten Sie die Datei */<Apache-Verzeichnis>/bin/envvars*, um LD_LIBRARY_PATH festzulegen und zu exportieren.
5. Bearbeiten Sie die Apache-Datei *conf/httpd.conf*.
   a. Fügen Sie folgende Zeilen hinzu, damit die Relay Server-Client- und -Servermodule geladen werden:

```
LoadModule iarelayserver_client_module install-dir/relayserver/
apache??/bin64/mod_rs_ap_client.so

LoadModule iarelayserver_server_module install-dir/relayserver/
apache??/bin64/mod_rs_ap_server.so
```

   Hinweis
   Alle Module werden mit verschiedenen URLs aufgerufen und alle Module suchen explizit nach der Zeichenfolge *iarelayserver* im URL-Pfad. Dieser Teil der URL muss nicht geändert werden.

   b. Fügen Sie die folgende Zeile hinzu, um das Unterstützungsmodul für den SQL Anywhere-Monitor zu laden:

```
LoadModule iarelayserver_monitor_module install-dir/relayserver/
apache??/bin64/mod_rs_ap_monitor.so
```

   c. Fügen Sie die folgende Zeile hinzu, um das Unterstützungsmodul für die Remote-Administration zu laden:

```
LoadModule iarelayserver_admin_module install-dir/relayserver/
apache??/bin64/mod_rs_ap_admin.so
```

   d. Fügen Sie folgende Zeile hinzu, um den Abschnitt <*LocationMatch*> für das Clientmodul zu erstellen:

```
<LocationMatch /cli/iarelayserver/* >
    SetHandler iarelayserver-client-handler
</LocationMatch>
```

   e. Fügen Sie folgende Zeile hinzu, um den Abschnitt <*LocationMatch*> für das Servermodul zu erstellen:

```
<LocationMatch /srv/iarelayserver/* >
    SetHandler iarelayserver-server-handler
```



```
    RSConfigFile "/install-dir/relayserver/apache??/bin64/rs.config"
</LocationMatch>
```

Hinweis
Sie müssen eine `RSConfigFile`-Direktive angeben, die den Speicherort der Relay Server-Konfigurationsdatei *rs.config* festlegt.

f. Fügen Sie folgende Zeile hinzu, um den Abschnitt <*LocationMatch*> für das SQL Anywhere-Monitormodul zu erstellen:

```
<LocationMatch  /mon/iarelayserver/* >
    SetHandler iarelayserver-monitor-handler
</LocationMatch>
```

g. Fügen Sie folgende Zeile hinzu, um den Abschnitt <*LocationMatch*> für das Remote-Administrationsmodul zu erstellen:

```
<LocationMatch /admin/iarelayserver/* >
  SetHandler iarelayserver-admin-handler
</LocationMatch>
```

h. Wenn die TimeOut-Direktive festgelegt wird, stellen Sie sicher, dass sie mindestens auf 60 Sekunden gesetzt wird.

6. Ist unter Linux eine der folgenden Umgebungsvariablen global festgelegt, wenn Apache einen Prozess startet, sind für die Konfiguration von Apache keine weiteren Aktionen erforderlich: $TMP, $TMPDIR oder $TEMP.

   Wenn eine der oben genannten Umgebungsvariablen nicht global festgelegt ist oder die standardmäßige Relay Server-Logdatei in einem spezifischen temporären Verzeichnis gespeichert werden soll (z.B. wenn der Status-Manager automatisch und ohne Anpassungen gestartet wird), bearbeiten Sie die Datei */<Apache-Verzeichnis>/bin/envvars*, um TMP festzulegen und zu exportieren.

   Beispiel: Zur Bearbeitung von $TMP in der envvars-Datei gehen Sie wie folgt vor:

   ```
   set TMP="/tmp"
   export TMP
   ```

   Hiermit legen Sie die Umgebungsvariable in der Shell fest, die Apache vor dem Starten der Prozesse erstellt.

   Hinweis
   Der Apache-Benutzerprozess muss eine Schreibberechtigung für das angegebene *tmp*-Verzeichnis besitzen.

7. Gehen Sie folgendermaßen vor, um die Relay Server-Konfiguration zu aktualisieren, während der Relay Server gestartet wird:

   a. Kopieren Sie die aktualisierte Konfigurationsdatei in das Verzeichnis *install-dir/relayserver/apache??/bin64*.

   b. Führen Sie im Verzeichnis *install-dir/relayserver/apache??/bin64* die folgende Befehlszeile aus, um die Aktualisierung der Konfiguration zu übernehmen:

```
rshost -u -f rs.config
```

c. Wenn der Relay Server als Farm mit mehreren Servern eingerichtet wurde, wiederholen Sie die oben beschriebenen Schritte für jeden Computer in der Relay Server-Farm.

Ergebnisse

Die Relay Server-Konfigurationsdatei wird auf allen Computern in der Relay Server-Farm bereitgestellt.

Hinweise dazu, welche Versionen von Apache unter Linux unterstützt werden, finden Sie unter http://www.sybase.com/detail?id=1061806.

Siehe auch

- „Relay Server-Konfigurationsdatei“ auf Seite 29
- „Performancetipps“ auf Seite 14
- „Relay Server-Status-Manager“ auf Seite 25
- „Relay Server-Status-Manager als Dienst“ auf Seite 25
- „Befehlszeilensyntax des Relay Server Status-Managers (rshost)“ auf Seite 26
- Dienstprogramm zum Verschleiern von Dateien (dbfhide) auf Seite 47

Gleichzeitige Verbindungen

Relay Server-Version
Dies gilt NUR für Relay Server 12.0.x und höher auf dem Apache-Webserver. Dies gilt nicht für Relay Server 11.0.x auf dem Apache-Webserver.

Der Apache-Webserver steuert gleichzeitige Verbindungen (gleichzeitige Anforderungen) unter Verwendung der MaxClient-Direktive. Die Standardeinstellung für die MaxClient-Direktive ist 256. Bei mehr als 256 gleichzeitigen Verbindungen mit den Apache-Webserver werden Verbindungen ab der Grenze von 256 in die Warteschlange gestellt, normalerweise basierend auf der ListenBacklog-Direktive. Die Standardeinstellung für die ListenBacklog-Direktive ist 511.

Damit der Apache-Webserver von mehr als 256 gleichzeitige Verbindungen verarbeiten kann, muss die Einstellung für die MaxClient-Direktive in der Datei *httpd.conf* festgelegt werden. Wenn der Wert für die MaxClient-Direktive erhöht wird, muss auch der Wert für die ServerLimit-Direktive geändert werden, um die höhere Anzahl von Apache-Prozessen auf dem Webserver zuzulassen.

Relay Server 12.0.x und höher umfassen einen Semaphor-Manager, mit dem die Nutzung von Semaphoren durch den Relay Server verwaltet wird. Dadurch ist es nicht nötig, die Anzahl der Semaphorengruppen im System zu erhöhen, wenn die Direktiven MaxClient und ServerLimit geändert werden.

Wenn Sie die Anzahl der gleichzeitigen Verbindungen erhöhen möchten, fügen Sie folgende Zeilen zur Datei *httpd.conf* hinzu:

```
ServerLimit 1000
MaxClient 1000
```

Zu den weiteren Apache-Direktiven, die angepasst werden können, wenn der Webserver stark belastet ist, gehören folgende:

```
MaxSpareServers
MinSpareServers
StartServers
```

Relay Server-Status-Manager

Der Relay Server-Status-Manager ist ein Prozess, der für die Verwaltung der Zustandsinformationen des Relay Servers für die Clientanforderungen und die Sitzungen des Outbound Enablers verantwortlich ist. Der Status-Manager ist auch für die Verwaltung der vom Relay Server verwendeten Logdatei verantwortlich. Der Status-Manager kann entweder automatisch vom Relay Server oder als Dienst gestartet werden.

Der Standardname der Logdatei ist *ias_relay_server_host.log*. Unter Windows befindet sich diese Datei in dem in der Umgebungsvariablen TEMP angegebenen Verzeichnis. Unter Linux befindet sich diese Datei in dem in der Umgebungsvariablen TMP, TEMP oder TMPDIR angegebenen Verzeichnis. Wenn keine dieser Variablen festgelegt ist, wird im Verzeichnis */tmp* eine Logdatei erstellt.

Hinweis
Der Apache-Benutzerprozess muss eine Schreibberechtigung für das angegebene *tmp*-Verzeichnis besitzen.

Wenn der Server normal heruntergefahren wird, benennt der Status-Manager die Logdatei um. Hierbei verwendet er das Format *<jjmmtt><nn>.log*, wobei *<jjmmtt>* das Datum ist, an dem die Logdatei umbenannt wurde, und *<nn>* die sequenzielle Versionsnummer der Logdatei für den Tag.

Das Starten des Status-Managers als Dienst ist die empfohlene Methode. Das manuelle Starten des Status-Managers von der Befehlszeile aus wird nicht unterstützt.

Sie können die Optionen festlegen, die vom Relay Server zum Starten des Status-Managers verwendet werden. Um die Optionen zu ändern, legen Sie die Eigenschaft **Start** im Optionen-Abschnitt der Relay Server-Konfigurationsdatei fest. Beispiel:

```
[options]
start = "rshost -o c:\temp\myrshost.log"
```

Sie müssen den Namen des Relay Server Status-Managers (`rshost`) vor den Optionen festlegen.

Relay Server-Status-Manager als Dienst

Der Status-Manager kann mithilfe des Dienstprogramms für Dienste (dbsvc) als Dienst gestartet werden.

Mithilfe des Dienstprogramms für Dienste können Dienste erstellt, geändert und gelöscht werden. Um eine vollständige Liste der Syntaxinformationen anzuzeigen, führen Sie *dbsvc* ohne Optionen aus.

So richten Sie einen automatisch gestarteten Status-Manager-Dienst mit dem Namen RelayServer unter Windows ein

```
dbsvc -as -s auto -t rshost -w RelayServer "%SQLANY16%\RelayServer\IIS\BinXX
\Server\rshost.exe" -q -qc -f "%SQLANY16%\RelayServer\IIS\BinXX\Server
\rs.config" -o "c:\temp\ias_relay_server.log"
```

So richten Sie einen automatisch gestarteten Status-Manager-Dienst mit dem Namen RelayServer unter Unix ein

```
dbsvc -y -a <apache-user> -t rshost -w RelayServer -q -qc -f /<your-
director>/rs.config -os 100K -ot /tmp/rs.log
```

Bemerkungen

Die Syntax von dbsvc unter Windows unterscheidet sich von der unter Unix. In Unix geben Sie nicht den vollständigen Pfad der Programmdatei als ersten Parameter nach der Parameteroption -w an.

Verwenden Sie nur vollständige Pfade.

Verwenden Sie unter Unix ein Benutzerkonto (möglichst dasselbe), sodass Apache-Benutzerprozesse sich an den gemeinsam genutzten Speicher des Status-Managers anhängen können und in der Lage sind, dort zu lesen und zu schreiben.

Starten des Diensts

```
dbsvc.exe -u rs
```

So stoppen Sie den Dienst

```
dbsvc.exe -x rs
```

So deinstallieren Sie den Dienst

```
dbsvc.exe -d rs
```

Siehe auch

- „Optionen-Abschnitt“ auf Seite 35.

Automatisches Starten des Relay Server-Status-Managers

Der Status-Manager-Prozess wird automatisch gestartet, wenn der erste Outbound Enabler eine Verbindung mit dem Relay Server herstellt. Der Standardspeicherort der Logdatei ist *%temp%\ias_relay_server_host.log*.

Befehlszeilensyntax des Relay Server Status-Managers (rshost)

rshost [*option*]+

Parameter

Optionen Der Status-Manager kann mit den folgenden Optionen konfiguriert werden. Sie sind alle optional.

rshost-Optionen	Beschreibung
-f *filename*	Zeigt den Namen der Relay Server-Konfigurationsdatei an.
-o *filename*	Zeigt den Namen der für die Protokollierung verwendeten Datei an. Der Standardspeicherort der Logdatei für Windows ist *%temp%* ***ias_relay_server_host.log**, wenn die Option -o nicht angegeben wird.
-os *size*	Steuert die Größe der Logdatei und liefert zusätzliche Informationen im Logdatei-Banner. Wenn -os angegeben ist, wird die alte Logdatei unter Verwendung des Formats *<jjmmtt><nn>.olg* umbenannt. Das Logdatei-Banner wird in die neue aktive Logdatei neu geschrieben, wobei der Computername, die Prozessorarchitektur, das Build-Ziel und Betriebs-systeminformationen hinzugefügt werden.
-oq	Verhindert, dass ein Popupfenster angezeigt wird, falls beim Start ein Fehler auftritt.
-q	Bewirkt die Ausführung in einem minimierten Fenster
-qc	Schließt das Fenster nach Abschluss des Vorgangs.
-u	Aktualisiert die Konfiguration eines laufenden Relay Servers.
-ua	Archiviert die Logdatei in *<jjmmtt><nn>.log* und kürzt die Datei.

Relay Server-Konfigurationsdatei

Mit einer Relay Server-Konfigurationsdatei werden eine Relay Server-Farm sowie die Backend-Serverfarmen definiert, die Verbindungen mit der Relay Server-Farm herstellen. Die Relay Server-Konfigurationsdatei ist in folgende Abschnitte unterteilt:

- Relay Server-Abschnitt
- Backend-Farm-Abschnitt
- Backend-Server-Abschnitt
- Optionen-Abschnitt

Jeder Abschnitt beginnt mit einem Abschnitt-Tag. In einem Abschnitt-Tag ist ein Schlüsselwort, das den Abschnittsnamen angibt, in eckige Klammern eingeschlossen. Beispiel: [relay_server] gibt den Start des Relay Server-Abschnitts an.

Auf den Abschnitt-Tag folgen eine Reihe von Zeilen, die verschiedene Eigenschaften für den definierten Abschnitt festlegen. Eine Eigenschaft wird durch den Eigenschaftsnamen links von einem Gleichheitszeichen und den dazugehörigen Wert rechts vom Gleichheitszeichen festgelegt. Zum Beispiel: `Eigenschaftsname = Wert`. Die Abschnitts- und Eigenschaftsnamen unterscheiden nicht zwischen Groß- und Kleinschreibung. Kommentare sind durch ein Rautenzeichen (#) am Anfang einer Zeile gekennzeichnet.

Die Konfigurationsdatei darf nur 7-Bit-ASCII-Zeichen enthalten. Die Abschnitte können in beliebiger Reihenfolge angegeben werden.

Relay Server-Konfigurationsdateien können mithilfe des Relay Server-Plug-Ins für Sybase Central erstellt, importiert und bereitgestellt werden.

Siehe auch

- „Relay Server-Plug-In für Sybase Central“ auf Seite 55
- „Konfigurationsaktualisierungen für die Relay Server-Farm“ auf Seite 51

Relay Server-Abschnitt

Im Relay Server-Abschnitt wird ein einzelner Relay Server festgelegt, sodass für jeden Relay Server in der Farm ein eigener Relay Server-Abschnitt erforderlich ist. Dieser Abschnitt ist durch das Schlüsselwort relay_server gekennzeichnet.

Eigenschaften im Relay Server-Abschnitt

Die folgenden Eigenschaften können in einem Relay Server-Abschnitt festgelegt werden:

- **enable** Legt fest, ob dieser Relay Server Bestandteil der Relay Server-Farm ist. Die möglichen Werte sind:
  - **Ja** Zeigt an, dass dieser Relay Server in die Relay Server-Farm einbezogen werden muss.

  - **Nein** Zeigt an, dass dieser Relay Server nicht in die Relay Server-Farm einbezogen werden soll.

  Der Standardwert ist Yes. Diese Eigenschaft ist optional.

- **host** Der Hostname oder die IP-Adresse, die vom Outbound Enabler verwendet werden müssen, um eine direkte Verbindung mit dem Relay Server herzustellen.
- **http_port** Der HTTP-Port, der vom Outbound Enabler verwendet werden muss, um eine direkte Verbindung mit dem Relay Server herzustellen. Der Wert **0** oder **off** deaktiviert HTTP-Verbindungen. Standardmäßig ist diese Eigenschaft aktiviert und auf 80 festgelegt.
  - **0 oder off** Deaktiviert den HTTP-Zugriff vom Outbound Enabler aus.
  - **1 bis 65535** Aktiviert HTTP am angegebenen Port.
- **https_port** Der HTTPS-Port, der vom Outbound Enabler verwendet werden soll, um eine direkte Verbindung mit dem Relay Server herzustellen. Der Wert **0** oder **off** deaktiviert HTTPS-Verbindungen. Standardmäßig ist diese Eigenschaft aktiviert und auf 443 festgelegt.
  - **0 oder off** Deaktiviert den HTTPS-Zugriff vom Outbound Enabler aus.
  - **1 bis 65535** HTTPS am angegebenen Port aktivieren.
- **description** Geben Sie eine benutzerdefinierte Beschreibung mit einer maximalen Länge von 2048 Zeichen ein. Diese Eigenschaft ist optional.

Backend-Farm-Abschnitt

Der Backend-Farm-Abschnitt legt die Eigenschaften einer Backend-Serverfarm fest. Eine Backend-Serverfarm ist eine Gruppe von homogenen Backend-Servern. Ein Client, der eine Anforderung über die Relay Server-Farm stellt, muss die Backend-Serverfarm angeben, an die die Anforderung gerichtet ist. Für jede Backend-Serverfarm ist ein eigener Backend-Farm-Abschnitt vorhanden.

Dieser Abschnitt ist durch das Schlüsselwort backend_farm gekennzeichnet.

Eigenschaften im Backend-Farm-Abschnitt

Die folgenden Eigenschaften können in einem Backend-Farm-Abschnitt festgelegt werden:

- **active_cookie** Gibt an, ob ein Cookie gesetzt ist, um die Client-Server-Affinität zu bewahren.
  - **yes** Dies ist die Standardeinstellung. Um die Client-Server-Affinität zu bewahren, fügt der Relay Server einen HTTP-Standardbefehl zum Setzen eines Cookies mit einem systemeigenen Cookie-Namen in die Antwort ein.
  - **no** Es ist kein aktives Cookie gesetzt. Verwenden Sie diese Option, wenn die Backend-Farm eine sitzungslose Browseranwendung bedient. Dies gilt beispielsweise, wenn die Backend-Farm einen sitzungslosen SQL Anywhere-Webdienst bereitstellt.

  Um beste Ergebnisse zu erzielen, setzen Sie dieses Steuerelement wie folgt:

Backend-Servertyp	active_cookie-Einstellung	active_header-Einstellung
MobiLink	no	yes
SQL Anywhere	no	no

Bei einem Backend mit MobiLink-Server sollte es keine Probleme geben, wenn sowohl active_cookie als auch active_header auf "yes" gesetzt werden. Wenn beide auf "yes" gesetzt werden, können jedoch redundante Sitzungsinformationen in jede HTTP-Anforderung bzw. -Antwort in einer Sitzung eingefügt werden. Um kumulative Datenübertragungskosten potenziell zu sparen, können Sie möglicherweise active_cookie auf "no" setzen. Testen Sie alle Netzwerkszenarien, um sich zu vergewissern, ob die gewählten Einstellungen für die Anweisungen in allen Fällen funktionieren.

- **active_header** Gibt an, ob ein Header gesetzt ist, um die Affinität der Client-Server-Sitzung zu bewahren.
  - **yes** Dies ist die Standardeinstellung. Um die Client-Server-Sitzungsaffinität zu bewahren, fügt der Relay Server einen systemeigenen Header in die Antwort ein, falls Zwischenstationen den active_cookie-Wert manipulieren.
  - **no** Es wird kein systemeigener Header gesetzt. Durch Aktivieren dieser Option können Sie das Verkehrsaufkommen verringern, wenn die Backend-Farm nur Browseranwendungen bedient oder wenn das aktive Cookie für alle Clients dieser Backend-Farm gut funktioniert.
- **renew_overlapped_cookie** Wenn ein Clientgerät ein Standard-Cookie für mehrere gleichzeitige Verbindungen gemeinsam nutzt, können Timeout-Fehler auftreten. Dieses Problem wird im Relay Server-Log angezeigt.
  - **yes** (Standardwert.) Wenn renew_overlapped_cookie auf "yes" gesetzt ist, erkennt Relay Server die Cookieüberlappung für die Farm, bei der diese Eigenschaft explizit aktiviert ist, und erneuert das überlappende Cookie durch Erstellen einer neuen Affinitätsbindung. Die Anforderung für die Erneuerung wird immer noch an denselben Backend-Server geleitet, aber nicht über dieselbe Backend-Verbindung wie die laufende Anforderung, mit der sie sich überlappt, sondern über eine neue Backend-Verbindung, die erstellt wird.
  - **no** Diese Option muss auf "no" gesetzt werden, wenn dieses Verhalten nicht erwünscht ist.
- **backend_security** Gibt die Sicherheitsstufe an, die von einem Outbound Enabler in der Backend-Serverfarm verlangt wird, um Verbindungen mit der Relay Server-Farm herzustellen. Die folgenden Werte können angegeben werden:
  - **on** Gibt an, dass alle Verbindungen von der Backend-Farm mit HTTPS hergestellt werden müssen.
  - **off** Gibt an, dass alle Verbindungen von der Backend-Farm mit HTTP hergestellt werden müssen.

  Diese Eigenschaft ist optional. Wenn keine Werte angegeben sind, können HTTP oder HTTPS für die Verbindung benutzt werden.

- **client_security** Gibt die Sicherheitsstufe an, die die Backend-Serverfarm von ihren Clients verlangen soll. Die folgenden Werte können angegeben werden:
  - **on** Legt fest, dass Clients sich über HTTPS verbinden müssen.
  - **off** Legt fest, dass Clients sich über HTTP verbinden müssen.

  Diese Eigenschaft ist optional. Wenn kein Wert angegeben ist, können sich Clients über HTTP oder HTTPS verbinden.
- **description** Geben Sie eine benutzerdefinierte Beschreibung mit einer maximalen Länge von 2048 Zeichen ein. Diese Eigenschaft ist optional.
- **enable** Gibt an, ob Verbindungen von dieser Backend-Serverfarm zugelassen werden sollen. Die möglichen Werte sind:
  - **Ja** Lässt Verbindungen von dieser Backend-Serverfarm zu.
  - **Nein** Lässt keine Verbindungen von dieser Backend-Serverfarm zu.

  Der Standardwert ist Yes. Diese Eigenschaft ist optional.
- **id** Der Name, der der Backend-Serverfarm zugeordnet ist, maximal 2048 Zeichen lang.
- **forward_x509_identity** Das SAP NetWeaver Gateway bietet mehrere Möglichkeiten zur Authentifizierung von Clients, einschließlich Weiterleitung des X.509-Zertifikats durch vertrauenswürdige Vermittler. Wenn diese Eigenschaft auf "yes" gesetzt ist, kann der Relay Server weitergeleitete Client-Identitätsinformationen von einem vertrauenswürdigen Vermittler extrahieren und mit HTTP-Headern an das SAP NetWeaver Gateway oder den Web Dispatcher weiterleiten. Die Standardeinstellung ist "no".
- **forwarder_certificate_issue** Falls eine Kette von SAP-Vermittlern vorhanden ist, sind die Header für die Client-Identität möglicherweise bereits in der Anforderung enthalten. Möglicherweise wird jedoch nicht allen Clients die Berechtigung als Vermittler erteilt. Das Standardverhalten ist deshalb, dass die vorhandenen Header durch die Identität des Vermittlers ersetzt werden. Um die Berechtigung für einen Weiterleitenden zum Weiterleiten anderer Clientidentitäten zu erteilen, können Sie **forwarder_certificate_issuer**=*match-string* und **forwarder_certificate_subject**=*match-string* setzen, wobei *match-string* anhand einer serialisierten Form des entsprechenden kombinierten Namensfelds im Zertifikat geprüft wird. Sie können ein Fragezeichen (?) verwenden, um nach Übereinstimmungen für ein beliebiges Zeichen zu suchen, und ein Sternchen (*) für Übereinstimmungen mit einer beliebigen Zeichenfolge. Verwenden Sie "\" als führendes Escapezeichen für ?, * oder \, wenn Sie nach einer literalen Übereinstimmung suchen müssen.

  Beispiel:

  ```
  forwarder_certificate_issuer = 'CN = quicksigner, OU = security
  department, O = my org, L = my city, S = my state, C = my country'
  ```
- **forwarder_certificate_subject** Falls eine Kette von SAP-Vermittlern vorhanden ist, sind die Header für die Client-Identität möglicherweise bereits in der Anforderung enthalten. Möglicherweise wird jedoch nicht allen Clients die Berechtigung als Vermittler erteilt. Das Standardverhalten ist

deshalb, dass die vorhandenen Header durch die Identität des Vermittlers ersetzt werden. Um die Berechtigung für einen Weiterleitenden zum Weiterleiten anderer Clientidentitäten zu erteilen, können Sie **forwarder_certificate_issuer**=*match-string* und **forwarder_certificate_subject**=*match-string* setzen, wobei *match-string* anhand einer serialisierten Form des entsprechenden kombinierten Namensfelds im Zertifikat geprüft wird. Sie können ein Fragezeichen (?) verwenden, um nach Übereinstimmungen für ein beliebiges Zeichen zu suchen, und ein Sternchen (*) für Übereinstimmungen mit einer beliebigen Zeichenfolge. Verwenden Sie "\" als führendes Escapezeichen für ?, * oder \, wenn Sie nach einer literalen Übereinstimmung suchen müssen.

Beispiel:

```
forwarder_certificate_subject = 'CN = mySapWD??.my.com, OU = Sybase, O =
SAP, *'
```

- **max_client_buffer** Es kann vorkommen, dass Shared Memory-Ressourcen aufgrund von Problemen mit übermäßigem Puffern von Serverantworten erschöpft werden. Dies kann an einer großen Anzahl von Clients, langsamen Lesevorgängen der Clients oder großen HTTP-Antworten liegen. **max_client_buffer =** *memory size* erlaubt Ihnen, einen Höchstwert für die Speicherpuffergröße für jeden Client anzugeben. Der Standardwert ist 1 MB. Der Höchstwert ist 4 GB.

- **verbosity** Sie können den Parameter verbosity auf folgende Stufen setzen:

  - **0** Nur Fehler protokollieren. Verwenden Sie diese Protokollierungsstufe für das Deployment. Dies ist die Standardeinstellung.

  - **1** Anforderungsprotokollierung. Alle HTTP-Anforderungen werden in die Logdatei geschrieben.

  - **2** Anforderungsprotokollierung. Bietet eine detailliertere Ansicht der HTTP-Anforderungen.

  - **3 oder höher** Ausführliche Protokollierung. Dies wird in erster Linie für den technischen Support verwendet.

  Fehler werden unabhängig von der Protokollstufe angezeigt, während Warnungen nur bei einer höheren Protokollstufe als 0 angezeigt werden.

Siehe auch

„Affinität“ auf Seite 6

Backend-Server-Abschnitt

Der Backend-Server-Abschnitt definiert eine Backend-Serververbindung. Er gibt die Informationen an, die vom Outbound Enabler verwendet werden, wenn er im Auftrag eines Backend-Servers eine Verbindung mit der Relay Server-Farm herstellt. Für jeden Outbound Enabler, der eine Verbindung mit der Relay Server-Farm herstellt, ist ein Backend-Server-Abschnitt vorhanden. Außerdem ordnet der Backend-Server-Abschnitt einen Backend-Server einer Backend-Serverfarm zu.

Die folgenden Backend-Server werden für die Verwendung mit dem Relay Server unterstützt:

- Afaria
- Mobile Office
- MobiLink
- Mobile Office
- SQL Anywhere
- Unwired Server
- Sybase Unwired Platform

Hinweis
Weitere Hinweise dazu, welche Backend-Server unterstützt werden, finden Sie in der Lizenzvereinbarung oder auf der Seite mit den SQL Anywhere-Komponenten nach Plattform. Siehe http://www.sybase.com/detail?id=1061806.

Dieser Abschnitt ist durch das Schlüsselwort backend_server gekennzeichnet.

Eigenschaften im Backend-Server-Abschnitt

Die folgenden Eigenschaften können in einem Backend-Server-Abschnitt festgelegt werden:

- **description** Geben Sie eine benutzerdefinierte Beschreibung mit einer maximalen Länge von 2048 Zeichen ein. Diese Eigenschaft ist optional.
- **enable** Gibt an, ob Verbindungen von diesem Backend-Server zugelassen werden sollen. Die möglichen Werte sind:
  - **Ja** Lässt Verbindungen von diesem Backend-Server zu.
  - **Nein** Lässt keine Verbindungen von diesem Backend-Server zu.

  Der Standardwert ist Yes. Diese Eigenschaft ist optional.
- **farm** Der Name der Backend-Serverfarm, zu der dieser Backend-Server gehört.
- **id** Der Name, der der Backend-Serververbindung zugeordnet ist, maximal 2048 Zeichen lang.
- **MAC** Die MAC-Adresse des Netzwerkadapters, die vom Outbound Enabler verwendet wird, um mit dem Relay Server zu kommunizieren. Die Adresse wird mithilfe des Formats IEEE 802 MAC-48 festgelegt. Das richtige Format der MAC-Adresse entnehmen Sie der Konsole oder dem Log des Relay Server Outbound Enablers. Diese Eigenschaft ist optional. Wenn sie nicht angegeben ist, wird die MAC-Adresse nicht überprüft.
- **token** Ein Sicherheitstoken, der vom Relay Server zum Authentifizieren der Backend-Serververbindung verwendet wird, maximal 2048 Zeichen lang. Diese Eigenschaft ist optional.
- **verbosity** Sie können den Parameter verbosity auf folgende Stufen setzen:

- **0** Nur Fehler protokollieren. Verwenden Sie diese Protokollierungsstufe für das Deployment. Dies ist die Standardeinstellung.
- **1** Anforderungsprotokollierung. Alle HTTP-Anforderungen werden in die Logdatei geschrieben.
- **2** Anforderungsprotokollierung. Bietet eine detailliertere Ansicht der HTTP-Anforderungen.
- **3 oder höher** Ausführliche Protokollierung. Dies wird in erster Linie für den technischen Support verwendet.

Fehler werden unabhängig von der Protokollstufe angezeigt, während Warnungen nur bei einer höheren Protokollstufe als 0 angezeigt werden.

Optionen-Abschnitt

Der Optionen-Abschnitt dient zur Festlegung von Eigenschaften, die auf jeden Relay Server in der Farm angewendet werden. Es ist nur ein Optionen-Abschnitt erlaubt.

Dieser Abschnitt ist durch das Schlüsselwort "options" gekennzeichnet.

Eigenschaften im Optionen-Abschnitt

Die folgenden Eigenschaften können in einem Optionen-Abschnitt festgelegt werden:

- **shared_mem** Legt den maximalen gemeinsam genutzten Speicher fest, den der Relay Server zur Statusprotokollierung verwendet. Es kann sinnvoll sein, diese Einstellung zu ändern, wenn eine oder mehrere der folgenden Bedingungen gegeben sind:
  - Verbesserte Geschwindigkeit im Netzwerk zwischen dem Relay Server und dem Outbound Enabler
  - Signifikante Zunahme bei der Anzahl der Backend-Farmen
  - Signifikante Zunahme bei der Größe der Backend-Farmen
  - Signifikante Zunahme bei der Anzahl der Clients
  - Signifikante Zunahme bei der HTTP-Antwort-Größe
  - Hinzufügen langsamerer Clients oder langsamerer Netzwerke

  Der Standardwert ist 10 MB. Der Höchstwert ist 4 GB. Diese Eigenschaft ist optional.

- **up_pad_size** Fügt eine Auffüllung der maximalen Größe einer Übertragungseinheit ein, um die Latenzzeit zu verbessern, wenn über den RSOE-Kanal eine geringe Anzahl an kleinen Anforderungen hochgeladen wird. Die Standardeinstellung ist 1460. Das ist die optimale MTU-Größe von (gewöhnlich) 1500, abzüglich 40 Byte bei TCP und IP Overhead. Um das Auffüllen mit Nullen zu deaktivieren, setzen Sie up_pad_size auf 0.

  Um den optimalen MTU-Wert unter Windows zu ermitteln, können Sie das Ping-Dienstprogramm mit den Optionen -f und -l verwenden .Die Option -f gibt an, dass die Daten nicht fragmentiert werden. Die Option -l gibt die Datengröße an. Das Ziel ist es, die MTU so einzustellen, dass es keine Datenfragmentierung gibt. Beispiel:

```
ping -f -l 1494 127.0.0.
Pinging 127.0.0.1 with 1494 bytes of data:
Packet needs to be fragmented but DF set.
...
ping -f -l 1492 127.0.0.1
Pinging 127.0.0.1 with 1492 bytes of data:
Reply from 127.0.0.1: bytes=1492 time<1ms TTL=128
```

In diesem Beispiel beträgt die maximale Datengröße, die ohne Fragmentierung übergeben werden kann, 1492. Der Ping enthält acht Byte ICMP Overhead und die MTU ist daher 1500.

Um den optimalen Wert für die MTU unter Unix zu ermitteln, benutzen Sie **ifconfig** oder **ping** mit der Option -M **hint** (um den Pfad für die MTU Discovery-Strategie auszuwählen) und die Option -s **packetsize** (um die Anzahl der zu sendenden Datenbytes anzugeben). Weitere Hinweise finden Sie in der Dokumentation zum Ping-Dienstprogramm.

- **verbosity** Sie können den Parameter verbosity auf folgende Stufen setzen:
  - **0** Nur Fehler protokollieren. Verwenden Sie diese Protokollierungsstufe für das Deployment. Dies ist die Standardeinstellung.
  - **1** Anforderungsprotokollierung. Alle HTTP-Anforderungen werden in die Logdatei geschrieben.
  - **2** Anforderungsprotokollierung. Bietet eine detailliertere Ansicht der HTTP-Anforderungen.
  - **3 oder höher** Ausführliche Protokollierung. Dies wird in erster Linie für den technischen Support verwendet.

  Fehler werden unabhängig von der Protokollstufe angezeigt, während Warnungen nur bei einer höheren Protokollstufe als 0 angezeigt werden.

Format der Relay Server-Konfigurationsdatei

Das grundlegende Format einer Relay Server-Konfigurationsdatei sieht wie folgt aus:

```
#
# Options
#
[options]
# List of Relay Server properties that apply to all Relay Servers
option = value

#
# Define a Relay Server section, one for each
# Relay Server in the Relay Server farm
#
[relay_server]
# List of properties for the Relay Server
property = value

#
# Define a backend server farm section, one for each backend
# server farm
#
```

```
[backend_farm]
# List of properties for a backend server farm
property = value

#
# Define a backend server section, one for each
# Outbound Enabler connecting to the Relay Server farm
#
[backend_server]
# List of properties for the backend server connection
property = value
```

Outbound Enabler

Der Outbound Enabler wird auf demselben Computer ausgeführt wie der Backend-Server. Er hat folgenden Zweck:

- Öffnen einer abgehenden Verbindung zu der Relay Server-Farm, die in der DMZ ausgeführt wird, vom Computer im Unternehmens-LAN aus.
- Weiterleiten von Clientanforderungen vom Relay Server an den Backend-Server und Weiterleiten von Antworten des Backend-Servers an den Client über den Relay Server.

Wenn der Outbound Enabler startet, stellt er eine HTTP-Anforderung, um die Liste der in der Farm laufenden Relay Server abzurufen. Hierzu wird der Server-URL verwendet, der der Webserver-Erweiterungskomponente des Relay Servers zugeordnet ist. Der Server-URL kann direkt einem Relay Server oder einem Lastverteiler zugeordnet werden. Wenn der Server-URL einem Lastverteiler zugeordnet wird, leitet der Lastverteiler die Anforderung an einen der Relay Server weiter, die in der Farm ausgeführt werden. Der Relay Server, der die Anforderung vom Outbound Enabler empfängt, gibt die Verbindungsinformationen für alle Relay Server in der Farm zurück. Der Outbound Enabler erstellt dann zwei abgehende Verbindungen, so genannte Kanäle, für jeden zurückgegebenen Relay Server. Ein Kanal, der so genannte Up-Kanal, wird während einer HTTP-Anforderung mit einer grundsätzlich endlosen Antwort erstellt. Die Antwort ist ein kontinuierlicher Datenstrom von Clientanforderungen vom Relay Server an den Outbound Enabler. Der zweite Kanal, der so genannte Down-Kanal, wird mithilfe einer HTTP-Anforderung mit einer grundsätzlich endlosen Länge des Inhalts erstellt. Die Anforderung besteht aus einem kontinuierlichen Datenstrom von Serverantworten an Clientanforderungen.

Wenn der Outbound Enabler auf dem UP-Kanal eine Clientanforderung von einem der verbundenen Relay Server empfängt, leitet er sie an den Backend-Server weiter, den der Outbound Enabler bedient. Sobald eine Antwort vom Backend-Server empfangen wurde, wird sie auf dem DOWN-Kanal an den Relay Server weitergeleitet, von dem die entsprechende Anforderung stammt.

Hinweis
Die folgenden Backend-Server werden für die Verwendung mit dem Relay Server unterstützt:

- Afaria
- Mobile Office
- MobiLink
- Mobile Office
- SQL Anywhere
- Unwired Server
- Sybase Unwired Platform

Weitere Hinweise dazu, welche Backend-Server unterstützt werden, finden Sie in der Lizenzvereinbarung oder auf der Seite mit den SQL Anywhere-Komponenten nach Plattform. Siehe http://www.sybase.com/detail?id=1061806.

Outbound Enabler-Syntax

rsoe [*option*]+

rsoe @{ *filename* | *environment-variable* } ...

Parameter

Optionen Die folgenden Optionen können für den Outbound Enabler verwendet werden. Optionen, die Standardwerte enthalten, sind optional. Als Minimum muss der Outbound Enabler die Verbindungszeichenfolge für den Relay Server (-cr), die Farm (-f) und den Servernamen (-id) übergeben. Wenn ein Sicherheitstoken konfiguriert ist, muss dieses ebenfalls angegeben werden (-t).

rsoe-Optionen	Beschreibung
@data	Liest Optionen aus der angegebenen Umgebungsvariablen oder Konfigurationsdatei. 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) auf Seite 47.

rsoe-Optionen	Beschreibung
-cr *"connection-string"*	Gibt die Relay Server-Verbindungszeichenfolge an. Das Format der Relay Server-Verbindungszeichenfolge ist eine durch Semikola getrennte Liste von Name-Wert-Paaren. Die Name-Wert-Paare bestehen aus Folgendem: ○ **host** IP-Adresse oder Hostname des Relay Servers. Der Standardwert ist localhost. Siehe „host“ [*MobiLink - Clientadministration*]. ○ **port** Der Port, an dem der Relay Server auf Daten wartet. Diese Angabe ist erforderlich. Siehe „port“ [*MobiLink - Clientadministration*]. ○ **http_userid** Benutzer-ID für die Authentifizierung. Optional. Sehen Sie in der Dokumentation für Ihren Webserver (oder Proxy) nach, wie Sie die HTTP-Authentifizierung einrichten. Siehe „http_userid“ [*MobiLink - Clientadministration*]. ○ **http_password** Kennwort für die Authentifizierung. Optional. Sehen Sie in der Dokumentation für Ihren Webserver (oder Proxy) nach, wie Sie die HTTP-Authentifizierung einrichten. Siehe „http_password“ [*MobiLink - Clientadministration*]. ○ **http_proxy_userid** Benutzer-ID für die Proxy-Authentifizierung. Optional. Sehen Sie in der Dokumentation für Ihren Webserver (oder Proxy) nach, wie Sie die HTTP-Authentifizierung einrichten. Siehe „http_proxy_userid“ [*MobiLink - Clientadministration*]. ○ **http_proxy_password** Kennwort für die Proxy-Authentifizierung. Optional. Sehen Sie in der Dokumentation für Ihren Webserver (oder Proxy) nach, wie Sie die HTTP-Authentifizierung einrichten. Siehe „http_proxy_password“ [*MobiLink - Clientadministration*]. ○ **proxy_host** Gibt den Hostnamen oder die IP-Adresse des Proxy-Servers an. Optional. Siehe „proxy_host“ [*MobiLink - Clientadministration*]. ○ **proxy_port** Legt die Portnummer des Proxyservers fest. Optional. Siehe „proxy_port“ [*MobiLink - Clientadministration*].

rsoe-Optionen	Beschreibung
	○ **url_suffix** URL-Pfad zur Servererweiterung des Relay Servers. Erforderlich. Standardmäßig erfordert der RSOE, dass url_suffix angegeben wird. Siehe „url_suffix“ [*MobiLink - Clientadministration*]. ○ **https** 0 – HTTP (Standard) 1 – HTTPS Für **https=1** können auch folgende Optionen angegeben werden: ● **certificate_name** Feld für den allgemeinen Namen des Zertifikats. ● **certificate_company** Feld für den Organisationsnamen des Zertifikatausstellers. ● **certificate_unit** Feld für die Organisationseinheit des Zertifikatausstellers. ● **identity** Stellt die Anmeldeinformationen für eine gegenseitig authentifizierte TLS-Verbindung zwischen dem Outbound Enabler und dem Backend-Server bereit. Für den Backend-Server ist gegenseitige Authentifizierung erforderlich. ● **identity_password** Stellt die Anmeldeinformationen für eine gegenseitig authentifizierte TLS-Verbindung zwischen dem Outbound Enabler und dem Backend-Server bereit. Für den Backend-Server ist gegenseitige Authentifizierung erforderlich. ● **fips** Wählen Sie, ob für TLS-Verschlüsselung und Ende-zu-Ende Verschlüsselung FIPS-zertifizierte Verschlüsselungsimplementierungen verwendet werden sollen. ● **trusted_certificates** Eine Datei mit einer Liste von vertrauenswürdigen Stammzertifikaten. Wenn der Backend-Server, und nur der Backend-Server, überprüft werden soll, setzen Sie diese Eigenschaft auf **backend_server_public_cert_filename**. `trusted_certificates=backend_server_public_cert_filename` Wenn unter Windows **trusted_certificates** nicht festgelegt ist, wird der Zertifikatsspeicher des Betriebssystems verwendet.

rsoe-Optionen	Beschreibung
	Weitere Hinweise finden Sie unter „Netzwerkprotokolloptionen des MobiLink-Clients“ [*MobiLink - Clientadministration*].

rsoe-Optionen	Beschreibung
-cs *"connection-string"*	Gibt die Backend-Server-Verbindungszeichenfolge an. Das Format der Verbindungszeichenfolge ist eine durch Semikola getrennte Liste von Name-Wert-Paaren. Die Name-Wert-Paare bestehen aus Folgendem: ○ **host** IP-Adresse oder Hostname des Backend-Servers. Der Standardwert ist localhost. Siehe „host“ [*MobiLink - Clientadministration*]. ○ **port** Der Port, an dem der Backend-Server auf Daten wartet. Diese Angabe ist erforderlich. Der Standardwert ist 0. Siehe „port“ [*MobiLink - Clientadministration*]. ○ **https** 0 – HTTP (Standard) 1 – HTTPS Standardmäßig startet MobiLink das TCP/IP-Kommunikationsprotokoll. Beim Start von MobiLink für die Verwendung mit dem RSOE müssen Sie das für Ihre RSOE-Konfiguration erforderliche Kommunikationsprotokoll starten. Wenn Sie beispielsweise HTTPS als Backend-Server-Sicherheit angeben, muss MobiLink mit HTTPS gestartet werden. Wenn der Parameter "https=1" in der Option -cs enthalten ist, wird der Standardport auf 443 geändert. Siehe „mlsrv16-Option -x“ [*MobiLink - Serveradministration*] Für **https=1** können auch folgende Optionen angegeben werden: ● **identity** Pfad und Dateiname der Identitätsdatei, die für die Serverauthentifizierung verwendet werden sollen. Stellt die Anmeldeinformationen für eine gegenseitig authentifizierte TLS-Verbindung zwischen dem Outbound Enabler und dem Backend-Server bereit. Für den Backend-Server ist gegenseitige Authentifizierung erforderlich. Siehe „identity“ [*MobiLink - Clientadministration*]. ● **identity_password** Ein optionaler Parameter, der ein Kennwort für die Identitätsdatei festlegt. Wenn diese Option angegeben ist, muss die identity-Option ebenfalls angegeben werden. Stellt die Anmeldeinformationen für eine gegenseitig authentifizierte TLS-Verbindung zwischen dem Outbound Enabler und dem

rsoe-Optionen	Beschreibung
	Backend-Server bereit. Für den Backend-Server ist gegenseitige Authentifizierung erforderlich. Siehe „identity_password“ [*MobiLink - Clientadministration*]. • **trusted_certificates** Eine Datei mit einer Liste von vertrauenswürdigen Stammzertifikaten. Wenn der Backend-Server, und nur der Backend-Server, überprüft werden soll, setzen Sie diese Eigenschaft auf **backend_server_public_cert_filename**. `trusted_certificates=backend_server_public_cert_filename` Wenn unter Windows **trusted_certificates** nicht festgelegt ist, wird der Zertifikatsspeicher des Betriebssystems verwendet. Siehe „trusted_certificates“ [*MobiLink - Clientadministration*].
-d *seconds*	Gibt an, wie häufig Verfügbarkeits-Pings und Statusanforderungen an den Backend-Server gesendet werden. Der Standardwert beträgt 5 Sekunden.
-dl	Verwenden Sie diese Option, um Logmeldungen in der Relay Server Outbound Enabler-Konsole anzuzeigen. Standardmäßig werden Logmeldungen für die Ausführlichkeitsstufen 1 und 2 nicht angezeigt.
-f *farm*	Gibt den Namen der Farm an, zu der der Backend-Server gehört.
-id *id*	Gibt den Namen an, der dem Backend-Server zugewiesen wird.
-o*file*	Gibt die Datei zur Protokollierung von Ausgabemeldungen an.
-oq	Verhindert, dass das Fehlerfenster angezeigt wird, falls beim Start ein Fehler auftritt.
-os *size*	Setzt die Maximalgröße der Meldungslogdateien. Die minimale Größenbegrenzung beträgt 10 kB.
-ot *file*	Kürzt die angegebene Logdatei und schreibt anschließend Meldungen hinein.
-q	Bewirkt die Ausführung beim Start in einem minimierten Fenster.
-qc	Schließt das Fenster nach Abschluss des Vorgangs.
-s	Stoppt den Outbound Enabler.

rsoe-Optionen	Beschreibung
-t *token*	Gibt den Sicherheitstoken an, der an den Relay Server übergeben werden muss.
-uc	Startet den RSOE im Shell-Modus. Dies ist die Standardeinstellung. Gilt für Unix und Mac OS X. Sie können nur jeweils eine der Optionen -uc, -ui, -um oder -ux angeben. Wenn Sie die Option -uc angeben, wird der RSOE auf dieselbe Weise gestartet wie bei früheren Versionen der Software.
-ud	Weist den RSOE an, als Daemon abzulaufen. Diese Option gilt nur für UNIX-Plattformen.
-ui	Startet den RSOE im Shell-Modus, wenn keine verwendbare Anzeige verfügbar ist. Diese Option wird unter Linux mit X Window-Serverunterstützung verwendet. Wenn die Option -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 RSOE im Shell-Modus.
-ux	Unter Linux wird das Meldungsfenster des RSOE geöffnet, in dem Meldungen angezeigt werden. Wenn die Option -ux angegeben ist, muss der RSOE 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 RSOE fehl. Um das Meldungsfenster des RSOE im dialogfreien Modus auszuführen, verwenden Sie -q. Unter Windows erscheint das Meldungsfenster des RSOE automatisch.

rsoe-Optionen	Beschreibung
-v *level*	Legt die Ausführlichkeitsstufe für die Protokollierung fest. *level* kann **0**, **1**, **2** oder höher sein (wobei höhere Stufen hauptsächlich vom technischen Support verwendet werden): ○ **0** Nur Fehler protokollieren. Verwenden Sie diese Protokollierungsstufe für das Deployment. ○ **1** Protokollierung auf Sitzungsebene. Dies ist eine Ansicht einer Synchronisationssitzung von einer höheren Ebene. ○ **2** Anforderungsprotokollierung. Bietet eine detailliertere Ansicht der HTTP-Anforderungen. ○ **3 oder höher** Ausführliche Protokollierung. Dies wird in erster Linie für den technischen Support verwendet. Die Stufen 1 und 2 werden nur die in die Logdatei geschrieben und nicht angezeigt. Um alle Logmeldungen anzuzeigen, verwenden Sie den Schalter -dl.

Dienstprogramm zum Verschleiern von Dateien (dbfhide)

Das Dienstprogramm zum Verschleiern von Dateien (dbfhide) verwendet die einfache Verschlüsselung, um den Inhalt von Konfigurations- und Initialisierungsdateien zu verschleiern.

Syntax

dbfhide *original-configuration-file encrypted-configuration-file*

Option	Beschreibung
original-configuration-file	Gibt den Namen der Originaldatei an.
encrypted-configuration-file	Gibt einen Namen für die neue verschleierte Datei an.

Der Relay Server und der Outbound Enabler erkennen, dass eine Konfigurationsdatei mithilfe von dbfhide verschleiert wurde, und verarbeiten sie.

Dieses Dienstprogramm akzeptiert nicht den @data-Parameter zum Einlesen von Optionen aus einer Konfigurationsdatei.

Integrierter Outbound Enabler (empfohlen für MobiLink)

Mit dem neuen **oe**-Protokoll für die Option -x für mlsrv16 können Sie einen integrierten Outbound Enabler anstelle der Standalone-Version des Outbound Enablers verwenden, der mit dem **rsoe**-Befehl aufgerufen wird. Der integrierte Outbound Enabler bietet folgende Vorteile:

- Verminderte Verwendung von Systemressourcen, vor allem Sockets.

- Bereitstellung einer einzelnen, integrierten Logdatei. Zeilen, die aus dem integrierten Outbound Enabler in das Log des MobiLink-Servers ausgegeben wurden, haben das Präfix **<OE>**.
- Deployment wird vereinfacht.
- Verfügbarkeitsprüfungen zwischen den Outbound Enabler und dem MobiLink-Server entfallen.

Weitere Hinweise zur Verwendung des integrierten Outbound Enablers finden Sie unter „mlsrv16-Option -x“ [*MobiLink - Serveradministration*].

Deployment-Hinweise

Beachten Sie bei der Verwendung des Outbound Enablers folgende Hinweise:

- **Outbound Enabler als Dienst** Der Outbound Enabler kann mithilfe des Dienstprogramms für Dienste als Dienst eingerichtet und verwaltet werden.
- **Authentifizierung** Sie können keine einfache oder Digest-Authentifizierung verwenden. Die Programmdatei *rsoe.exe* unterstützt keine einfache oder Digest-Authentifizierung mit Webservern, unabhängig vom Webservertyp oder dem Betriebssystem.

Siehe auch

- „Outbound Enabler als Dienst“ auf Seite 48

Outbound Enabler als Dienst

Der Outbound Enabler kann mithilfe des Dienstprogramms für Dienste (dbsvc) als Dienst gestartet werden. Mithilfe des Dienstprogramms für Dienste können Dienste erstellt, geändert und gelöscht werden. Um eine vollständige Liste der Syntaxinformationen anzuzeigen, führen Sie *dbsvc* ohne Optionen aus.

So richten Sie einen automatisch gestarteten RSOE-Dienst mit dem Namen oes (Outbound Enabler-Dienst) unter Windows ein

```
dbsvc -as -s auto -t rsoe -w oes "%SQLANY16%\BinXX\rsoe.exe"
-cr "host=relayserver.sybase.com;port=80 " -cs "host=localhost;port=80 " -f
FarmName -id ServerName -t token
```

So richten Sie einen automatisch gestarteten RSOE-Dienst mit dem Namen oes (Outbound Enabler-Dienst) unter Unix ein

```
dbsvc -y -a <some-user-account> -t rsoe -w oes @/<full-dir-path>/oe.config
```

Bemerkungen

Die Syntax von dbsvc unter Windows unterscheidet sich von der unter Unix. In Unix geben Sie nicht den vollständigen Pfad der Programmdatei als ersten Parameter nach der Parameteroption -w an.

Verwenden Sie nur vollständige Pfade.

Geben Sie unter Unix die Outbound Enabler-Parameter nur in einer Befehlsdatei an. Verwenden Sie die Befehlszeilenoptionen nicht im Befehl setup dbsvc.

Starten des Diensts

```
dbsvc.exe -u oes
```

So stoppen Sie den Dienst

```
dbsvc.exe -x oes
```

So deinstallieren Sie den Dienst

```
dbsvc.exe -d oes
```

Siehe auch

- „SQL Anywhere-Webdienste mit Hochverfügbarkeit und Scale-Out-Lösungen“ [*SQL Anywhere Server - Datenbankadministration*]

Konfigurationsaktualisierungen für die Relay Server-Farm

Die Konfiguration einer Relay Server-Farm wird vom Inhalt der Server-Konfigurationsdatei festgelegt. Alle Relay Server in einer Relay Server-Farm verwenden dieselbe Relay Server-Konfigurationsdatei. Wenn Sie die Konfiguration einer Relay Server-Farm aktualisieren, müssen Sie daher die Relay Server-Konfigurationsdatei auf jedem Relay Server in der Farm aktualisieren. Eine Aktualisierung kann folgende Änderungen umfassen:

- Der Relay Server-Farm einen neuen Relay Server hinzufügen
- Eine neue Backend-Serverfarm erstellen und ihr Zugriff auf die Relay Server-Farm gewähren
- Einen neuen Backend-Server zu einer vorhandenen Backend-Serverfarm hinzufügen
- Die Eigenschaften eines Relay Servers, einer Backend-Serverfarm oder eines Backend-Servers ändern
- Optionen ändern

Eine Möglichkeit, eine Relay Server-Konfiguration zu ändern, besteht darin, alle Relay Server herunterzufahren, die Relay Server-Konfigurationsdatei durch die aktualisierte Version zu ersetzen und alle Relay Server neu zu starten. Das Herunterfahren und Neustarten der Relay Server bedeutet, dass es für die Benutzer des Relay Servers möglicherweise zu einer Unterbrechung des Dienstes kommt.

Die bevorzugte Methode zur Aktualisierung einer Relay Server-Konfiguration ist, mithilfe des Relay Server-Status-Managers die Konfiguration zu aktualisieren, während eine Relay Server-Farm läuft, ohne den Dienst zu unterbrechen.

Eine Relay Server-Konfiguration wird aktualisiert, indem eine neue Instanz des Relay Server-Status-Managers mit dem folgenden Befehlszeilenformat gestartet wird:

```
rshost -u -f filename
```

Die Option –u weist den Relay Server-Status-Manager an, einen Aktualisierungsvorgang auszuführen. Mit der Option –f wird der Name der Konfigurationsdatei angegeben, die die aktualisierte Konfiguration enthält.

Im folgenden Überblick werden die Schritte beschrieben, die zur Aktualisierung der Konfiguration einer Relay Server-Farm erforderlich sind.

1. Nehmen Sie die Änderungen an der Master-Kopie der Relay Server-Konfigurationsdatei vor.
2. Führen Sie auf jedem Computer, auf dem eine Instanz eines Relay Servers ausgeführt wird, der zur aktualisierten Relay Server-Farm gehört, folgende Aufgaben aus:
    a. Ersetzen Sie die alte Konfigurationsdatei durch die aktualisierte Konfigurationsdatei.
    b. Führen Sie den Relay Server-Status-Manager mit der aktualisierten Konfigurationsdatei aus.

Siehe auch

- „Relay Server-Status-Manager“ auf Seite 25

Relay Server-Konfiguration für Microsoft IIS unter Windows aktualisieren

Möglicherweise müssen Sie gelegentlich Relay Server-Konfigurationsdateien aktualisieren, um Relay Server oder Relay Server-Farmen hinzuzufügen oder zu ändern sowie Eigenschaften und Optionen für Server und Farmen zu ändern.

Voraussetzungen

Eine Relay Server-Konfigurationsdatei für eine vorhandene Relay Server-Farm.

Aufgabe

1. Kopieren Sie für jeden Computer, der zur aktualisierten Relay Server-Farm gehört, die aktualisierte Konfigurationsdatei in das Verzeichnis *%SQLANY16%\RelayServer\IIS\BinXX\Server*, das sich unter dem Stammverzeichnis der Relay Server-Website befindet.
2. Führen Sie im Verzeichnis *%SQLANY16%\RelayServer\IIS\BinXX\Server* die folgende Anweisung in der Befehlszeile aus, um die Aktualisierung der Konfiguration zu übernehmen:

   ```
   rshost -u -f rs.config
   ```

3. Wiederholen Sie die oben beschriebenen Schritte für jeden Computer in der zu aktualisierenden Relay Server-Farm.

Ergebnisse

Die Relay Server-Konfiguration wird aktualisiert.

Relay Server-Konfiguration für Apache unter Linux aktualisieren

Möglicherweise müssen Sie gelegentlich Relay Server-Konfigurationsdateien aktualisieren, um Relay Server oder Relay Server-Farmen hinzuzufügen oder zu ändern sowie Eigenschaften und Optionen für Server und Farmen zu ändern.

Voraussetzungen

Eine Relay Server-Konfigurationsdatei für eine vorhandene Relay Server-Farm.

Aufgabe

1. Kopieren Sie die aktualisierte Konfigurationsdatei in das Verzeichnis */modules* unter dem Apache-Installationsverzeichnis.

2. Führen Sie im Verzeichnis */Apache-install/modules* die folgende Anweisung in der Befehlszeile aus, um die Aktualisierung der Konfiguration zu übernehmen:

   ```
   rshost -u -f rs.config
   ```

3. Wiederholen Sie die oben beschriebenen Schritte für jeden Computer in der zu aktualisierenden Relay Server-Farm.

Ergebnisse

Die Relay Server-Konfiguration wird aktualisiert.

Relay Server-Plug-In für Sybase Central

Das Relay Server-Plug-In für Sybase Central bietet eine einfache Möglichkeit für die Arbeit mit dem Relay Server. Verwenden Sie das Relay Server-Plug-In für folgende Aufgaben:

- Erstellung, Import und Deployment der Relay Server-Konfigurationsdateien
- Anzeigen der Eigenschaften der Relay Server-Konfigurationsdatei
- Hinzufügen von Relay Servern, Relay Server-Farmen, Backend-Servern und Backend-Serverfarmen
- Anzeigen und Bearbeiten von Relay Servern, Relay Server-Farmen, Backend-Servern und Backend-Serverfarmen

Mit Relay Server-Konfigurationsdateien arbeiten

Sie können Sybase Central verwenden, um mit Relay Server-Konfigurationsdateien zu arbeiten. In Sybase Central können Sie Folgendes durchführen:

- Eine Relay Server-Konfigurationsdatei erstellen
- Eine Relay Server-Konfigurationsdatei öffnen
- Eine Relay Server-Konfigurationsdatei importieren
- Eine Relay Server-Konfigurationsdatei bereitstellen

Einer Relay Server-Konfigurationsdatei erstellen

Erstellen Sie eine Relay Server-Konfigurationsdatei, um eine Relay Server-Farm sowie die Backend-Serverfarmen für die Verbindungen mit der Relay Server-Farm zu definieren.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Rechtsklicken Sie in der Ansicht **Ordner** von Sybase Central auf **Relay Server 16** und klicken Sie auf **Neu** » **Konfigurationsdatei**.
2. Wechseln Sie auf dem Computer, auf dem Sybase Central ausgeführt wird, zu dem Verzeichnis, in dem Sie die Konfigurationsdatei speichern möchten. Dies ist nicht identisch mit dem Speicherort für das Deployment.
3. Geben Sie in Feld **Dateiname** den Namen der Konfigurationsdatei ein. Normalerweise ist dies *rs.config*.

4. Vergewissern Sie sich, dass die Erweiterung *.config* im Feld **Dateityp** ausgewählt ist.
5. Klicken Sie auf **Speichern**.

Ergebnisse

Eine Relay Server-Farm wird automatisch erstellt.

Nächste Schritte

Sie können nun die erforderlichen Relay Server und Backend-Server hinzufügen.

Eine Relay Server-Konfigurationsdatei öffnen

Öffnen Sie eine Relay Server-Konfigurationsdatei, um die Konfigurationseinstellungen zu bearbeiten.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Rechtsklicken Sie in der Ansicht **Ordner** von Sybase Central auf **Relay Server 16** und klicken Sie auf **Konfigurationsdatei öffnen**.
2. Wechseln Sie zu dem Verzeichnis, in dem sich die Konfigurationsdatei befindet, rechtsklicken Sie auf die Datei und klicken Sie auf **Öffnen**.

Ergebnisse

Die Konfigurationsdatei wird geöffnet.

Nächste Schritte

Sie können nun die Datei bearbeiten.

Eine Relay Server-Konfigurationsdatei importieren

Importieren Sie eine Relay Server-Konfigurationsdatei, um Einstellungen aus einer anderen Relay Server-Einrichtung zu verwenden.

Voraussetzungen

Die Konfigurationsdatei muss bereits vorhanden sein.

Kontext und Bemerkungen

Wenn der Relay Server HTTPS-Kommunikation voraussetzt, muss das Stammzertifikat für den Server im Java-Dienstprogramm für die Schlüssel- und Zertifikatsverwaltung abgelegt werden. Verwenden Sie

hierzu das Java Keytool-Dienstprogramm. Sybase Central greift auf das Java-Dienstprogramm für die Schlüssel- und Zertifikatsverwaltung zu, wenn das Stammzertifikat für die Kommunikation benötigt wird.

Aufgabe

1. Rechtsklicken Sie in der Ansicht **Ordner** von Sybase Central auf **Relay Server 16** und klicken Sie auf **Konfigurationsdatei importieren**.
2. Geben Sie die URL für den vorhandenen Relay Server ein.
3. Wenn der Relay Server eine Authentifizierung erfordert, geben Sie **Benutzername** und **Kennwort** ein und klicken Sie auf **OK**.

Ergebnisse

Die Relay Server-Konfiguration wird importiert.

Nächste Schritte

Sie können nun die Datei bearbeiten.

Eine Relay Server-Konfigurationsdatei bereitstellen

Führen Sie das Deployment einer Relay Server-Konfigurationsdatei durch, um Ihre Relay Server-Einrichtung zu konfigurieren

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Kontext und Bemerkungen

Wenn der Relay Server HTTPS-Kommunikation voraussetzt, muss das Stammzertifikat für den Server im Java-Dienstprogramm für die Schlüssel- und Zertifikatsverwaltung abgelegt werden. Verwenden Sie hierzu das Java Keytool-Dienstprogramm. Sybase Central greift auf das Java-Dienstprogramm für die Schlüssel- und Zertifikatsverwaltung zu, wenn das Stammzertifikat für die Kommunikation benötigt wird.

Aufgabe

1. Rechtsklicken Sie in der Ansicht **Ordner** auf die Relay Server-Konfigurationsdatei, die Sie bereitstellen möchten, und klicken Sie auf **Bereitstellen**.
2. Geben Sie die URL für den Relay Server ein.
3. Wenn der Relay Server eine Authentifizierung erfordert, geben Sie **Benutzername** und **Kennwort** ein und klicken Sie auf **OK**.
4. Die Seite **Serverliste** zeigt vorhandene Relay Server. Um das Deployment der Konfigurationsdatei auf einem oder mehreren der Relay Server durchzuführen, wählen Sie die Server aus der Liste und klicken auf **Hinzufügen**.

Ergebnisse

Die Relay Server-Konfiguration wird bereitgestellt.

Um einen Relay Server aus der Liste zu entfernen, wählen Sie den Server aus und klicken Sie auf **Entfernen**.

Relay Server und Relay Server-Farmen verwalten

Mit Sybase Central können Sie Relay Server und Relay Server-Farmen verwalten. In Sybase Central können Sie Folgendes durchführen:

- Relay Server einer Relay Server-Farm hinzufügen
- Eigenschaften von Relay Servern anzeigen oder bearbeiten
- Eigenschaften von Relay Server-Farmen anzeigen oder bearbeiten

Relay Server zu einer Farm hinzufügen

Sie können einen oder mehrere Relay Server in einer Relay Server-Farm ausführen.

Voraussetzungen

Die Relay Server-Farm muss bereits vorhanden sein.

Aufgabe

1. Rechtsklicken Sie im Fensterausschnitt **Ordner** unter der gewünschten Konfigurationsdatei auf die Relay Server-Farm, der Sie Relay Server hinzufügen möchten, und klicken Sie auf **Neu** » **Relay Server**.
2. Vergewissern Sie sich, dass die Option **Diesen Relay Server aktivieren** ausgewählt ist.
3. Geben Sie die Pfadinformationen für den **Host** ein, mit dem Sie eine Verbindung herstellen wollen. Klicken Sie auf **Ping**, um zu prüfen, ob eine Verbindung mit dem angegebenen Host hergestellt werden kann.
4. Wählen Sie das zu verwendende Kommunikationsprotokoll. Dieser Wert kann **HTTP** oder **HTTPS** sein.
5. Geben Sie die Ports an, die für die ausgewählte Protokolle verwendet werden sollen.
6. Falls gewünscht, geben Sie eine Beschreibung des Relay Servers in das Feld **Beschreibung** ein.
7. Klicken Sie auf **Übernehmen**, um mit dem Hinzufügen von Relay Servern fortzufahren, oder auf **OK**, um den Relay Server hinzuzufügen und das Fenster **Relay Server erstellen** zu schließen.

Ergebnisse

Der Relay Server wird der Farm hinzugefügt.

Nächste Schritte

Sie können nun die Eigenschaften von Relay Servern anzeigen oder bearbeiten, um die Relay Server-Einstellungen zu ändern.

Eigenschaften von Relay Servern anzeigen oder bearbeiten

Zeigen Sie die Eigenschaften von Relay Servern an oder bearbeiten Sie sie, um die Relay Server-Einstellungen zu ändern.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Klicken Sie im Fensterausschnitt **Ordner** auf die Relay Server-Farm mit dem gewünschten Relay Server. Die Relay Server in dieser Farm werden im rechten Fensterausschnitt aufgeführt.
2. Rechtsklicken Sie auf den Relay Server, den Sie bearbeiten oder anzeigen möchten, und klicken Sie auf **Eigenschaften**.
3. Nehmen Sie die erforderlichen Änderungen an den Eigenschaften des Relay Servers vor und klicken Sie auf **Übernehmen** oder **OK**.

Ergebnisse

Die Änderungen werden auf den Relay Server angewendet.

Nächste Schritte

Sie können die Eigenschaften von Relay Servern anzeigen oder bearbeiten, um die Einstellungen der Relay Server-Farm zu ändern.

Eigenschaften von Relay Server-Farmen anzeigen oder bearbeiten

Zeigen Sie die Eigenschaften von Relay Servern an oder bearbeiten Sie sie, um die Relay Server-Farmeinstellungen zu ändern.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Rechtsklicken Sie im Fensterausschnitt **Ordner** auf die Relay Server-Farm, mit der Sie arbeiten möchten, und klicken Sie auf **Eigenschaften**.
2. Nehmen Sie die erforderlichen Änderungen an den Eigenschaften der Relay Server-Farm vor und klicken Sie auf **Übernehmen** oder **OK**.

Ergebnisse

Alle Änderungen der Eigenschaften der Relay Server-Farm werden gespeichert.

Backend-Server und Backend-Serverfarmen verwalten

Mit Sybase Central können Sie Backend-Server und Backend-Serverfarmen verwalten. In Sybase Central können Sie Folgendes durchführen:

- Eine Backend-Serverfarm erstellen und ihr Backend-Server hinzufügen
- Eigenschaften von Backend-Servern anzeigen oder bearbeiten
- Eigenschaften von Backend-Serverfarmen anzeigen oder bearbeiten

Backend-Serverfarm erstellen

Erstellen Sie eine Backend-Serverfarm. Eine Backend-Serverfarm ist eine Gruppe gleichartiger Backend-Server, an die Clients über die Relay Server-Farm Anforderungen absetzen.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Rechtsklicken Sie im linken Fensterausschnitt auf die Backend-Server-Konfigurationsdatei, mit der Sie arbeiten möchten, und klicken Sie auf **Neu** » **Backend-Serverfarm**.
2. Vergewissern Sie sich, dass die Option **Diese Backend-Serverfarm aktivieren** ausgewählt ist.
3. Geben Sie den Namen ein, der der neuen Backend-Serverfarm zugeordnet ist.
4. Wählen Sie unter **Client-Sicherheit** ein Protokoll, dass Clients für Verbindungen mit der Backend-Serverfarm verwenden sollen.
5. Wählen Sie unter **Backend-Sicherheit** ein Protokoll, dass der Relay Server Outbound Enabler (RSOE) für Verbindungen mit der Backend-Serverfarm verwenden soll.

6. Um die Client-Server-Affinität zu bewahren, wählen Sie den Servertyp aus, den Sie verwenden. Klicken Sie bei einem MobiLink-HTTP-Server mit Standalone-Outbound Enabler oder einem MobiLink-Server mit eingebettetem Outbound Enabler auf **MobiLink**. Klicken Sie bei einem typischen SQL Anywhere-Webdienst auf **SQL Anywhere**. Erweiterte benutzerdefinierte Einstellungen sind verfügbar, wenn Sie auf den Servertyp **Benutzerdefiniert** klicken. Wenn der Servertyp **Benutzerdefiniert** ausgewählt ist, haben Sie die vollständige Kontrolle über die folgenden Affinitätseinstellungen:
   a. Prüfen Sie die Option **Aktives Cookie**, wenn Sie möchten, dass der Relay Server den Standard-HTTP-Befehl set-cookie verwendet, um die Client-Server-Affinität zu bewahren.
   b. Prüfen Sie die Option **Aktiver Header**, wenn Sie möchten, dass der Relay Server einen systemeigenen Header verwendet, um die Client-Server-Affinität zu bewahren.
7. Geben Sie optional im Feld **Beschreibung** eine Beschreibung der Backend-Serverfarm ein.
8. Klicken Sie auf **Übernehmen**, wenn Sie mit dem Hinzufügen von Backend-Serverfarmen fortfahren möchten, oder auf **OK**, um die Backend-Serverfarm hinzuzufügen und das Fenster **Backend-Serverfarm erstellen** zu schließen.

Ergebnisse

Die Backend-Serverfarm wird erstellt.

Nächste Schritte

Sie können Eigenschaften von Backend-Serverfarmen anzeigen oder bearbeiten, um die Einstellungen der Serverfarm zu verwalten.

Siehe auch

„Affinität“ auf Seite 6

Eigenschaften von Backend-Serverfarmen anzeigen oder bearbeiten

Sie können Eigenschaften von Backend-Serverfarmen anzeigen oder bearbeiten, um die Einstellungen der Serverfarm zu verwalten.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Rechtsklicken Sie im Fensterausschnitt **Ordner** auf die Backend-Serverfarm, mit der Sie arbeiten möchten, und klicken Sie auf **Eigenschaften**.
2. Nehmen Sie die erforderlichen Änderungen an den Eigenschaften der Backend-Serverfarm vor und klicken Sie auf **Übernehmen** oder **OK**.

Ergebnisse

Ihre Änderungen der Eigenschaften des Backend-Servers werden gespeichert.

Nächste Schritte

Sie können Servereigenschaften anzeigen oder bearbeiten, um das Verhalten des Backend-Servers zu ändern.

Server zu einer Backend-Serverfarm hinzufügen

Fügen Sie Server zu einer Backend-Serverfarm hinzu, um die Arbeitsauslastung zu verwalten.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Rechtsklicken Sie im Fensterausschnitt **Ordner** auf die Backend-Serverfarm, mit der Sie arbeiten möchten, und klicken Sie auf **Neu** » **Backend-Server**.
2. Vergewissern Sie sich, dass die Option **Diesen Backend-Server aktivieren** ausgewählt ist.
3. Geben Sie den Namen ein, der dem neuen Backend-Server zugeordnet ist.
4. Um die Prüfung der MAC-Adresse zu erzwingen, aktivieren Sie das Kontrollkästchen **MAC-Adressprüfung erzwingen**.
5. Wenn Sie die MAC-Adressprüfung ausgewählt haben, geben Sie die MAC-Adresse des RSOE im Format IEEE 802 MAC-48 ein. Das richtige Format der MAC-Adresse entnehmen Sie der Konsole oder dem Log des Relay Server Outbound Enablers. Mehrere MAC-Adressen, die durch Ausrufezeichen (!) voneinander getrennt sind, werden vom Outbound Enabler gemeldet, falls auf Ihrem Backend-Servercomputer mehrere Adapter aktiv sind. Wählen Sie den am längsten bestehenden für die Prüfung durch den Relay Server. Der Befehl `ipconfig /all` unter Windows liefert eine detaillierte Liste Ihrer Netzwerkadapter zusammen mit den zugeordneten MAC-Adressen.
6. Geben Sie den Sicherheitstoken an, der vom Relay Server zum Authentifizieren der Backend-Serververbindung verwendet wird. Sie können eine maximale Länge von 2048 Zeichen verwenden.
7. Falls gewünscht, geben Sie im Feld **Beschreibung** eine Beschreibung des Backend-Servers ein.
8. Klicken Sie auf **Übernehmen**, wenn Sie mit dem Hinzufügen von Backend-Servern fortfahren möchten, oder auf **OK**, um den Backend-Server hinzuzufügen und das Fenster **Backend-Server erstellen** zu schließen.

Ergebnisse

Die Server werden der Backend-Serverfarm hinzugefügt.

Nächste Schritte

Sie können die Eigenschaften von Backend-Servern anzeigen oder bearbeiten, um die Einstellungen Ihrer Relay Server-Farm zu ändern.

Eigenschaften von Backend-Servern anzeigen oder bearbeiten

Zeigen Sie die Servereigenschaften an oder ändern Sie sie, um das Verhalten des Backend-Servers zu ändern.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Klicken Sie im Fensterausschnitt **Ordner** auf die Backend-Serverfarm mit dem gewünschten Backend-Server. Die Backend-Server in dieser Farm werden im rechten Fensterausschnitt aufgeführt.
2. Rechtsklicken Sie auf den Backend-Server, den Sie bearbeiten oder anzeigen möchten, und klicken Sie auf **Eigenschaften**.
3. Nehmen Sie die erforderlichen Änderungen an den Eigenschaften des Backend-Servers vor und klicken Sie auf **Übernehmen** oder **OK**.

Ergebnisse

Die Änderungen an den Eigenschaften des Backend-Servers werden gespeichert.

Protokollierung und Administration des Relay Servers

Die Logdatei des Relay Servers zeigt folgende Arten von Meldungen:

- **Information** Hier werden Basisinformationen über die aktuelle Sitzung angezeigt.
- **Warning** Hier werden Warnungen über aufgetretene Vorgänge angezeigt.
- **Error** Hier werden Fehlermeldungen über fehlgeschlagene Vorgänge angezeigt.

Der Standard-Logdateiname und das Verzeichnis für Windows lauten *%temp%**ias_relay_server_host.log***. Um den Namen der Logdatei zu ändern, verwenden Sie die Option **rshost -o** *filename*.

Die Relay Server-Protokollierung unterstützt außerdem die folgenden Funktionen:

- Die Logdateien von Relay Server und Outbound Enabler protokollieren mit Zeitstempeln in einer Millisekundenauflösung.

  Zeitstempel werden im lokalen differentialen Zeitstempelformat berichtet (+/- hhmm).

- Beim Verarbeiten von HTTP-Anforderungen mit SAP Passport-Header erhöhen Relay Server und Outbound Enabler die Ausführlichkeitsstufe der Anforderungsverarbeitung entsprechend der erforderlichen Protokollierungsstufe, die im Passport enthalten ist, und fügen außerdem den zugeordneten Logzeilen einen Suffix hinzu, um die Schlüsselinformationen des Passports aufzulisten.

Verbosity

Sie können die Ausführlichkeit für den Relay Server auf folgende Stufen setzen:

- **0** Nur Fehler protokollieren. Verwenden Sie diese Protokollierungsstufe für das Deployment. Dies ist die Standardeinstellung.
- **1** Anforderungsprotokollierung. Übersichten von HTTP-Anforderungen werden in abgekürzter Form in die Logdatei geschrieben.
- **2** Anforderungsprotokollierung. Bietet eine detailliertere Ansicht der HTTP-Anforderungen.
- **3 oder höher** Ausführliche Protokollierung. Dies wird in erster Linie für den technischen Support verwendet.

Fehler werden unabhängig von der Protokollstufe angezeigt, während Warnungen nur bei einer Protokollstufe von 1 oder höher angezeigt werden.

Siehe auch

„Befehlszeilensyntax des Relay Server Status-Managers (rshost)“ auf Seite 26

Relay Server-Protokollierung und SAP Passports

SAP Passports werden verwendet, um Anforderungen vom Client bis zum Backend-Server zu protokollieren. Der Passport kann eine Direktive enthalten, mit der die Ausführlichkeit der Protokollierung beim Durchfluss durch den Server erhöht wird. Sowohl der Relay Server als auch der Relay Server Outbound Enabler befolgen diese Direktive. In Fällen, in denen die Ausführlichkeitsstufe des Relay Server Outbound Enablers höher als die Stufe gesetzt ist, die von Passport impliziert wird, ist die höhere Stufe wirksam. Dies bedeutet, dass ein Benutzer keinen Passport mit einer niedrigen Protokollierungsstufe verwenden kann, um Nutzlastprotokollierung zu unterdrücken, wenn der Relay Server-/Outbound Enabler-Administrator in der Relay Server-/Outbound Enabler-Konfiguration eine Protokollierung mit hoher Ausführlichkeit eingestellt hat.

SAP Passport Version 2 und 3 werden unterstützt.

Passport-Protokollierungsstufe	RS/OE-Ausführlichkeitsstufe	Beschreibung
Niedrig	1	Protokollierung auf Zugriffsebene
Mittel	4	Debug-Protokollierung mit Paket-Protokollierung (plus Anforderungs-Header-Protokollierung auf der Relay Server-Seite)
Hoch	5	Debug-Protokollierung mit gesamter Nutzlast

Der Relay Server und der Outbound Enabler protokollieren Zeilen, die mit einer Anforderung verknüpft sind, an der ein SAP Passport mit Suffixen beteiligt ist, wie sie in der folgenden Tabelle aufgelistet sind:

SAP Passport-Version	Logzeilen-Suffix	Logzeilen-Beispiel
2	`#SAP-PPK#V2#<Transaction-uuid>`	`I. 2012-07-19 15:06:32.720 <15852.9340.F0B0S5R0> Relaying PPK#V2#8fa46833ea42b94a8181b5bc8da3a33c`
3	`#SAP-PPK#V3#<Transaction-uuid>#<Root-context-uuid>#<Connection-uuid>#<Connection-counter>`	`I. 2012-07-19 15:06:32.909 <15852.14180.F0B0S6R0> Relaying PPK#V3#a65762116e4e48b6b1fccdc428785d49#dd880e77415f4811bd`

Relay Server Record

Der Relay Server Record (RSR) besteht aus einer kurzen Zusammenfassung der Relay Server-Verarbeitung und enthält Informationen über Anforderungen, Timing, Affinität, Anforderungsstatus und Datenträger. Sie können den RSR verwenden, um Relay Server-Ausfälle zu diagnostizieren und Performance-Eigenschaften zu studieren.

Der Relay Server Record wird als Teil der Relay Server-Protokollierung generiert, wenn die Ausführlichkeitsstufe auf 1 oder höher gesetzt ist.

Der RSR besteht aus einer einzelnen Zeile in der Relay Server-Logdatei und enthält viele Werte, die eine HTTP-Anforderung zusammenfassen. Zur Unterstützung beim Interpretieren von Relay Server Records enthält die Relay Server-Logdatei einen Header, der die RSR-Werte beschreibt. Die Header-Zeile besteht aus Symbolen, die auch in der Datei beschrieben werden. Für die Symbole gilt die folgende Konvention:

Symbol	Datentyp
b:	Bytezähler
c:	Anderer Zähler
i:	ID oder nummerischer Code
m:	Zeit gemessen in Millisekunden
x:	Hexadezimalzahl
name:*string*	Benannte Zeichenfolgenwerte variabler Länge Dazu kann Folgendes gehören: Relay Server-Fehlernamen, Relay Server-Fehlerparameter und SAP Passport-Informationen.
oe	Vom Outbound Enabler gemeldetes Element
up	Element, das mit der Anforderung (außer der Antwort) vom Relay Server an den Backend-Server verknüpft ist
rtp	Element, das mit Roundtrip-Transport und der Verarbeitung des letzten Aufwärts- und ersten Abwärtspakets verknüpft ist
dn	Element, das mit der Anforderung (außer der Antwort) vom Backend an den Relay Server verknüpft ist
in	Verstrichene Zeit beim Warten auf Eingabe zum Lesen
out	Verstrichene Zeit beim Warten auf Eingabe beim Schreiben
A.B	Diese Notierung gibt an, dass B eine untergeordnete Komponente, ein Unterprozess oder ein Aspekt von A ist.

Symbol	Datentyp
pkt/packet	Bezieht sich auf das erstellte Paket, das vom Relay Server oder vom Outbound Enabler für die Kommunikation über den Aufwärts-/Abwärts-Kanal erstellt wurde.

Die Symbole sind zusammengesetzt (als Feldnamen), um auf bestimmte Arten von Daten zu verweisen. Beispiel: **m:up.out** entspricht der Zeit (**m:**), die innerhalb des Relay-Zeitraums im Aufwärts-Kanal mit dem (**up**) Schreiben von Paketen (**.out**) verbracht wurde. Die Punktnotation (.) zeigt auch an, dass **out** ein Unterprozess **up** ist.

Feldname	Datentyp
x:sfp	Sitzungsfingerabdruck (eine Komponente von Affinitätsdaten)
i:oe.sidx	Sitzungsindex, der vom Outbound Enabler zugewiesen wurde
flag[0]	Affinitätsentscheidung; Werte: n = Neu, c = Fortgesetzt, h = Zugeordnet, r = Erneuert. *Neu* steht für eine neue Affinität; *Fortgesetzt* zeigt eine folgende Anforderung in einer eingerichteten Affinitätssitzung an; *Zugeordnet* steht für eine neue Affinität mit dem designierten Backend-Server; *Erneuert* bedeutet, dass es eine Kollision gibt, sodass die Affinitätsdaten erneuert werden und ein neuer Abschnitt geöffnet wird.
flag[1]	Anforderungsbeständigkeit; Werte: p = Beständig, n = Nicht beständig
flag[2]	Kodierung der Anforderungsübertragung; Werte: k = In Abschnitte aufgeteilt, l = Inhaltslänge
flag[3]	Antwortbeständigkeit; Werte: p = Beständig, n = Nicht beständig, u = Unbekannt
flag[4]	Kodierung der Antwortübertragung; Werte: k = In Abschnitte aufgeteilt, l = Inhaltslänge, u = Unbekannt
b:up	Anforderungsgröße in Byte
b:dn	Antwortgröße in Byte
c:up:pkt	Anzahl von Aufwärts-Anforderungspaketen, die an den Outbound Enabler gesendet wurden
m:up	Anforderungs-Relay-Zeitraum mit überlappenden Lesevorgängen vom Client, Schreibvorgänge auf dem Aufwärts-Kanal und Paketbildung für Anforderungen
m:up.in	Summe der verstrichenen Zeit beim Warten auf Anforderungsnutzlast vom Client innerhalb des Anforderungs-Relay-Zeitraums
m:up.out	Summe der verstrichenen Zeit beim Schreiben von Paketen auf dem Aufwärts-Kanal innerhalb des Relay-Zeitraums

Feldname	Datentyp
m:rtp	Roundtrip-Verarbeitungszeitraum vom Senden des letzten Anforderungspakets bis zum Empfang des ersten Antwortpakets zwischen dem Relay Server und dem Backend-Server, einschließlich der Verarbeitungszeit auf dem Backend-Server
m:oe:rtp	Roundtrip-Verarbeitungszeitraum vom Senden des letzten Anforderungspakets bis zum Empfang des ersten Antwortpakets zwischen dem Outbound Enabler und dem Backend-Server, einschließlich der Verarbeitungszeit auf dem Backend-Server
m:rtp:kpi	Roundtrip-Relay-Verarbeitung und Transportzeit des letzten Anforderungspakets und des ersten Antwortpakets zwischen dem Relay Server und dem Outbound Enabler, mit der folgenden Formel berechnet: (m: rtp - m: oe.rtp)
c:dn.pkt	Anzahl der Abwärts-Antwortpakete, die vom Outbound Enabler empfangen wurden
m:dn	Antwort-Relay-Zeitraum mit überlappendem Lesen vom Abwärts-Kanal und Schreibvorgänge auf dem Client
m:dn.in	Summe der verstrichenen Zeit beim Warten auf Antwortpakete aus dem Abwärts-Kanal innerhalb des Antwort-Relay -Zeitraums
m:dn.out	Summe der verstrichenen Zeit beim Warten auf das Schreiben der Antwortnutzlast an den Client innerhalb des Antwort-Relay-Zeitraums
m:oe.dn	Antwort-Empfangszeitraum wie vom Outbound Enabler beobachtet. Dieser Zeitraum überschneidet sich mit der Zeitzählung in m:dn.in und m:dn.out.
m:close	Die verstrichene Zeit zwischen dem Ende von m:dn und dem Beenden der **rs_client**-Erweiterung
b:dn.maxLQ	Höchste Speichernutzung der lokalen Antwortpaket-Warteschlange dieser Anforderung
c:dn.maxSQ	Höchste Anzahl der Antwortpakete in der Antwortpaket-Warteschlange im gemeinsam genutzten Speicher dieser Anforderung
i:dn.stts	HTTP-Antwortstatus
i:err	Fehler-ID
i:warn	Warnungs-ID
m:appTO	Anwendungs-Timeout der Anforderung
err	Name des Fehlers
warn	Name der Warnung
oe.err	Name des Outbound Enabler-Fehlers

Feldname	Datentyp
oe.err.p0	Erster Fehlerparameter vom Outbound Enabler
oe.err.p1	Zweiter Fehlerparameter vom Outbound Enabler
oe.err.p2	Dritter Fehlerparameter vom Outbound Enabler
up.ua	Benutzeragenten-Header der Anforderung
up.uq	URL-Abfrageparameter in *name=value*-Paaren. Dies kann nützlich sein für Tagging-Informationen innerhalb der Anforderung, wenn keine SAP Passports zur Verfügung stehen.
up.AfHdr	Affinität
up.cookie	Cookie-Header in der Anforderung

k.A.	Zusammengesetztes Anforderungspräfix (am Ende jeder Zeile):
	`<processId.threadId.F(BEFarmIdx)B(BEServerIdx)S(BESessionNum)R(RequestIdx)>`
	`<BEFarmName>`
	`<BeServerName>`
	Beispiel:
	`<11436.4592.F0B0S0R0> <RSTEST02.F0> <S0>`

Ein Relay Server Record sieht ähnlich aus wie im folgenden Beispiel:

```
I. 2012-11-09 02:35:11.296-0500 RSR header: x:sfp  i:oe.sidx  flag
b:up       b:dn |  c:up.pkt       m:up    m:up.in   m:up.out |     m:rtp
m:oe.rtp  m:rtp.kpi |  c:dn.pkt       m:dn    m:dn.in   m:dn.out    m:dn.oe
|   m:close | b:dn.maxLQ c:dn.maxSQ | i:dn.stts  i:err i:warn |
m:appTO ...other-variable-length-elements...
I. 2012-11-09 02:35:38.192-0500 RSR row: d9d72990          0 npkpk
1024477    1031252 |       444       2366       2305          0 |
1001        999          2 |       107      17764          0
17741         24 |         0 |     953152         87 |       200      0
0 |     60000 <err:RSE_NO_ERROR> <warn:RSW_NO_WARNING> <oe.err:N/A>
<oe.err.p0:> <oe.err.p1:> <oe.err.p2:> <up.userAgent:RSTestClient> <up.uq:>
<up.AfHdr:> <up.cookie:> <11436.4592.F0B0S0R0> <RSTEST02.F0> <S0>
I. 2012-11-09 02:36:02.278-0500 RSR row: d9d72990          0 cpkpk
1024534    1031211 |       428       2440       2379          0 |
1000        999          1 |       124      17763          0
17741         22 |         0 |     951657        109 |       200      0
0 |     60000 <err:RSE_NO_ERROR> <warn:RSW_NO_WARNING> <oe.err:N/A>
<oe.err.p0:> <oe.err.p1:> <oe.err.p2:> <up.userAgent:RSTestClient> <up.uq:>
<up.AfHdr:ias-rs-sessionid="kCnX2QAAAAAAAAAAUzAA";> <up.cookie:>
<11436.4592.F0B0S0R0> <RSTEST02.F0> <S0>
I. 2012-11-09 02:36:26.515-0500 RSR row: d9d72990          0 cnknk
1024529    1031230 |       400       2440       2379          0 |
1000        999          1 |       153      17924          0
17899         22 |         0 |     942016        114 |       200      0
0 |     60000 <err:RSE_NO_ERROR> <warn:RSW_NO_WARNING> <oe.err:N/A>
<oe.err.p0:> <oe.err.p1:> <oe.err.p2:> <up.userAgent:RSTestClient> <up.uq:>
```

```
<up.AfHdr:ias-rs-sessionid="kCnX2QAAAAAAAAAAUzAA";> <up.cookie:>
<11436.4592.F0B0S0R0> <RSTEST02.F0> <S0>
I. 2012-11-09 02:36:31.830-0500 RSR row: 63685111 4294967295 upkuu
1024000          0 |         0          0       2379          0 |
0          0          0 |         0          0          0
0          0 |      2439 |         0          0 |         0   1001      0
|     60000 <err:RSE_ROOT_FARM_NOT_FOUND_BY_CLIENT> <warn:RSW_NO_WARNING>
<oe.err:N/A> <oe.err.p0:> <oe.err.p1:> <oe.err.p2:>
<up.userAgent:RSTestClient> <up.uq:> <up.AfHdr:> <up.cookie:> <11436.4592.->
I. 2012-11-09 02:36:34.397-0500 RSR row: 592cbcd2 4294967295 upkuu
1024000          0 |         0          0       2362          0 |
0          0          0 |         0          0          0
0          0 |      2422 |         0          0 |         0   1001      0
|     60000 <err:RSE_ROOT_FARM_NOT_FOUND_BY_CLIENT> <warn:RSW_NO_WARNING>
<oe.err:N/A> <oe.err.p0:> <oe.err.p1:> <oe.err.p2:>
<up.userAgent:RSTestClient> <up.uq:> <up.AfHdr:> <up.cookie:> <13896.6936.->
I. 2012-11-09 02:36:36.964-0500 RSR row: de525dfc 4294967295 unkuu
1024000          0 |         0          0       2361          0 |
0          0          0 |         0          0          0
0          0 |      2422 |         0          0 |         0   1001      0
|     60000 <err:RSE_ROOT_FARM_NOT_FOUND_BY_CLIENT> <warn:RSW_NO_WARNING>
<oe.err:N/A> <oe.err.p0:> <oe.err.p1:> <oe.err.p2:>
<up.userAgent:RSTestClient> <up.uq:> <up.AfHdr:> <up.cookie:> <7160.7896.->
I. 2012-11-09 02:36:58.638-0500 RSR row: dff5c798          1 npkpk
1024477    1031252 |       285       2440       2377          0 |
1001       1000          1 |        21      17927          2
17911         51 |         0 |     865392          9 |       200      0
0 |     60000 <err:RSE_NO_ERROR> <warn:RSW_NO_WARNING> <oe.err:N/A>
<oe.err.p0:> <oe.err.p1:> <oe.err.p2:> <up.userAgent:RSTestClient> <up.uq:>
<up.AfHdr:> <up.cookie:> <7160.7896.F0B0S1R0> <RSTEST02.F0> <S0>
I. 2012-11-09 03:53:12.359-0500 RSR row: 87293f45          0 cnluu
1515          0 |         2          1          0          0 |
0          0          0 |         1          0          0
0          0 |         1 |         0          0 |         0   4015    103
|    120000 <err:RSE_CLIENT_RSOE_REPORT_SESSION_ERR>
<warn:RSW_CONTENT_LENGTH_RESPONSE_NOT_COMPLETED>
<oe.err:OEE_RS_IDX_NOT_FOUND(1029)> <oe.err.p0:61166> <oe.err.p1:_unused_>
<oe.err.p2:_unused_> <up.userAgent:RSTestClient> <up.uq:> <up.AfHdr:ias-rs-
sessionid=RT8phwAA7u4AAAAAUzAA> <up.cookie:> <7800.11500.F0B0S0R0>
<RSTEST02.F0> <S0>
```

Siehe auch

„Affinität“ auf Seite 6

Outbound Enabler Record

Sie können den Outbound Enabler Record (OER) verwenden, um Relay-Ausfälle zu diagnostizieren und die Performance zu studieren. Der OER besteht aus einer kurzen Zusammenfassung der Outbound Enabler-Verarbeitung einer einzelnen Anforderung und enthält Informationen über Anforderungen, Timing, wichtige Informationen für die Fehlersuche, Anforderungsstatus und Datenträger. Der Outbound Enabler Record wird als Teil der Outbound Enabler-Logdatei generiert, wenn die Ausführlichkeitsstufe auf 1 oder höher gesetzt ist. Es gibt eine OER-Datenzeile für jede HTTP-Anforderung.

Der Outbound Enabler Record besteht aus einer Gruppe von Datentypen (durch Symbole dargestellt) und Werten. Die Symbole und die zugehörigen Datentypen sind in der folgenden Tabelle aufgelistet:

Symbol	Datentyp
b:	Bytezähler
c:	Allgemeiner nummerischer Zähler
i:	ID oder nummerischer Code
m:	Zeit gemessen in Millisekunden
x:	Hexadezimalzahl
name:*string*	Benannte Zeichenfolgenwerte variabler Länge Dazu können folgende Elemente gehören: Benutzeragentendaten, Affinitätsdaten, Cookie-Informationen, Outbound Enabler-Fehler, Outbound Enabler-Fehlerparameter, Name des Relay Servers, Relay Server-Fehlernamen und Label.
up	Element im Zusammenhang mit der Relay-Anforderung (mit Ausnahme der Antwort) an den Backend-Server
rtp	Element, das mit Roundtrip-Transport und der Verarbeitung des letzten Aufwärts- und ersten Abwärtspakets verknüpft ist
dn	Element im Zusammenhang mit der Relay-Anforderung (mit Ausnahme der Antwort) vom Backend-Server an den Relay Server

Die Symbole sind zusammengesetzt (als Feldnamen), um auf bestimmte Arten von Daten zu verweisen. Beispiel: **b:up.maxReqQ** Entspricht der Bytezählung (**b:**) der höchsten Speichernutzung pro Anforderung (**up**) der Aufwärts-Warteschlange(**maxReqQ**).

Feldname	Datentyp
i:oidx	Relay Server-Index in der Relay Server-Farm
i:ridx	Temporärer Anforderungsindex, zugewiesen vom Relay Server
i:sidx	Anwendungssitzungsindex, zugewiesen vom Outbound Enabler an die erste Anforderung
x:snum	Sitzungsseriennummer, zugewiesen vom Relay Server an die erste Anforderung
x:sfp	Sitzungsfingerabdruck, zugewiesen vom Relay Server an die erste Anforderung
m:up	Anforderungs-Relay-Zeitraum vom Senden des ersten Anforderungspakets bis zum Senden des letzten Anforderungspakets

Feldname	Datentyp
m:rtp	Roundtrip-Verarbeitungszeitraum vom Senden des letzten Anforderungspakets bis zum Empfang des ersten Antwortpakets zwischen dem Outbound Enabler und dem Backend-Server, einschließlich der Verarbeitungszeit auf dem Backend-Server
m:dn	Antwort-Relay-Zeitraum vom Empfang des ersten Antwortpakets bis zum Empfang des letzten Antwortpakets
m:close	Inaktiver Zeitraum vom Empfang des letzten Antwortpakets bis zur Abfalldaten-sammlung oder der Wiederverwendung des Affinitätskontexts.
b:up.maxReqQ	Höchste Speichernutzung der Aufwärts-Paketwarteschlange nach Anfrage
b:dn.maxShrQ	Höchste Speichernutzung der gemeinsam genutzten Abwärts-Paketwarteschlange während des Antwort-Relay-Zeitraums dieser Anforderung
i:err	Fehler-ID
err.p0	Erster Fehlerparameter vom Outbound Enabler
err.p1	Zweiter Fehlerparameter vom Outbound Enabler
err.p2	Dritter Fehlerparameter vom Outbound Enabler

Ein Outbound Enabler Record sieht ähnlich wie folgendes Beispiel aus:

```
I. 2012-11-09 02:35:12.821-0500 <OEHost> OER header:   i:oidx     i:ridx
i:sidx     x:snum      x:sfp |       m:up      m:rtp       m:dn    m:close |
b:up.maxReqQ b:dn.maxShrQ |  i:err ...other-variable-length-elements...
I. 2012-11-09 02:35:41.070-0500 <Backend-0000> OER row:     0
0          0          0   d9d72990 |       2364        999         23
20620 |        49723        65455 |      0 <err:OEE_NO_ERROR> <err.p0:>
<err.p1:> <err.p2:>
I. 2012-11-09 02:36:05.153-0500 <Backend-0000> OER row:     0
0          0          0   d9d72990 |       2439        999         21
20615 |        56968        65455 |      0 <err:OEE_NO_ERROR> <err.p0:>
<err.p1:> <err.p2:>
I. 2012-11-09 02:36:08.614-0500 <Backend-0000> OER row:     0
0          0          0   d9d72990 |       2439        999
21          0 |        56968        65455 |      0 <err:OEE_NO_ERROR>
<err.p0:> <err.p1:> <err.p2:>
I. 2012-11-09 03:53:12.362-0500 <Backend-0000> OER row:     0
0          1          0   87293f45 |          0          0
0          0 |           0           0 |   1029 <err:OEE_RS_IDX_NOT_FOUND>
<err.p0:61166> <err.p1:> <err.p2:>
```

Siehe auch

„Affinität“ auf Seite 6

Entfernte Verwaltung der Relay Server-Logdatei

Die entfernte Verwaltung der Relay Server-Logdatei wird durch die Klasse *AdminChannel* in *rstool.jar* durchgeführt.

Befehlszeilenverwendung:

```
java ianywhere.ml.rs.AdminChannel [{options}]...
```

Dies können eine oder mehrere der folgenden Optionen sein

Option	Beschreibung
-url *rsAdminUrl*	Verweist auf die **rs_admin**-Erweiterung
-uid *user* -pwd *password*	Stellt Anmeldeinformationen für die HTTP-Authentifizierung für den Zugriff auf **rs_admin** bereit
-ping	Ping der **rs_admin**-Erweiterung
-getRSConfig	Ruft die Relay Server-Konfiguration auf
-setRSConfig	Aktualisiert und sichert die Relay Server-Konfiguration
-hello	Verhandelt Administrationsprotokollversion und Ping
-archiveRSLog	Kürzt und archiviert die aktuelle Online-Logdatei des Relay Servers
-xol *outFile* **none** | *beginTime* **none** | *endTime nRegex* "*regex*"...	Extrahiert Logzeilen aus den archivierten bzw. online vorhandenen lokalen Outbound Enabler-Logdateien. Der Relay Server muss nicht laufen und die Funktion bietet nur lokale Extraktion.
-xrl *outFile* **none** | *beginTime* **none** | *endTime nRegex* "*regex*"...	Extrahiert Logzeilen entfernt von archivierten und/oder online vorhandenen Relay Server-Logdateien. Hierbei gilt Folgendes: • **Time* ist der lokale Zeitstempel im Format JJJJ-MM TT HH: mm: ss.SSS. • *nRegex* ist ein regulärer Ausdruck.
? | -? | /? | -h | /h	Ausgabe dieser Syntaxbeschreibung

Beispiel:

```
java.exe -cp rstool.jar ianywhere.ml.rs.AdminChannel -url https://rs.my.com/
rs16/admin/rs_admin.dll -uid me -pwd changit -hello
java.exe -cp rstool.jar ianywhere.ml.rs.AdminChannel -url https://rs.my.com/
rs16/admin/rs_admin.dll -uid me -pwd changit -xrl rr.xrl none none 1 "RSR
(element|header|row)"
```

Sybase Hosted Relay Service

Der Sybase Hosted Relay Service ist eine Farm von Relay Servern, die von Sybase gehostet wird. Dieser Dienst vereinfacht die Entwicklung mobiler Anwendungen, die die MobiLink-Datensynchronisation verwenden, sowie den Evaluierungsprozess für Entwickler, insbesondere für Anwendungen, bei denen Daten über öffentliche drahtlose Netzwerke gesendet werden. Insbesondere müssen Sie nicht die IT-Abteilung bitten, Anwendungen zu installieren oder die Unternehmensfirewall an bestimmten Stellen für eingehende Verbindungen zu öffnen. Für die gesamte Kommunikation zwischen MobiLink und dem Hostingdienst wird HTTP(S) über eine von MobiLink initiierte abgehende Verbindung verwendet.

Der Sybase Hosted Relay Service ist nicht für Produktionsdeployments vorgesehen. Vor dem Deployment der Produktionsanwendung müssen Sie zunächst den Relay Server in Ihrer eigenen Unternehmensinfrastruktur installieren.

Sybase Hosted Relay Service subskribieren

Um den Sybase Hosted Relay Service zu verwenden, müssen Sie ihn zunächst subskribieren.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Gehen Sie zu http://relayserver.sybase.com/account. Sie gelangen auf die Startseite des Sybase Hosted Relay Service.
2. Erstellen Sie ein Konto, indem Sie auf **Register (Registrieren)** klicken.
3. Sie werden aufgefordert, eine **Subscription ID** (Subskriptions-ID) (wählen Sie eine eindeutige ID für Ihre Organisation) und ein **Password** (Kennwort) einzugeben, Kontaktinformationen für sich selbst und Ihre Organisation anzugeben und die **Hosted Relay Service Terms of Service** (Bedingungen des gehosteten Relay-Dienstes) zu akzeptieren. Klicken Sie auf **Submit** (Übermitteln).

Ergebnisse

Wenn Sie sich erfolgreich registriert haben, erhalten Sie eine E-Mail, die Ihre Registrierung bestätigt.

Nächste Schritte

Sie können sich nun mit Ihrer Subskriptions-ID und Ihrem Kennwort beim Sybase Hosted Relay Server anmelden und eine Serverfarm hinzufügen.

Eine Serverfarm hinzufügen

Bevor Sie den Sybase Hosted Relay Service verwenden, müssen Sie eine Serverfarm hinzufügen.

Voraussetzungen

Es gibt keine Voraussetzungen für diese Aufgabe.

Aufgabe

1. Klicken Sie auf den Typ der hinzuzufügenden Farm. Wählen Sie eine der folgenden Optionen:
   - **Add New MobiLink Farm** (Neue MobiLink-Farm hinzufügen)
   - **Add New SQL Anywhere Web Read-Write Farm** (Neue nicht schreibgeschützte SQL Anywhere-Webserverfarm hinzufügen)
   - **Add New SQL Anywhere Web Read Farm** (Neue lesbare SQL Anywhere-Webserverfarm hinzufügen)
   - **Add New SQL Anywhere Web Read-Only Farm** (Neue schreibgeschützte SQL Anywhere-Webserverfarm hinzufügen)
   - **Add New SQL Remote Message Server Farm** (Neue SQL Remote-Nachrichtenserverfarm hinzufügen)
   - **Add New Afaria Farm** (Neue Afaria-Farm hinzufügen)
   - **Neue SAP Sybase Mobile Office-Farm hinzufügen** (Neue SAP Sybase Mobile Office-Farm hinzufügen)
   - **Add New Sybase Unwired Platform Farm** (Neue Sybase Unwired Platform-Farm hinzufügen)
2. Geben Sie im Feld **Farm Name** (Farmname) einen eindeutigen beschreibenden Namen für die Serverfarm ein.
3. Geben Sie für jeden Server in der Farm einen eindeutigen Namen an. Sie können maximal zwei Server angeben.
4. Klicken Sie auf **Create Farm** (Farm erstellen). Wenn die Farm erfolgreich hinzugefügt wurde, wird eine Bestätigungsseite angezeigt.
5. Klicken Sie auf **Configuration Instructions** (Konfigurationsanweisungen), um weitere Informationen über den Dienst zu erhalten. Die Anweisungen basieren auf den von Ihnen bereitgestellten Informationen.
6. Klicken Sie auf **Log Out** (Abmelden), wenn Sie fertig sind.

Ergebnisse

Der Serverfarm wird hinzugefügt.

Der Relay Server mit MobiLink

Die folgenden Abschnitte enthalten Informationen über die Verwendung des Relay Servers mit MobiLink.

Siehe auch

- Weitere Hinweise dazu, welche Betriebssysteme und Browser für den Relay Server unterstützt werden, finden Sie unter http://www.sybase.com/detail?id=1002288
- Weitere Hinweise zum Deployment des Outbound Enablers finden Sie unter „Bereitstellung des MobiLink-Servers“ [*MobiLink - Serveradministration*]

Client mit der Relay Server-Farm verbinden

Wenn eine Relay Server-Farm richtig konfiguriert wurde, stellt ein Client mithilfe der folgenden URL eine Verbindung mit der Relay Server-Farm her:

http://<*Relay Server client extension URL*>/<*farmname*>

Optionen

Option	Beschreibung
<*Relay Server client extension URL*>	Für Microsoft IIS unter Windows <*Domänenname*><*relayserver.sybase.com*>/*rs/client/rs_client.dll* Für Apache unter Linux <*Domänenname*>/*cli/iarelayserver* Verwenden Sie *relayserver.sybase.com* als <*Domänennamen*>, wenn Sie den öffentlich verfügbaren Sybase Hosted Relay Service verwenden. Weitere Hinweise zum Subskribieren des Diensts sowie eine Anleitung zum Einrichten Ihrer Backend-Server finden Sie unter „Sybase Hosted Relay Service subskribieren“ auf Seite 75.
<*farmname*>	Identifiziert die Backend-Farm (eine Gruppe von Backend-Servern), an die dieser Relay Server die Clientanforderung weiterleitet.

Verbindungsbeispiel für SQL Anywhere-MobiLink-Clients

Ein SQL Anywhere-MobiLink-Client sollte für eine Verbindung mit der Serverfarm **F1** folgende Optionen festlegen:

```
-e "ctp=http;
     adr='host=relayserver.sybase.com;
        url_suffix=/rs/client/rs_client.dll/F1'"
```

Für HTTPS ändern Sie `http` in `https`.

Verbindungsbeispiel für UltraLite/UltraLiteJ-MobiLink-Clients

Ein UltraLite/UltraLiteJ-MobiLink-Client sollte für eine Verbindung mit der Serverfarm **F1** folgende Eigenschaften in der Klasse ULSyncParms festlegen:

- Den Datenstromtyp auf HTTP oder HTTPS festlegen.
- Den Datenstromparameter wie folgt festlegen:

```
"host=relayserver.sybase.com;url_suffix=/rs/client/rs_client.dll/F1"
```

Eine Relay Server-Farm einrichten

Bevor MobiLink-Clients eine Verbindung mit einer Farm herstellen können, müssen Sie die Konfigurationsdatei mit den geeigneten Einstellungen konfigurieren und bereitstellen.

Voraussetzungen

Verwenden Sie in diesem Szenario die Microsoft IIS-Version des Relay Servers.

Kontext und Bemerkungen

Angenommen, Unternehmen ABC hat eine mobile Anwendung entwickelt und möchte nun die Deployment-Laufzeitversion einrichten, um die mobile Anwendung zu bedienen. Zunächst umfasst das Deployment 10000 Geräte, und es wird eine Zunahme der Geräte erwartet. Der Kunde möchte daher eine fehlertolerante Umgebung mit Lastverteilung, die die heutige Arbeitslast verarbeiten und problemlos auf eine größere Zahl mobiler Deployments erweitert werden kann. Basierend auf den Merkmalen der Datensynchronisation der mobilen Anwendung hat der Kunde sich für folgende Konfiguration entschieden:

- 2 MobiLink-Server
- 2 Relay Server
- 1 Lastverteiler

- Jeder Relay Server wird auf einem eigenen Computer per Deployment bereitgestellt. Zwei Computer mit den Hostnamen **rs1.abc.com** und **rs2.abc.com** werden verwendet.
- Jeder MobiLink-Server wird auf einem eigenen Computer per Deployment bereitgestellt. Den beiden MobiLink-Servern werden die Namen **ml1** und **ml2** zugeordnet. Sie gehören zu einer Backend-Serverfarm namens abc.mobilink.
- Auf den Lastverteiler kann mit dem Hostnamen **www.abc.com** zugegriffen werden.
- Zur maximalen Sicherheit wird HTTPS von allen Clients und Outbound Enablern verwendet, um eine Verbindung zu den Relay Servern herzustellen. Es wird davon ausgegangen, dass alle Webserver über ein Zertifikat von einer bekannten Zertifizierungsstelle verfügen und alle Backend-Servercomputer das entsprechende vertrauenswürdige Stammzertifikat in ihren Standardzertifikatsspeichern haben.

Aufgabe

1. Im ersten Schritt erstellen Sie die Relay Server-Konfigurationsdatei.

   Die Datei mit der Konfiguration muss *rs.config* heißen. Für dieses spezielle Szenario wird folgende Konfigurationsdatei verwendet.

   ```
   #
   # Options
   #
   [options]
   verbosity = 1

   #
   # Define the Relay Server farm
   #
   [relay_server]
   host = rs1.abc.com

   [relay_server]
   host = rs2.abc.com

   #
   # Define the MobiLink backend server farm
   #
   [backend_farm]
   id = abc.mobilink
   client_security = on
   backend_security = on

   #
   # List MobiLink servers that are connecting to the Relay Server farm
   #
   [backend_server]
   farm = abc.mobilink
   id = ml1
   token = mltoken1

   [backend_server]
   farm = abc.mobilink
   id = ml2
   token=mltoken2
   ```

2. Nehmen Sie ein Deployment der Konfigurationsdatei *rs.config* sowie der Relay Server-Komponenten auf den beiden Computern vor, auf denen der Relay Server ausgeführt wird.

3. Starten Sie mit dem integrierten Outbound Enabler den MobiLink-Server auf den beiden Computern, auf denen die MobiLink-Server ausgeführt werden.

   Auf dem Computer, auf dem der MobiLink-Server mit der ID ml1 ausgeführt wird:

   ```
   mlsrv16 -x oe<config=oe1.txt> -zs ml1 <other ML options>
   ```

   wobei oe1.txt = `-f abc.mobilink -id ml1 -t mltoken1 -cr "host=www.abc.com;port=443;https=1"`

   Auf dem Computer, auf dem der MobiLink-Server mit der ID ml2 ausgeführt wird:

   ```
   mlsrv16 -x oe<config=oe2.txt> -zs ml2 <other ML options>
   ```


wobei oe2.txt = `-f abc.mobilink -id ml2 -t mltoken2 -cr "host=www.abc.com;port=443;https=1"`

Siehe „mlsrv16-Option -x“ [*MobiLink - Serveradministration*].

Ergebnisse

Wenn alle Server und Outbound Enabler ausgeführt werden, können MobiLink-Clients mithilfe der folgenden Verbindungsinformationen eine Verbindung zu der Farm herstellen:

- **HTTPS** protocol
- **host** www.abc.com
- **url_suffix** /rs/client/rs_client.dll/abc.mobilink

Nächste Schritte

MobiLink-Clients mit der Farm verbinden.

Siehe auch

- „Ende-zu-Ende-Verschlüsselung“ [*SQL Anywhere Server - Datenbankadministration*]

Index

Symbole

A

B

C

D

E

F

G

H

I

R

S

U

V

W