SQL Anywhere - リレーサーバ
文書バージョン: 17 – 2016-05-11

リレーサーバ

目次

1 Relay Server

このマニュアルでは、Relay Server の設定方法と使用方法について説明します。Relay Server は、SAP Afaria、SAP Mobile Office、Mobile Link、SAP SQL Anywhere、SQL Remote、SAP Mobile Server、SAP Mobile Platform サーバとモバイルデバイス間で、Web サーバを介した安全な通信を実現します。

このセクションの内容:

次の条件に従うかぎり、このマニュアルの全部または一部を使用、印刷、再生、配布することができます。

1.1 Relay Server の概要

Relay Server は、Web サーバを通じて通信するモバイルデバイスとバックエンドサーバの間で、安全な負荷分散通信を実現します。サポートされているバックエンドサーバには、SAP Afaria、SAP Mobile Office、Mobile Link、SAP SQL Anywhere、SQL Remote、SAP Mobile Server、および SAP Mobile Platform などがあります。

Relay Server には次の機能があります。

- モバイルデバイスとバックエンドサーバの通信に共通の通信アーキテクチャを提供。
- 負荷が分散されたフォールトトレラントな環境を可能にするメカニズムをバックエンドサーバに提供。
- 企業の既存のファイアウォール設定やポリシーと簡単に統合でき、モバイルデバイスとバックエンドサーバ間の通信を可能化。

このセクションの内容:

1.1.1 Relay Server のアーキテクチャ

一般的に、Relay Server はモバイルデバイス、1 Relay Server、および 1 つ以上のバックエンドサーバという構成で配備されます。

Relay Server の配備環境の構成要素を次に示します。

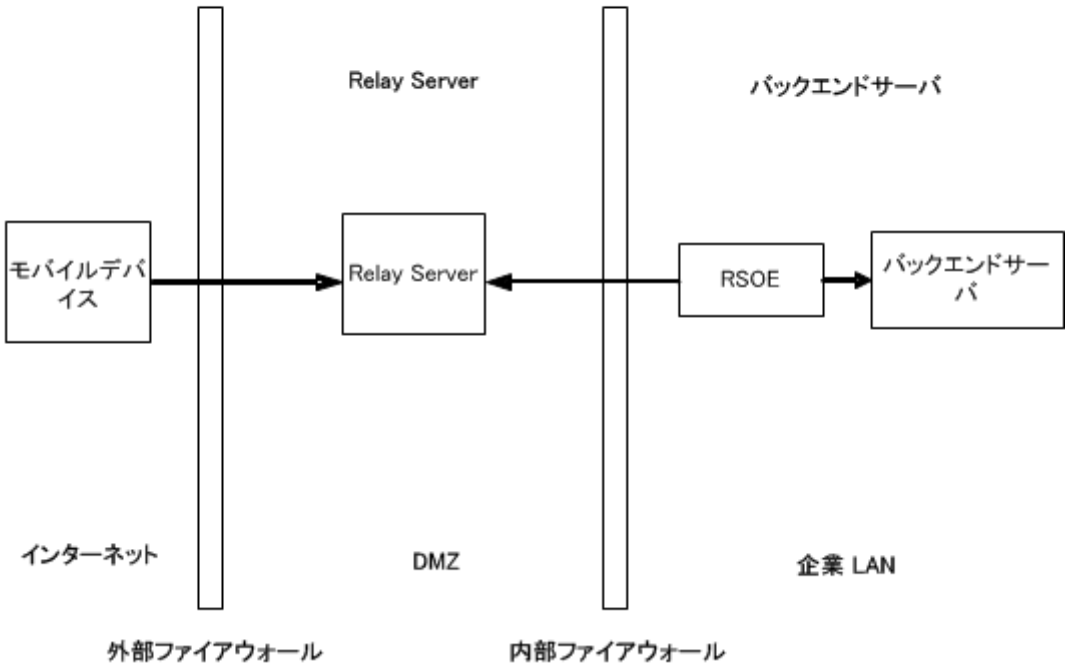
- 企業 LAN 内のバックエンドサーバで通信するクライアントアプリケーションおよびサービスが稼動しているモバイルデバイス。
- モバイルデバイスからの要求を Relay Server のグループに送信する、オプションのロードバランサ。
- 企業 DMZ 内で実行されている 1 つ以上の Relay Server。
- 企業 LAN 内で実行され、クライアント要求を処理する 1 つ以上のバックエンドサーバ。次のバックエンドサーバは、Relay Server とともに使用できます。
  - SAP Afaria
  - SAP Mobile Office

- Mobile Link
- SAP Mobile Platform
- SAP SQL Anywhere
- SQL Remote
- SAP Mobile Server

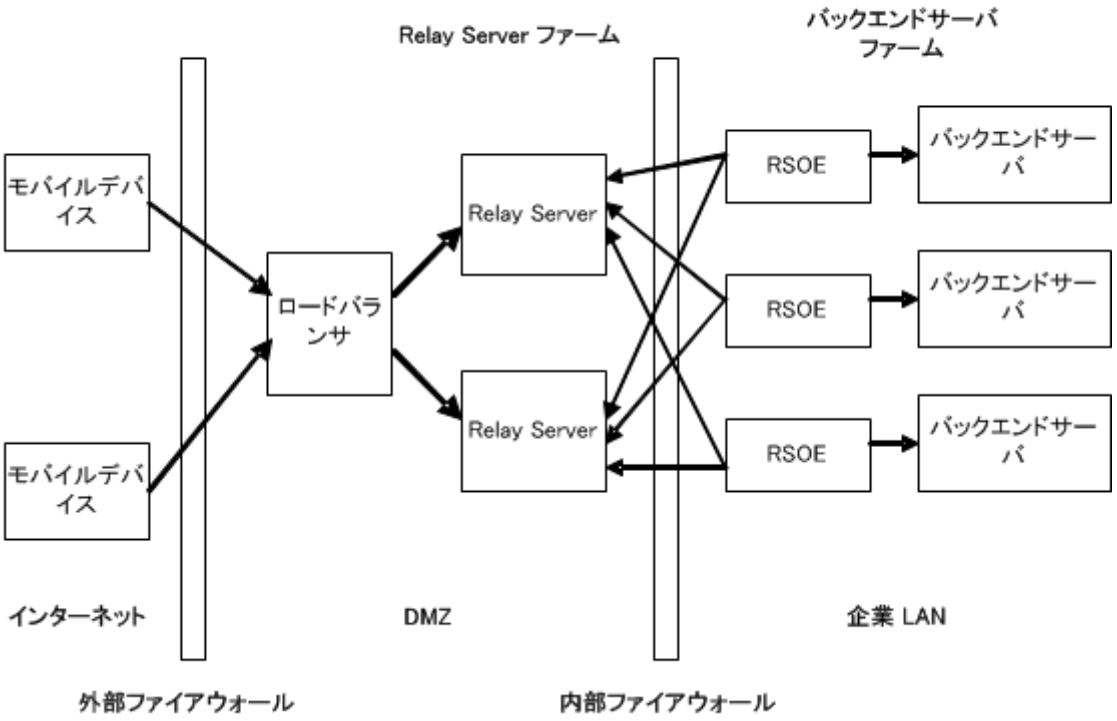
> **注記**
>
> Relay Server のテストには、適切に定義された HTTP 要求および応答を通じて通信する特定のバックエンドサーバとクライアントが使用されています。SQL Anywhere を Web サーバとして使用するなど、カスタム HTTP トラフィックを使用する配備では、徹底的にトラフィックをテストして、その配備が確実に Relay Server で機能するようにする必要があります。
>
> どのバックエンドサーバがサポートされているかの情報については、ライセンス契約または "SQL Anywhere がサポートするプラットフォームおよびエンジニアリングサポート状況" ページを参照してください。

- Relay Server Outbound Enabler (RSOE) は、通常はバックエンドサーバごとに 1 つしかありません。Outbound Enabler は、バックエンドサーバと Relay Server ファームとの間のあらゆる通信を管理します。

次の図は、単一の Relay Server を持つ Relay Server のアーキテクチャを示しています。



次の図に示されている Relay Server アーキテクチャでは、Relay Server ファームとバックエンドサーバファームは 1 つずつですが、システムが少し複雑になっています。



Relay Server は、Web サーバ、3 つの Web 拡張機能 (管理拡張、クライアント拡張、サーバ拡張)、およびステータス情報を保持するバックグラウンドプロセス (Apache のみ) で構成されます。クライアント拡張機能は、モバイルデバイスで実行されているアプリケーションから行われたクライアント要求を処理します。サーバ拡張機能は、バックエンドサーバに代わって Outbound Enabler からの要求を処理します。

Relay Server は Web サーバで実行されている Web 拡張機能であるため、すべての通信は HTTP または HTTPS を使用して実行されます。HTTP を使用すると、企業の既存のファイアウォール設定やポリシーとの統合が容易です。Relay Server では、企業 LAN から Relay Server への接続を企業 LAN 内から開始することが必要になります。この構成によって、DMZ から企業 LAN 内へのインバウンド接続が必要ではなくなるため、より安全な配備環境が提供されます。

共有メモリとセキュリティ **(Apache** のみ**)**

Relay Server は、クライアントとサーバプラグインとの間で HTTP 要求と応答を転送するのに共有メモリを使用します。安全な配備では、クライアントと Relay Server、Outbound Enabler と Relay Server の間で HTTPS を使用します。このシナリオでは、Web サーバが HTTPS を HTTP に複号化し、Relay Server が Outbound Enabler へ向かう途中で HTTP を再暗号化します。Relay Server でデータの暗号化が解除される短い時間のことを、ワイヤレスアプリケーションプロトコルギャップまたは WAP ギャップと呼ぶことがあります。

このデータを同じコンピュータ上の不良プロセスから守るには 2 つの方法があります。1 つ目はエンドツーエンドの暗号化をサポートするクライアントとバックエンドサーバを使用する方法です。ほとんどの Mobile Link クライアントは、エンドツーエンド暗号化をサポートしています。2 番目の方法は、すべての Relay Server に最小限推奨するものですが、サポートされている Web サーバとオペレーティングシステムごとに文書化されている標準技術を使用して DMZ に配備する Web サーバとオペレ

ーティングシステムを強化することです。この強化には、Web サーバコンピュータのオペレーティングシステムアカウントの数を削減する手順が含まれる必要があります。この強化はまた、コンピュータ/VM を Relay Server と Web サーバの実行だけに制限します。システム強化の目標は、コンピュータ上でのプロセスの数を最小限にしながら、不良エージェントが不良プログラムを追加するのを防ぐことです。

このセクションの内容:

1.1.1.1 Relay Server ファーム

Relay Server ファームは、1 つから任意の数までの Relay Servers で構成されます。

複数の Relay Server がある場合、フロントエンドのロードバランサを使用するのが一般的です。さらに、ラウンドロビンといった他の負荷分散ソリューションを使用することもできます。

バックエンドサーバファーム

バックエンドサーバファームは、同種のバックエンドサーバから構成されるグループです。Relay Server ファームを通じて要求を行うクライアントは、対象とするバックエンドサーバファームを指定します。

負荷分散装置

ロードバランサは、モバイルデバイスからの要求を Relay Server ファームで実行されている Relay Server に送信します。Relay Server が 1 つだけである場合、負荷分散装置は必要ありません。

Mobile Link Relay Server Outbound Enabler

Relay Server Outbound Enabler (RSOE) は、バックエンドサーバと同じコンピュータ上で実行されます。RSOE は、バックエンドサーバに代わって Relay Server ファーム内のすべての Relay Server へのアウトバウンド接続を開始します。通常、バックエンドサーバあたりの RSOE は 1 つのみです。

関連情報

Outbound Enabler [32 ページ]

1.1.1.2 Relay Server のセキュリティ

Relay Server には組み込みセキュリティ機能がありますが、安全な通信をサポートするために Web サーバで提供されるセキュリティ機能にも依存しています。

Relay Server は、Web サーバとの組み合わせにより、サーバ、クライアント、および RSOE 認証のためのトランスポートレイヤセキュリティ、HTTP 認証、RSOE プロキシ認証、RSOE MAC アドレスフィルタリング、RSOE トークン認証、およびクライアント暗号化技術 (プロトコルレベルの暗号化) をサポートしています。

サーバ側証明書とトランスポートレイヤセキュリティ

Relay Server と通信するクライアントまたは RSOE は、サーバ側の証明書を使用して、Relay Server を実行している Web サーバが信頼できるかどうかを確認できます。クライアントまたは RSOE に格納されているルート証明書を使用して、Web サーバのパブリック証明書を確認します。証明書が確認されると、キー交換が行われ、暗号化された接続が確立されます。

クライアント側証明書とトランスポートレイヤセキュリティ

Web サーバは、クライアント側の証明書を使用して、Relay Server と通信するクライアントまたは RSOE が信頼できるかどうかを確認できます。Web サーバコンピュータの証明書マネージャに格納されているルート証明書を使用して、クライアントのパブリック証明書を確認します。証明書が確認されると、キー交換が行われ、暗号化された接続が確立されます。

バックエンドサーバとバックエンドファームの設定

Relay Server は、`rs.config` ファイルを使用して、バックエンドファームとバックエンドサーバの設定を含め、ファーム環境で稼働しているときの Relay Server のピアリストを定義します。Relay Server ファーム内の Relay Server はそれぞれ、`rs.config` ファイルの同一のコピーが必要です。

バックエンドファームとバックエンドサーバが設定されていると、Relay Server は設定済みのコンピュータのみと通信するようになります。Relay Server が設定されていないコンピュータと通信しようとしても拒否されます。

HTTP 認証

Relay Server は、Web サーバによって提供される HTTP 認証オプションを完全に継承します。たとえば、クライアントまたは RSOE と Web サーバとの間では、基本、ダイジェスト、Windows 8、およびクライアント認証マッピングを使用できます。

RSOE MAC アドレスフィルタリングとトークン認証

`rs.config` ファイルのバックエンドサーバセクションには、バックエンドファーム内の各サーバが、ID および関連するファーム名で設定されています。ID はサーバ名に対応します。Relay Server には、RSOE で稼働しているコンピュータの MAC アドレスを確認する機能があります。これにより、内部ファイアウォールから通信するサーバが信頼できることを確認し、Relay Server との接続の確立を許可することができます。MAC プロパティは、RSOE で使用するネットワークアダプタの MAC アドレスです。アドレスは、IEEE 802 MAC-48 フォーマットで指定します。

また、バックエンドサーバセクションには、バックエンドサーバ接続を認証するために Relay Server が使用するセキュリティトークンを設定することもできます。Relay Server との接続を確立する場合、RSOE の起動時にトークンが指定されている必要があります。

RSOE プロキシ認証

RSOE は、ユーザ ID とパスワードを使用する HTTP プロキシ認証をサポートしています。

Mobile Link のセキュリティ

Mobile Link クライアントは、HTTP または HTTPS を使用して Relay Server と通信します。HTTPS 通信では、データが一時的に復号化され、クライアントとバックエンドサーバ間でデータが交換されるときに再び暗号化されます。WAP ギャップを介した安全な通信を保証するため、Mobile Link エンドツーエンド暗号化機能を使用して Relay Server に渡されるデータをさらに保護してください。Mobile Link のエンドツーエンド暗号化機能は、SQL Anywhere、Ultra Light クライアント、および Mobile Link サーバ間で、RSA を介したプロトコルレベルの暗号化を実行します。トランスポートレイヤセキュリティの有無にかかわらず、エンドツーエンドの暗号化が可能です。

バックエンドサーバの HTTP 受信ポート

セキュリティ上の理由から、Outbound Enabler を使用する場合は、バックエンドサーバの HTTP 受信ポートをループバック IP アドレス (127.0.0.1) に明示的にバインドすることで、攻撃にさらす部分を最小化してください。

関連情報

Relay Server 設定ファイル [20 ページ]
SSL を使用する Microsoft IIS での Relay Server の設定
SSL を使用する Apache での Relay Server の設定

1.1.2 アフィニティ

Relay Server を使用しているネットワークにおいて、アフィニティとは、複数の HTTP 要求全体でのクライアントとバックエンドサーバの間の関連付けを意味します。アフィニティは、クライアントが同じバックエンドサーバに複数の要求を送信するときのみ必要となります。

Relay Server はアフィニティの情報を HTTP 応答に追加し、HTTP 要求で送信されたアフィニティ情報に従います。アフィニティ情報は HTTP cookie を使用して送信されます。Relay Server を通じて同じバックエンドサーバに複数の要求を送信するクライアントは、各 HTTP 応答で受信したアフィニティ情報を、次の HTTP 要求に挿入して返送する必要があります。バックエンドサーバは、何もしなくても、Relay Server のアフィニティに参加できます。同じバックエンドサーバに送信される一連の要求に非永続的 HTTP が使用される場合、要求ごとに新しい TCP ソケット接続が作成されます。1 つのファームに複数のバックエンドサーバがある場合、クライアントは要求間でアフィニティ情報を維持する必要があります。Relay Server は、アフィニティ情報によって、一連の要求が同じバックエンドサーバに関連付けられ、そのサーバを宛先としていることを知ります。

アプリケーションが RESTful ではなく、それが理由でアフィニティ要求がある場合、クライアントは標準の HTTP cookie を反映するためのサポートを必要とします。Relay Server は非永続的 HTTP cookie を使用してアフィニティを保持します。

関連情報

バックエンドファームの設定 (コマンドライン) [26 ページ]

1.1.3 Relay Server のステータスページ

Relay Server のステータスページは、サービス、ホスト、およびバックエンドファームの利用可能性といった情報を提供します。

Relay Server のステータスは次のとおりです。

- ホスト名
- `rs.config` で指定された記述。
- サービス開始時刻 (UTC)
- ステータスキャプチャ時刻 (UTC)
- ステータス再表示間隔または手動で再表示されることを示すインジケータ
- 全体的な可用性
- 使用できないバックエンドファームのリスト

- 部分的に使用できないバックエンドファームのリスト
- 使用可能なバックエンドファームのリスト

詳細ステータスページとバックエンドサーバのステータスページは両方とも、バックエンドサーバのステータス情報を提供します。

> **注記**
>
> ステータスページに情報が表示されない場合、設定に問題があるために拡張機能がこのページを生成できない可能性があります。

ias-rs-status-refresh-sec = n パラメータを使用して、バックエンドサーバのステータスページの自動更新の頻度を設定します。n がゼロの場合、自動更新はオフになります。ステータスページを分ごとに更新するには、Microsoft IIS Relay Server で下記の URL を使用します。

```
http://host/rs/client/rs.dll?ias-rs-status-refresh-sec=60&ias-rs-farm=SimpleTestApp-farm
```

Microsoft IIS ステータスページの通常の URL は、`http://MyHost:80/rs17/server/rs.dll` です。

Apache ステータスページの通常の URL は、`http://MyHost:80/srv/ias_relay_server/server/server.dll` です。

その他のステータスページ

次のステータスページも使用できます。

ステータスページ	ユーザまたはロケーション	URL フォーマットの例
詳細なステータス	Relay Server の管理者	IIS: `http://host/rs17/admin/rs.dll` Apache: `http://host/admin/iarelayserver`
全体的なステータス	リモートユーザ	IIS: `http://host/rs17/client/rs.dll` Apache: `http://host/cli/iarelayserver`
バックエンドファームのステータス	リモートユーザ	IIS: `http://host/rs17/client/rs.dll?ias-rs-farm=App-farm` Apache: `http://host/cli/iarelayserver?ias-rs-farm=App-farm`

ステータスページ	ユーザまたはロケーション	URL フォーマットの例
バックエンドサーバのステータス	リモートユーザ	IIS: `http://host/rs17/client/rs.dll?ias-rs-farm=App-farm&ias-rs-server=App-server` Apache: `http://host/cli/iarelayserver?ias-rs-farm=App-farm&ias-rs-server=App-server`
全体的なステータス	バックエンドサーバ管理者	IIS: `http://host/rs17/server/rs.dll` Apache: `http://host/srv/iarelayserver`
バックエンドファームのステータス	バックエンドサーバ管理者	IIS: `http://host/rs17/server/rs.dll?ias-rs-farm=App-farm` Apache: `http://host/srv/iarelayserver?ias-rs-farm=App-farm`
バックエンドサーバのステータス	バックエンドサーバ管理者	IIS: `http://host/rs17/server/rs.dll?ias-rs-farm=App-farm&ias-rs-server=App-server` Apache: `http://host/srv/iarelayserver?ias-rs-farm=App-farm&ias-rs-server=App-server`

1.2 Relay Server の配備

Relay Server は、IIS 7.0、7.5、8.0、または 8.5、および Linux 上の Apache に配備できます。

1.3 Microsoft Windows Server への Relay Server コンポーネントの配備 (コマンドライン)

Relay Server ファーム内の各コンピュータに Relay Server ファイルを設定し、配備します。

前提条件

配備プラットフォームを選びます。

- Windows Server 2008 (Agentry トラフィック用 WebSocket サポートのない IIS 7.0)
- Windows Server 2008 R2 (Agentry トラフィック用 WebSocket サポートのない IIS 7.5)
- Windows Server 2012 (Agentry トラフィック用 WebSocket サポートのある IIS 8.0)
- Windows Server 2012 R2 (Agentry トラフィック用 WebSocket サポートのある IIS 8.5)

Relay Server コンポーネントは、SQL Anywhere インストールを使用してインストールされます。デフォルトでは、すべてのファイルが *%SQLANY17%* にインストールされます。

- *%SQLANY17%*`¥Bin64` には、サポートされている DLL と管理用実行プログラムが含まれています。
- *%SQLANY17%*`¥RelayServer¥IIS¥Bin64` には、`rs.dll`、`rs.config` 設定ファイル、およびログフォルダが含まれています。
- *%SQLANY17%*`¥java¥rstool.jar` には、Relay Server 用の管理ツールが含まれています。

コンテキスト

%SQLANY17%`¥RelayServer¥IIS¥iis7_plus_setup.bat` 設定スクリプトは次のタスクを実行します。

1. Microsoft IIS 7 または Microsoft IIS 8 をインストールし、必要な機能をオンにします。
2. ご使用の Microsoft IIS 設定をバックアップします。
3. Microsoft IIS 7 または Microsoft IIS 8 を Relay Server 用に設定します。

Web サーバ管理者は、必要に応じてスクリプトをカスタマイズできます。

Windows 用の Relay Server は、次の実行プログラムから構成されます。

- `rs.dll`
- `dblgen`*17*`.dll`

- `dbfhide.exe`
- `dbicu17.dll`
- `dbicudt17.dll`
- `dbsupport.exe`
- `dbghelp.dll`

手順

1. `iis7_plus_setup.bat` を実行します。
2. Windows 上で Microsoft IIS 用の Relay Server 設定を更新します。更新中の Relay Server ファームに属するコンピュータごとに、更新した設定ファイルを Relay Server Web サイトのホームディレクトリにある *%SQLANY17%*`¥RelayServer¥IIS¥Bin64¥Server` ディレクトリにコピーします。

結果

Relay Server 設定ファイルが、指定された Relay Server に配備されます。

関連情報

Relay Server 設定ファイル [20 ページ]

1.4 Linux 上の Apache への Relay Server コンポーネントの配備 (コマンドライン)

Apache で Relay Server を実行する前に、Relay Server ファーム内の各コンピュータに Relay Server ファイルを設定、配備します。

前提条件

SQL Anywhere インストールには、Relay Server コンポーネントが含まれます。Relay Server ファイルは、*/opt/sqlanywhere17* に直接インストールされます。

コンテキスト

> **注記**
>
> 対話型クイック設定機能を使用して、Relay Server 向けに Apache Web サーバを設定することもできます。これは、ここに記載されている手順の代替手段となります。`/relayserver/quicksetup_apache` ディレクトリにある `ap-setup.sh` スクリプトを実行します。

Apache 用の Relay Server は、次の実行ファイルから構成されます。

- `mod_rs_ap_client.so`
- `mod_rs_ap_server.so`
- `rshost`
- `dblgen17.res`
- `libdbtasks17.so`
- `libdbtasks17_r.so`
- `libdbicudt17.so`
- `libdbicu17_r.so`
- `dbsupport`
- `dbfhide`
- `mod_rs_ap_admin.so`

手順

1. Relay Server 設定ファイル `rs.config` を作成します。
2. `install-dir/relayserver/apache??/bin64` ディレクトリに `rs.config` をコピーします。`??` は、Apache 2.2.x の場合は 22、Apache 2.4.x の場合は 24 です。
3. 次のガイドラインに従って、Relay Server 設定ファイル `rs.config` を編集します。
   - ファイルには 4 つのセクションがあり、各セクションの先頭には各カッコに囲まれたセクションタグが配置されています。
     - Relay Server セクション
     - バックエンドファームセクション
     - バックエンドサーバセクション
     - オプションセクション
   - 各セクションに適切なプロパティを追加します。各プロパティを、キーワードと値のペアとして定義します。
   - 設定ファイルは 7 ビット ASCII 文字のみをサポートします。
4. 以下のディレクトリを LD_LIBRARY_PATH 環境変数に追加します。
   - `install-dir/lib64`
   - `install-dir/relayserver/apache??/bin64`

   `/apache-dir/bin/envvars` ファイルを編集して、パスの設定、および export LD_LIBRARY_PATH を記述します。
5. クライアントまたはサーバ側の Apache インスタンスを実行する前に、source `/apache-dir/bin/envvars` を実行します。

6. サーバ側の Apache インスタンスを次のように設定します。
   a. Apache の `conf/httpd.conf` ファイルをコピーし、`httpd.conf.server` という名前を付けます。
   b. 下記の行を `conf/httpd.conf` に追加し、Relay Server サーバモジュールをロードします。

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

> **注記**
>
> 異なる URL を使用するすべてのモジュールが関係し、すべてのモジュールは、URL パスで `iarelayserver` 文字列を明示的に指定して検索します。URL のその部分を変更する必要はありません。

   c. Remote 管理のサポートモジュールをロードするための、次の行を追加します。

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

> **注記**
>
> Remote 管理モジュールの設定はオプションです。設定はサーバ側またはクライアント側の設定ファイルに常駐させることができます。

   d. サーバモジュールの `<LocationMatch>` セクションを作成するために次の行を追加します。

```
<LocationMatch /srv/iarelayserver/* >
    SetHandler iarelayserver-server-handler
     "/install-dir/relayserver/apache??/bin64/rs.config"
</LocationMatch>
```

   e. Remote 管理モジュールの `<LocationMatch>` セクションを作成するために次の行を追加します。

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

   f. 受信ディレクティブを更新し、サーバインスタンスがクライアントインスタンス以外のポートまたはアダプタを受信するようにします。このポートはまた、Outbound Enabler を実行しているコンピュータにアクセス可能である必要があります。
   g. サーバ側のインスタンスが別のファイルにログされるよう、ErrorLog ディレクティブを更新し、エラーログファイルの名前を変更します。
   h. サーバ側のインスタンスが別のファイルにログされるよう、CustomLog ディレクティブを更新し、アクセスログファイルの名前を変更します。
   i. クライアント PID ファイルとサーバ PID ファイルが互いに上書きできないよう、PidFile ディレクティブを追加します。次に例を示します。

```
PidFile "logs/httpd_server.pid"
```

   j. Apache が HTTPS 用に設定されている場合、`conf/extra/httpd-ssl.conf` (たとえば `httpd-ssl.conf.server`) の別のコピーを作成し、SSL 受信ポートを変更します。両方の Apache インスタンスが同じファイルを含んでいる場合、2 番目に開始された Apache インスタンスにソケットバインドエラーが発生します。
   k. タイムアウトディレクティブを 60 秒に設定します。タイムアウト値は、max_junction_idle_sec 値と、アプリケーションのタイムアウトロジックで予想される値の両方より大きい値である必要があります。
   l. Apache reqtimeout_module を無効化します。

7. クライアント側の Apache インスタンスを次のように設定します。
   a. Apache `conf/httpd.conf` ファイルを編集します。
   b. Relay Server のクライアントモジュールをロードするため、次の行を追加します。

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

   c. クライアントモジュールに `<LocationMatch>` セクションを作成するため、次の行を追加します。

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

   d. タイムアウトディレクティブを 60 秒に設定します。タイムアウト値は、アプリケーションのタイムアウトロジックで予想される値より大きい値である必要があります。
8. Linux では、Apache がプロセスを生成するときに $TMP、$TMPDIR、または $TEMP 環境変数がグローバルに設定されている場合、Apache 設定は完了です。

   上記のいずれかの環境変数のいずれかがグローバルに設定されていない場合、または特定のテンポラリディレクトリにデフォルトの Relay Server ログファイルを配置したい場合、`/apache-dir/bin/envvars` ファイルを編集して設定を行い、次に $TMP をエクスポートします。

   httpd コマンドラインを使用して直接 Apache を起動する場合、まず `/apache-dir/bin/envvars` をソースに指定してください。

   たとえば、envvars ファイルで $TMP の場所を編集する場合は、次のようにします。

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

   > **注記**
   >
   > Apache ユーザプロセスには、指定された `tmp` ディレクトリへの書き込みパーミッションが必要です。

   これにより、Apache がプロセスを作成する前に、Apache が作成するシェル内で環境変数が設定されます。
9. 起動中に Relay Server の設定を更新するには、次の手順に従います。
   a. 更新された設定ファイルを `install-dir/relayserver/apache??/bin64` にコピーします。
   b. `install-dir/relayserver/apache??/bin64` ディレクトリから、次のコマンドを実行して設定の更新を適用します。

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

   c. 複数のサーバを含むファームとして Relay Server を設定する場合は、Relay Server ファーム内のコンピュータごとに前述の手順を繰り返します。

結果

Relay Server 設定ファイルは、Relay Server ファームのすべてのコンピュータに配備されます。

関連情報

Relay Server 設定ファイル [20 ページ]
Relay Server ステイトマネージャ (Linux) [18 ページ]
Relay Server ステイトマネージャサービス (Linux) [18 ページ]
Relay Server ステイトマネージャ (rshost) のコマンドラインの構文 (Linux) [19 ページ]

1.5 Relay Server ステイトマネージャ (Linux)

Relay Server ステイトマネージャは、クライアント要求と Outbound Enabler セッションを通じて Relay Server のステータス情報を保持するプロセスです。ステイトマネージャは、Relay Server ログファイルの管理も行います。

ステイトマネージャはサービスとして実行されます。

デフォルトの Relay Server ログファイル名は `ias_relay_server_host.log` です。このファイルは TMP、TEMP、または TMPDIR 環境変数によって指定されたディレクトリにあります。これらの変数のどれも設定されていない場合、Relay Server はログファイルを `/tmp` ディレクトリ内に作成します。

> **注記**
>
> Apache ユーザプロセスには、指定された `tmp` ディレクトリへの書き込みパーミッションが必要です。

通常のシャットダウン時に、ステイトマネージャは、ログファイルを `yymmddnn.olg` という形式のファイル名に変更します。`yymmdd` はログファイルの名前が変更されたときの日付を表し、`nn` はその日のログファイルの連続するバージョン番号を表します。

このセクションの内容:

Relay Server ステイトマネージャサービス (Linux) [18 ページ]
: Linux 用 SQL Anywhere サービスユーティリティ (dbsvc) を使用すると、サービスの作成、変更、削除を行うことができます。これを使用して、Relay Server ステイトマネージャを起動時に始動するサービスとして設定します。

Relay Server ステイトマネージャ (rshost) のコマンドラインの構文 (Linux) [19 ページ]
: rshost コマンドは、Linux の Relay Server ステイトマネージャを設定します。この機能は Windows では使用できません。

1.5.1 Relay Server ステイトマネージャサービス (Linux)

Linux 用 SQL Anywhere サービスユーティリティ (dbsvc) を使用すると、サービスの作成、変更、削除を行うことができます。これを使用して、Relay Server ステイトマネージャを起動時に始動するサービスとして設定します。

使用法の詳細については、オプションを指定しないで `dbsvc` を実行してください。

注記

rshost サービスを作成するときは、絶対パスを使用してください。

備考

Apache ユーザのプロセスをステイトマネージャの共有メモリに付加できるようにし、共有メモリへの読み込みと書き込みを可能にするために、同じユーザアカウントを使用します。

例

自動起動するステイトマネージャサービス RelayServer を設定する場合	`dbsvc -y -a apache-user -t rshost -w RelayServer -q -qc -f /your-directory/rs.config -os 100K -ot /tmp/rs.log`
サービスを開始する場合	`dbsvc -u RelayServer`
サービスを停止する場合	`dbsvc -x RelayServer`
サービスをアンインストールする場合	`dbsvc -d RelayServer`

関連情報

Relay Server の設定 (コマンドライン) [23 ページ]

1.5.2 Relay Server ステイトマネージャ (rshost) のコマンドラインの構文 (Linux)

rshost コマンドは、Linux の Relay Server ステイトマネージャを設定します。この機能は Windows では使用できません。

```
rshost [ option ]+
```

パラメータ

オプション

ステイトマネージャを設定するオプションは次のとおりです。これらはすべて省略可能です。

rshost のオプション	説明
-f`filename`	Relay Server 設定ファイルのファイル名を示します。
-o`filename`	ログを取るために使用するファイルの名前を示します。
-os`size`	ログファイルのサイズを制御し、ログファイルバナーの追加情報を示します。-os が指定されている場合、`<yymmdd><nn>.olg` フォーマットを使用して、古いログの名前が変更されます。ログバナーは、コンピュータ名、プロセッサのアーキテクチャ、対象ビルド、オペレーティングシステム情報が追加されて、新しいアクティブなログファイルに書き換えられます。
-oq	起動エラーの発生時にポップアップウィンドウを表示しません。
-q	最小化ウィンドウで実行します。
-qc	完了時にウィンドウを閉じます。
-u	実行している Relay Server の設定を更新します。
-ua	ログファイルを `<yymmdd><nn>.log` にアーカイブして、ファイルをトランケートします。

1.6 Relay Server 設定ファイル

Relay Server 設定ファイルは、Relay Server ファームや、Relay Server ファームによって利用可能となっているバックエンドサーバファームのプロパティを定義します。

Relay Server 設定ファイルの各セクションの先頭にはセクションタグがあります。セクションタグは、セクション名を識別するキーワードを角カッコで囲んだ形式になっています。ファイルは、[options]、[relay_server]、[backend_farm]、および [backend_server] というセクションに分かれています。セクションは任意の順序で指定できます。

セクションタグの後ろに続くいくつかの行が、そのセクションのプロパティを定義します。`property name= value` を指定することでプロパティを定義します。どのセクション名やプロパティ名も、大文字と小文字が区別されません。コメントを示すには、行の先頭にシャープ記号 (#) を配置します。設定ファイルは 7 ビット ASCII 文字のみをサポートします。

ファイル非表示ユーティリティ (dbfhide) は、暗号化を使用して、設定ファイルと初期化ファイルの内容を隠します。Relay Server と Outbound Enabler は、dbfhide により設定ファイルが難読化されたことを自動的に検出して、処理します。

設定ファイルのモニタリング

Microsoft IIS では、Relay Server によって設定ファイルが継続的にモニタされ、次のようなステータス変更がログファイルにレポートされます (レポートは 2 秒遅延します)。

- 設定ファイルが無効になった場合における致命的なエラー。Relay Server は、無効な設定ファイルがある状態で再実行することはできません。
- 設定ファイルは有効であるものの、使用されているコピーと一致しない場合における警告。
- エラーまたは警告が解決された場合における情報メッセージ。

Relay Server の設定ファイルの例

下記の Relay Server 設定ファイルは、2 つの Relay Server、4 つのバックエンドファーム、および 5 つのバックエンドサーバで Relay Server ファームを定義しています。

```
# Relay Servers:
     rs1.rs.com
     rs2.rs.com
#
# Assume the load balanced address of the Relay Server farm is www.rs.com.
#
# Backend farms and servers and their corresponding rsoe2 command lines:
     Company1.Sales.MLFarm
         MLServer01   rsoe2.exe -cr host=www.rs.com -f Company1.Sales.MLFarm -id
MLServer01 -t 7b2493b0-d0d4-464f-b0de-24643e1e0feb
         MLServer02   rsoe2.exe -cr host=www.rs.com -f Company1.Sales.MLFarm -id
MLServer02 -t de1aac83-a653-4e0f-8a6c-0a161a6ee407
     Company1.Manufacturing.MLFarm
         MLServer01   rsoe2.exe -cr host=www.rs.com -f
Company1.Manufacturing.MLFarm -id MLServer01 -t 621ece03-9246-4da7-99e3-c07c7599031c
     Company2.AFFarm
         AFServer01   rsoe2.exe -cr host=www.rs.com -f Company2.AFFarm -id
AFServer01 -t a688728f-1ae7-4438-969e-6f5e30a07882
     Company2.SAPFarm
         SAPG1        rsoe2.exe -cr host=www.rs.com -f Company2.SAPFarm -id SAPG1 -
t de68b75b-ff33-4c81-a920-2333494dfd8a -cs
"host=localhost;port=443;https=1;identity=c:
¥rsoe.id;identity_password=****;trusted_certificates=c:¥testrsaserver.crt"
# Note: The two instances of MobiLinkServer01 are different servers.
#
# The Relay Server configuration file contains authentication information.
# Access to this file must be administered.
#
# URL prefix for client applications:
#     https://www.rs.com/rs17/client/rs.dll/Company1.Sales.MLFarm
#     https://www.rs.com/rs17/client/rs.dll/Company1.Manufacturing.MLFarm
#     https://www.rs.com/rs17/client/rs.dll/Company2.AFFarm
#     https://www.rs.com/rs17/client/rs.dll/Company2.SAPFarm
#
#-------------------------------------
# Relay Server options
#-------------------------------------
[options]
verbosity = 1
#--------------------
# Relay Server peers
#--------------------
[relay_server]
enable          = yes
host            = rs1.rs.com
http_port       = 80
https_port      = 443
description     = Machine #1 in RS farm
[relay_server]
enable          = yes
host            = rs2.rs.com
```

```
http_port       = 80
https_port      = 443
description     = Machine #2 in RS farm
#---------------
# Backend farms
#---------------
[backend_farm]
id              = Company1.Sales.MLFarm
[backend_farm]
enable          = yes
verbosity       = 4
id              = Company1.Manufacturing.MLFarm
description     = Company1's MobiLink farm in their Manufacturing department
[backend_farm]
enable          = yes
id              = Company2.AFFarm
description     = Company2's Afaria farm
[backend_farm]
enable          = yes
id              = Company2.SAPFarm
forward_x509_identity= yes
forwarder_certificate_subject= CN = ¥*.mysap.com, *
forwarder_certificate_issuer= CN = ca??.mysap.com, *
description     = Company2's SAP Gateway farm
#-----------------
# Backend servers
#-----------------
[backend_server]
farm     = Company1.Sales.MLFarm
id       = MLServer01
[backend_server]
farm     = Company1.Sales.MLFarm
id       = MLServer02
[backend_server]
enable   = no
farm     = Company1.Manufacturing.MLFarm
verbosity= inherit
mac      = 01-23-45-67-89-ad
id       = MLServer01
token    = 621ece03-9246-4da7-99e3-c07c7599031c
description= ML server number 1
[backend_server]
enable   = yes
farm     = Company2.AFFarm
id       = AFServer01
mac      = 01-23-45-67-89-ae
token    = a688728f-1ae7-4438-969e-6f5e30a07882
[backend_server]
enable   = yes
farm     = Company2.SAPFarm
id       = SAPG1
mac      = 01-23-45-67-89-af
token    = de68b75b-ff33-4c81-a920-2333494dfd8a
```

このセクションの内容:

Relay Server の設定 (コマンドライン) [23 ページ]
Relay Server 設定ファイル内の Relay Server 用プロパティを定義します。

バックエンドファームの設定 (コマンドライン) [26 ページ]
Relay Server 設定ファイルの [backend_farm] セクションで、バックエンドサーバファームに対するプロパティを設定します。

バックエンドサーバの設定 (コマンドライン) [29 ページ]

Outbound Enabler がバックエンドサーバに代わって Relay Server ファームに接続するときに使用するバックエンドサーバ接続のプロパティを定義します。

Relay Server の自動設定 (コマンドライン) [31 ページ]

auto_config オプションを使用して、バックエンドサーバとバックエンドファームに対するデフォルトのプロパティを設定します。

SAP ホストのリレーサービス [32 ページ]

SAP ホストのリレーサービスとは、SAP がホストする Relay Server ファームのことです。このサービスは、Mobile Link データ同期を使用するモバイルアプリケーションの開発、特に公共無線ネットワークを使用してデータを送信する場合に開発者による評価プロセスを簡素化します。

1.6.1 Relay Server の設定 (コマンドライン)

Relay Server 設定ファイル内の Relay Server 用プロパティを定義します。

手順

1. `rs.config` ファイルを作成します。
2. ファイルの [relay_server] セクションで、下記のプロパティを使用して Relay Server を設定します。

プロパティ	説明
enable	(オプション) Relay Server ファームにこの Relay Server が含まれているかどうかを指定します。 **yes** (デフォルト) Relay Server ファームにこの Relay Server を含めることを示します。 **no** Relay Server ファームにこの Relay Server を含めないことを示します。
host	Outbound Enabler が Relay Server への直接接続を行うために使用するホスト名または IP アドレスを指定します。

プロパティ	説明
http_port	Outbound Enabler が Relay Server への直接接続を行うために使用する HTTP ポートを指定します。値 **0** または **off** を指定すると、HTTP 接続が無効になります。デフォルトでは、このプロパティは有効であり、80 に設定されています。 **0** または **off** Outbound Enabler からの HTTP アクセスを無効にします。 **1 ～ 65535** 指定されたポートでの HTTP アクセスを有効にします。
https_port	Outbound Enabler が Relay Server への直接接続を行うために使用する HTTPS ポートを指定します。値 **0** または **off** を指定すると、HTTPS 接続が無効になります。デフォルトでは、このプロパティは有効であり、443 に設定されています。 **0** または **off** Outbound Enabler からの HTTPS アクセスを無効にします。 **1 ～ 65535** 指定されたポートでの HTTPS アクセスを有効にします。
description	(オプション) カスタム説明を最大 2048 文字で指定します。

3. ファイルの [options] セクションで、下記のプロパティを使用して Relay Server を設定します。ファイルには [options] セクションが 1 つしか含まれていませんが、その設定がファイルに定義されたすべての Relay Server に適用されます。

プロパティ	説明
auto_config	Outbound Enabler が事前の設定なしに非表示のバックエンドファームとバックエンドサーバのトークンを使用して接続できるよう、Relay Server を設定します。 下記のいずれかの条件が true になると、新しい RSOE が接続可能となります。 1. 既知でないファーム名を使用して接続する最初の RSOE です。 2. 既知のファーム名と、そのファーム名で最初に接続したときと同じトークンを使用して接続します。 ファーム名とトークンとの間の関係は、Relay Server の再起動時に失われます (またはリセットされます)。
log_size_limit (Microsoft Windows のみ)	Relay Server ログファイルの最大サイズを設定します。サポートされている単位は、k、K、m、M、g、G です。デフォルト値は 0 です (制限なし)。

プロパティ	説明
min_thread (Microsoft Windows のみ)	スレッドプールに割り当てるスレッドの最小数を指定します。デフォルト設定は *auto* です。アイドルプールのサイズが *max_thread* の半分以上である場合、スレッドはアイドルプールに返されません。
max_thread (Microsoft Windows のみ)	スレッドプールに割り当てるスレッドの最大数を指定します。デフォルト設定は *auto* です。Relay Server はプール内のアクティブスレッドの数を制限しません。アイドルプール内にスレッドが残っていないときに要求が到着すると、Relay Server は 65,535 をスレッド上限として新しいスレッドを割り当てます。*max_thread* は *min_thread* よりも大きい値である必要があります。 Microsoft IIS 7 以降に対して同時要求を 5000 よりも大きくするには、下記の設定すべてを 1 つのラインで指定して、new_limit を 3x5000 より大きくする必要があります。 `%systemroot%¥system32¥inetsrv ¥appcmd.exe set config "Default Web Site" -section:system.webServer/ serverRuntime/ appConcurrentRequestLimit:"new-limit"/ commit:apphost` *max_thread* をソフト制限として機能させるには、`new-limit` を *max_thread* よりも高く設定します。
shared_mem (Linux のみ)	(オプション) Relay Server がステータスの追跡機能に使用する共有メモリの最大量を指定します。デフォルト値は 10 MB です。最大設定値は 4 GB です。下記の条件のいずれかが発生した場合、この設定を変更することを検討してください。 ○ Relay Server と Outbound Enabler の間のネットワークを高速化したい。 ○ バックエンドファームの数を大幅に増やしたい。 ○ バックエンドファームのサイズを大幅に拡大したい。 ○ クライアントの数を大幅に増やしたい。 ○ HTTP 応答のサイズを大幅に拡大したい。 ○ 低速のクライアントやネットワークを追加したい。

プロパティ	説明
verbosity	冗長性レベルを指定します。 **0** (デフォルト) エラーのみをログ。配備には、このログレベルを使用してください。 **1** 要求ロギング。すべての HTTP 要求が Relay Server ログファイルに書き込まれます。 **2** 要求ロギング。HTTP 要求のより詳細な表示を提供します。 **3 以上** 詳細ロギング。主に、技術サポートのために使用されます。 エラーは指定されたログレベルに関係なく表示され、警告はログレベルが 0 よりも大きい場合にのみ表示されます。

4. `rs.config` ファイルを保存します。

結果

Relay Server が設定されます。

1.6.2 バックエンドファームの設定 (コマンドライン)

Relay Server 設定ファイルの [backend_farm] セクションで、バックエンドサーバファームに対するプロパティを設定します。

コンテキスト

Relay Server ファームを通じて要求を行うクライアントは、対象とするバックエンドサーバファームを指定します。バックエンドサーバファームごとに、1 つのバックエンドファームセクションがあります。各セクションは、backend_farm キーワードによって識別されます。

手順

1. `rs.config` ファイルを作成します。
2. ファイルの [backend_farm] セクションで、下記のプロパティを使用して バックエンドファームを設定します。

プロパティ	説明
description	(オプション) カスタム説明を最大 2048 文字で指定します。
enable	(オプション) このバックエンドサーバファームからの接続を許可するかどうかを指定します。 **yes** (デフォルト) このバックエンドサーバファームからの接続を許可します。 **no** このバックエンドサーバファームからの接続を禁止します。
forward_x509_identity	SAP NetWeaver Gateway がどのようにクライアントを認証するかを制御します。方法の 1 つとして、信頼できるフォワーダを通じて X.509 証明書を転送することが挙げられます。このプロパティを yes に設定すると、Relay Server は信頼できるフォワーダから転送されたクライアント ID 情報を抽出し、HTTP ヘッダを使用してこれを SAP NetWeaver Gateway または Web Dispatcher に転送できます。デフォルト設定は no です。
forwarder_certificate_issue	クライアント ID ヘッダの取り扱い方法を制御します。SAP 仲介者のチェーンが存在する場合、クライアント ID ヘッダが要求内にすでに存在することがあります。ただし、すべてのクライアントに、フォワーダとして動作するパーミッションが付与されるわけではありません。既存のヘッダをフォワーダの ID で置き換えることがデフォルトの動作となっています。フォワーダが他のクライアント ID を転送できる権限を付与するには、*forwarder_certificate_issuer*=`match-string` および *forwarder_certificate_subject*=`match-string` を設定します。ここで `match-string` は証明書の対応する複合名フィールドのシリアル化形式に対して照合されます。任意の文字を表すには ? を、任意の文字列を表すには * を使用します。文字を照合するには、?、*、¥ の先頭エスケープ文字として ¥ を使用します。次に例を示します。 `forwarder_certificate_issuer = 'CN = quicksigner, OU = security department, O = my org, L = my city, S = my state, C = my country'`

プロパティ	説明
forwarder_certificate_subject	SAP 仲介者のチェーンが存在する場合、クライアント ID ヘッダが要求内にすでに存在することがあります。ただし、すべてのクライアントに、フォワーダとして動作するパーミッションが付与されるわけではありません。既存のヘッダをフォワーダの ID で置き換えることがデフォルトの動作となっています。フォワーダが他のクライアント ID を転送できる権限を付与するには、*forwarder_certificate_issuer*=`match-string` および *forwarder_certificate_subject*=`match-string` を設定します。ここで `match-string` は証明書の対応する複合名フィールドのシリアル化形式に対して照合されます。任意の文字を表すには ? を、任意の文字列を表すには * を使用します。文字を照合するには、?、*、¥ の先頭エスケープ文字として ¥ を使用します。次に例を示します。 `forwarder_certificate_subject = 'CN = mySapWD??.my.com, OU = SEC, O = SAP, *'`
id	バックエンドサーバファームの名前を最大 2048 文字で指定します。
inject_affinity_query_header	(オプション) OM ADM (Afaria Open Mobile Alliance Device Management) バックエンドサーバとの下位互換性を提供します。 **yes** 下位互換性を提供します。 **no** (オプション) 下位互換性を提供しません。 値が指定されていない場合、HTTP、HTTPS のどちらを使用しても接続できます。
token	(オプション) Relay Server がバックエンドサーバ接続を認証するために使用する、2048 文字までのセキュリティトークン。

プロパティ	説明
verbosity	バックエンドファームに対する冗長性を設定します。 **0** (デフォルト) エラーのみをログ。配備には、このログレベルを使用してください。 **1** 要求ロギング。すべての HTTP 要求が Relay Server ログファイルに書き込まれます。 **2** 要求ロギング。HTTP 要求のより詳細な表示を提供します。 **3 以上** 詳細ロギング。主に、技術サポートのために使用されます。 エラーは指定されたログレベルに関係なく表示され、警告はログレベルが 0 よりも大きい場合にのみ表示されます。

3. `rs.config` ファイルを保存します。

結果

バックエンドファームが設定されます。

関連情報

アフィニティ [10 ページ]

1.6.3 バックエンドサーバの設定 (コマンドライン)

Outbound Enabler がバックエンドサーバに代わって Relay Server ファームに接続するときに使用するバックエンドサーバ接続のプロパティを定義します。

コンテキスト

Relay Server ファームに接続する Outbound Enabler ごとに、1 つのバックエンドサーバセクションがあります。各バックエンドサーバセクションは、backend_server キーワードによって識別されます。バックエンドサーバファームへのバックエンドサーバの割り当ても、バックエンドサーバセクションで行います。

Relay Server は次のバックエンドサーバをサポートします。

- Mobile Link
- SAP Afaria
- SAP Mobile Office
- SAP Mobile Platform
- SAP Mobile Server
- SAP SQL Anywhere

> **注記**
>
> どのバックエンドサーバがサポートされているかの情報については、ライセンス契約または "SQL Anywhere がサポートするプラットフォームおよびエンジニアリングサポート状況" ページを参照してください。

手順

1. `rs.config` ファイルを作成します。
2. ファイルの [backend_farm] セクションで、下記のプロパティを使用して バックエンドサーバを設定します。

プロパティ	説明
description	(オプション) カスタム説明を最大 2048 文字で指定します。
enable	(オプション) このバックエンドサーバからの接続を許可するかどうかを指定します。 **yes** (デフォルト) このバックエンドサーバからの接続を許可します。 **no** このバックエンドサーバからの接続を禁止します。
farm	このバックエンドサーバが属するバックエンドサーバファームの名前を指定します。
id	バックエンドサーバ接続の名前を最大 2048 文字で指定します。
MAC	(オプション) Outbound Enabler が Relay Server と通信するために使用するネットワークアダプタの MAC アドレスを指定します。アドレスの指定には IEEE 802 MAC-48 フォーマットを使用します。MAC アドレスに適したフォーマットを判別するには、Mobile Link Relay Server Outbound Enabler のコンソールまたはログを確認します。プロパティが指定されない場合、MAC アドレスの確認は行われません。
max_junction	Relay Server で利用可能なアクティブジャンクションとアイドルジャンクションの最大数を設定します。この設定は、rsoe2 に対する -jl オプションより優先されます。このオプションのデフォルト値は 1000 です。

プロパティ	説明
max_junction_idle_sec	Outbound Enabler によって破棄されるまでにジャンクションが非アクティブを維持する時間を制御します。アイドルジャンクションプールのサイズが、-jsl オプションによって制御されるスペアジャンクションの下限値を下回ると、Outbound Enabler は自動的にアイドルジャンクションを補充します。デフォルト値は 30 です。
token	(オプション) Relay Server がバックエンドサーバ接続を認証するために使用する、2048 文字までのセキュリティトークン。
verbosity	冗長性レベルを指定します。 **0** エラーのみをログに記録します。配備には、このログレベルを使用してください。これはデフォルトです。 **1** 要求ロギング。すべての HTTP 要求が Relay Server ログファイルに書き込まれます。 **2** 要求ロギング。HTTP 要求のより詳細な表示を提供します。 **3 以上** 詳細ロギング。主に、技術サポートのために使用されます。 エラーは指定されたログレベルに関係なく表示され、警告はログレベルが 0 よりも大きい場合にのみ表示されます。

3. `rs.config` ファイルを保存します。

結果

バックエンドサーバが設定されます。

1.6.4 Relay Server の自動設定 (コマンドライン)

auto_config オプションを使用して、バックエンドサーバとバックエンドファームに対するデフォルトのプロパティを設定します。

コンテキスト

auto_config オプションをオンに設定すると、Relay Server は TOFU (trust on first use) システムとして実行されます。自動設定は、テスト目的や管理者を置かない環境において役立ちます。

Outbound Enabler は未知のバックエンドファームやバックエンドサーバと接続します。同じバックエンドファームに属する Outbound Enabler は、ファーム全体のトークンを使用して接続します。

Relay Server を再起動しても、自動設定されたファーム設定は維持されます。バックエンドサーバの設定もまた、バックエンドファームに対する一意のサーバトークンとともに、Outbound Enabler ごとに維持されます。

自動設定を使っても、バックエンドファームやバックエンドサーバの設定を変更できます。変更を行うには、IIS または Apache の Apache または AdminChannel に対して rshost をローカルに使用します。

手順

1. Relay Server が起動する前に、Relay Server 設定ファイルの backend_farm セクションのトークンプロパティでファーム名を指定することで、ファーム名を予約します。

   同じ自動設定ファームに属するすべての Outbound Enablers は、ファーム全体のトークンと一致するトークンを供給する必要があります。トークンが一致しない場合、アクセスが拒否されます。
2. で、auto_config オプションを に設定します。

結果

未知のファーム名を持つ最初の Outbound Enabler を Relay Server が処理すると、新しいバックエンドファーム設定が作成されます。Relay Server は元の設定ファイルを更新し、供給されたトークンを新しいバックエンドファームプロパティに格納します。他のバックエンドファームプロパティはデフォルト値に初期化されます。

1.6.5 SAP ホストのリレーサービス

SAP ホストのリレーサービスとは、SAP がホストする Relay Server ファームのことです。このサービスは、Mobile Link データ同期を使用するモバイルアプリケーションの開発、特に公共無線ネットワークを使用してデータを送信する場合に開発者による評価プロセスを簡素化します。

SAP ホストのリレーサービスを使用することで、何かをインストールしたり企業ファイアウォールに穴を開けたりするように IT 部門に依頼する必要はありません。Mobile Link とホスティングサービスの間の通信はすべて、Mobile Link をホストするコンピュータで開始されたアウトバウンド接続を介した HTTP(S) を使用して行われます。

SAP ホストのリレーサービスは、運用環境への配備を目的としていません。運用アプリケーションを配備する前に、自社のインフラストラクチャまたはクラウドに Relay Server をインストールしてください。

SAP ホストのリレーサービスを使用するには、SAP ホストのリレーサービスにアクセスしてください。

1.7 Outbound Enabler

Outbound Enabler は、企業 LAN 内で稼働しているコンピュータから DMZ 内で実行されている Relay Server ファームへのアウトバウンド接続を開き、Relay Server から受信したクライアント要求をバックエンドサーバに、バックエンドサーバからの応答を Relay Server 経由でクライアントに転送します。

Outbound Enabler は、バックエンドサーバと同じコンピュータ上で実行されます。Outbound Enabler は、起動されると、直接 URL または負荷分散された URL を使用して、ファーム内で実行されている Relay Server のリストを取得するための HTTP 要求を行います。Outbound Enabler からの要求を受信した Relay Server は、Relay Server のサーバ拡張機能コンポーネントにマッピングするサーバ URL を使用して、ファーム内のすべての Relay Server の接続情報を返します。URL は、Relay Server ファーム環境内のさまざまなプロキシ、ロードバランス、または高可用性オプションを介して、直接または間接的に Relay Server のサーバ拡張にマッピングします。URL は、各 Relay Server の Relay Server 設定ファイルに格納されます。

Outbound Enabler は、Relay Server への論理接続とバックエンドサーバへの接続からなるジャンクションを生成します。ジャンクションは、これらの接続の 1 対 1 の関連性を示します。ジャンクションは、Relay Server とバックエンドサーバとの間で、クライアント要求とそれに対応するサーバ応答を伝送します。Relay Server と Outbound Enabler はジャンクションの半分を使用して互いに通知しあうこともあります。Outbound Enabler は、リレーが必要な時に遅延を低減するため、ジャンクションのプールを保持しています。要求が出されると、その要求の有効期間にわたってジャンクションが HTTP 要求に割り当てられます。要求が完了すると、割り当てられたジャンクションは新しいバックエンド接続と再度組み合わされ、今後の使用のためにプールに戻されます。指定された時間内に使用されなかったジャンクションは破棄され、再度生成されます。ジャンクションのプールを保持するには、コストがかかります。rsoe2 ユーティリティにはプールのサイズを制御するオプションがあるほか、Relay Server 設定ファイルでは、各バックエンドサーバに対して許可されたジャンクションの最大数を制限することができます。

このセクションの内容:

1.7.1 Relay Server Outbound Enabler の構文

rsoe2 コマンドは、Relay Server Outbound Enabler (RSOE) を実行します。

```
rsoe2 [ option ]+
```

```
rsoe2 @{ filename | environment-variable } ...
```

パラメータ

オプション

デフォルトのあるオプションは省略可能です。Outbound Enabler は、最低でも、Relay Server (-cr)、ファーム (-f)、およびサーバ (-id) の名前の接続文字列を提供する必要があります。セキュリティトークンが設定されている場合は、それも指定する (-t) 必要があります。

rsoe2 オプション	説明
@data	指定された環境変数または設定ファイルからオプションを読み込みます。 設定ファイル内の情報を保護する場合は、ファイル非表示ユーティリティ (dbfhide) を使用して、設定ファイルの内容をエンコードします。

rsoe2 オプション	説明
`-cr"connection-string"`	Relay Server の接続文字列を指定します。Relay Server の接続文字列のフォーマットは、キーワードと値のペアがセミコロンで区切られたリストです。 **host** Relay Server の IP アドレスまたはホスト名。デフォルトは localhost です。 **port** 必須。Relay Server が受信しているポート。 **http_userid** 省略可能です。認証用ユーザ ID。HTTP 認証の設定方法については、Web サーバ (またはプロキシ) のマニュアルを参照してください。 **http_password** 省略可能です。認証用パスワード。HTTP 認証の設定方法については、Web サーバ (またはプロキシ) のマニュアルを参照してください。 **http_proxy_userid** 省略可能です。プロキシ認証用ユーザ ID。HTTP 認証の設定方法については、Web サーバ (またはプロキシ) のマニュアルを参照してください。 **http_proxy_password** 省略可能です。プロキシ認証用パスワード。HTTP 認証の設定方法については、Web サーバ (またはプロキシ) のマニュアルを参照してください。 **proxy_host** 省略可能です。プロキシサーバのホスト名または IP アドレス。 **注記** プロキシサーバのバッファリングに非常に長い時間がかかるという問題が発生する場合は、HTTPS を使用します。これによって、プロキシのバッファリングが回避されます。 **proxy_port** 省略可能です。プロキシサーバのポート番号。 **url_suffix** 必須。Relay Server のサーバ拡張機能への URL パス。 **https** 0 - HTTP (デフォルト)

rsoe2 オプション	説明
	1 - HTTPS *https=1* の場合は、次のオプションも指定できます。Outbound Enabler が正しいバックエンドサーバに接続されていることを確認するために、certificate_name、certificate_company、certificate_unit のうちの少なくとも 1 つを指定してください。証明書をチェックしないようにするには、certificate_name_check オプションを指定してください。 **certificate_name** 証明書の通称フィールド。 **certificate_company** 証明書の組織名フィールド。 **certificate_unit** 証明書の組織単位フィールド。 **identity** このオプションは、Outbound Enabler とバックエンドサーバとの間で相互に認証された TLS を確立するためのクレデンシャルを提供します。相互認証はバックエンドサーバに必要です。 **identity_password** このオプションは、Outbound Enabler とバックエンドサーバとの間で相互に認証された TLS を確立するためのクレデンシャルを提供します。相互認証はバックエンドサーバに必要です。 **fips** TLS 暗号化とエンドツーエンド暗号化に FIPS 認定の暗号化の実装を使用するかどうかを選択します。 **skip_certificate_name_check** クライアントライブラリがサーバホスト名とデータベースサーバの証明書ホスト名との照合を省略するかどうかを制御します。バックエンドサーバのホスト名がルート証明書のホスト名のいずれかと一致するかどうか制御するには、このブールオプションを ON または OFF に設定します。このオプションを有効にすると、クライアントはサーバの認証を完全に行うことができないので、攻撃に対して脆弱なままとなります。TLS または HTTPS 接続を開始する際、クライアントライブラリは、RFC 2818 に記述された手順を使用して、バックエンドサーバのホスト名をサーバが提供する証明書と照らし合わせます。この照合が行われるのは、certificate_name、certificate_company、

rsoe2 オプション	説明
	certificate_unit のいずれのオプションも指定されていない場合、あるいは、skip_certificate_name_check オプションが有効でない場合のみです。certificate_name、certificate_company、certificate_unit のいずれかが指定されていれば、それらのオプションだけが確認されます。skip_certificate_name_check オプションを有効にすると、ホスト名のチェックが無効になります。ホスト名や IP アドレスは、subjectAltName (サブジェクトの別名、SAN) 拡張、および Common Name (CN) フィールドから取得します。SAN には、ワイルドカードによって複数のホスト名が指定されている場合があります。たとえば、Google 社の証明書には、*.google.com、*.google.ca、*.android.com が指定されているかもしれません。その場合、www.google.ca は有効なホスト名ということになります。 **trusted_certificates** 信頼できるルート証明書のリストを含むファイル。 バックエンドサーバだけを確認するには、このプロパティを *backend_server_public_cert_filename* に設定します。 `trusted_certificates=backend_server_public_cert_filename` Microsoft Windows については、*trusted_certificates* が設定されていない場合、オペレーティングシステムの証明書ストアが使用されます。

rsoe2 オプション	説明
-cs"connection-string"	バックエンドサーバの接続文字列を指定します。接続文字列のフォーマットは、名前と値のペアがセミコロンで区切られたリストです。 **host** バックエンドサーバの IP アドレスまたはホスト名。デフォルトは localhost です。 **port** 必須。バックエンドサーバが受信しているポート。デフォルトは 0 です。 **https** 0 - HTTP (デフォルト) 1 - HTTPS デフォルトでは、Mobile Link は TCP/IP 通信プロトコルを起動します。RSOE で使用する Mobile Link を起動する場合、お客様の RSOE 設定で必要な通信プロトコルを起動してください。たとえば、HTTPS をバックエンドセキュリティとして指定する場合、Mobile Link を HTTPS で起動する必要があります。 https=1 パラメータが -cs オプションに含まれている場合、デフォルトポートは 443 に変更されます。 *https=1* の場合、次のオプションを指定できます。Outbound Enabler が正しいバックエンドサーバに接続されていることを確認するために、certificate_name、certificate_company、certificate_unit のうちの少なくとも 1 つを指定してください。証明書をチェックしないようにするには、skip_certificate_name_check オプションを指定してください。 **identity** サーバ認証で使用される ID ファイルのパスとファイル名。Outbound Enabler とバックエンドサーバとの間で相互に認証された TLS を確立するためのクレデンシャルを提供します。相互認証はバックエンドサーバに必要です。 **identity_password** ID ファイルのパスワードを指定するオプションのパラメータ。このオプションが指定された場合、identity オプションも指定する必要があります。Outbound Enabler とバックエンドサーバとの間で相互に認証された TLS を確立するためのクレデンシャルを提供します。相互認証はバックエンドサーバに必要です。

rsoe2 オプション	説明
	skip_certificate_name_check クライアントライブラリがサーバホスト名とデータベースサーバの証明書ホスト名との照合を省略するかどうかを制御します。バックエンドサーバのホスト名がルート証明書のホスト名のいずれかと一致するかどうか制御するには、このブールオプションを ON または OFF に設定します。このオプションを有効にすると、クライアントはサーバの認証を完全に行うことができないので、攻撃に対して脆弱なままとなります。TLS または HTTPS 接続を開始する際、クライアントライブラリは、RFC 2818 に記述された手順を使用して、バックエンドサーバのホスト名をサーバが提供する証明書と照らし合わせます。この照合が行われるのは、certificate_name、certificate_company、certificate_unit のいずれのオプションも指定されていない場合、あるいは、skip_certificate_name_check オプションが有効でない場合のみです。certificate_name、certificate_company、certificate_unit のいずれかが指定されていれば、それらのオプションだけが確認されます。skip_certificate_name_check オプションを有効にすると、ホスト名のチェックが無効になります。ホスト名や IP アドレスは、subjectAltName (サブジェクトの別名、SAN) 拡張、および Common Name (CN) フィールドから取得します。SAN には、ワイルドカードによって複数のホスト名が指定されている場合があります。たとえば、Google 社の証明書には、*.google.com、*.google.ca、*.android.com が指定されているかもしれません。その場合、www.google.ca は有効なホスト名ということになります。 **trusted_certificates** 信頼できるルート証明書のリストを含むファイル。 バックエンドサーバだけを確認するには、このプロパティを *backend_server_public_cert_filename* に設定します。 `trusted_certificates=backend_server_public_cert_filename` Microsoft Windows については、*trusted_certificates* が設定されていない場合、オペレーティングシステムの証明書ストアが使用されます。
-*d*seconds	バックエンドサーバの活性 ping とバックエンドサーバステータス要求の頻度を設定します。デフォルトは 5 秒です。

rsoe2 オプション	説明
-dl	Relay Server Outbound Enabler コンソールにログメッセージを表示するには、このオプションを使用します。デフォルトでは、冗長レベル 1 と 2 の場合にログメッセージは表示されません。
*-f*farm	バックエンドサーバが属するファームの名前を指定します。
*-id*id	バックエンドサーバに割り当てられている名前を指定します。
*-jsh*number	アイドルジャンクションプールにおけるジャンクションの最大数を指定します。ジャンクションの合計数は、ファーム内の Relay Server の数全体に均等に割り当てられます。デフォルト値は 200 です。
*-jsl*number	アイドルジャンクションプールにおけるジャンクションの最小数を指定します。ジャンクションの合計数は、ファーム内の Relay Server の数全体に均等に割り当てられます。デフォルト値は 10 です。
*-jl*number	ジャンクションの最大数 (アクティブおよびアイドルジャンクションの合計数) を指定します。アクティブおよびアイドルジャンクションの合計数は、ファーム内の Relay Server の数全体に均等に割り当てられます。この設定は、Relay Server 設定ファイルのバックエンドサーバセクションで指定された max_junction 設定によって無効化されます。デフォルト値は 1000 です。
*-o*file	出力メッセージのログの記録先ファイルを指定します。
-oq	起動エラーの発生時にエラーウィンドウが表示されないようにします。
*-os*size	メッセージログファイルの最大サイズを設定します。最小のサイズ制限は 10 KB です。
*-ot*file	指定されたログファイルをトランケートし、そのファイルにメッセージを記録します。
-q	起動時に最小化ウィンドウで実行します。
-qc	完了時にウィンドウを停止します。
-s	Outbound Enabler を停止します。
*-t*token	Relay Server に渡すセキュリティトークンを設定します。
-uc	rsoe2 をシェルモードで起動します。これがデフォルトです。UNIX と Mac OS X に適用されます。 -uc、-ui、-um、-ux のうち 1 つだけを指定してください。-uc を指定すると、RSOE はソフトウェアの以前のリリースと同じ方法で起動されます。

rsoe2 オプション	説明
-ud	rsoe2 をデーモンとして実行するように指示します。UNIX プラットフォームにのみ適用されます。
-ui	使用可能な表示がない場合は、rsoe2 はシェルモードで起動されます。このオプションは、X-Window Server がサポートされている Linux 用です。 -ui を指定すると、RSOE は使用可能な表示を探そうとします。X-Window Server が実行されていなかったなどの理由で、使用可能なディスプレイが見つからなかった場合は、rsoe2 はシェルモードで起動されます。
-ux	Linux についてのメッセージを表示する、RSOE メッセージウィンドウを開きます。 Windows の場合、RSOE メッセージウィンドウは自動的に表示されます。 -ux が指定されている場合、RSOE は使用可能な表示を見つけます。たとえば、DISPLAY 環境変数が設定されていなかったり、X-Window Server が実行されていなかったりしたために、使用可能な表示が見つからなかった場合、RSOE は起動できません。 クワイエットモードで RSOE のメッセージウィンドウを実行するには、-q を使用します。
`-vlevel`	ログを取るために使用する冗長性レベルを設定します。`level` には、0、1、2、またはそれ以上を指定できます (3 以上のレベルは、主にテクニカルサポートに使用されます)。 **0** エラーのみをログに記録します。配備には、このログレベルを使用してください。 **1** セッションレベルロギング。これは、同期セッションの概要です。 **2** 要求ロギング。HTTP 要求のより詳細な表示を提供します。 **3 以上** 詳細ロギング。主に、技術サポートのために使用されます。 レベル 1 と 2 では、メッセージログファイルへの書き込みのみが行われ、表示はされません。すべてのログメッセージを表示するには、-dl オプションを使用します。

備考

Outbound Enabler をサービスとして実行するには、dbsvc ユーティリティを使用します。dbsvc の構文は、Microsoft Windows と UNIX では異なります。UNIX では、実行ファイルのフルパスを -w オプション引数の後にある最初のパラメータとして指定しません。

UNIX では、Outbound Enabler パラメータをコマンドラインでのみ指定します。サービスの設定にコマンド内のコマンドラインオプションを使用しないでください。

例

Microsoft Windows で自動起動する RSOE サービス oes (Outbound Enabler サービス) を設定する場合

```
dbsvc -as -s auto -t rsoe2 -w oes "%SQLANY17%¥Bin64¥rsoe2.exe"
-cr "host=relayserver.sap.com;port=80 " -cs "host=localhost;port=80 " -f FarmName
-id ServerName -t token
```

UNIX で自動起動する RSOE サービス oes (Outbound Enabler サービス) を設定する場合

```
dbsvc -y -a user-account -t rsoe2 -w oes @/full-dir-path/oe.config
```

1.7.2 Outbound Enabler 配備に関する考慮事項

Outbound Enabler を使用するときは、多くの考慮事項を認識している必要があります。

サービスとしての Outbound Enabler

Outbound Enabler をサービスとして設定し、維持するには、サービスユーティリティ (dbsvc) を使用します。

認証

基本認証またはダイジェスト認証を使用します。

1.8 Relay Server ファーム設定の更新

Relay Server ファームの設定は、Relay Server 設定ファイルで定義されます。Relay Server ファーム内のすべての Relay Server は、同じ設定ファイルを共有します。

Relay Server ファーム設定を更新するには、ファーム内の各 Relay Server の Relay Server 設定ファイルを更新します。設定の更新内容は次のとおりです。

- 新しい Relay Server を Relay Server ファームに追加します。
- 新しいバックエンドサーバファームを作成して、Relay Server ファームへのアクセスを許可します。
- 新しいバックエンドサーバを既存のバックエンドサーバファームに追加します。
- Relay Server、バックエンドサーバファーム、またはバックエンドサーバのプロパティを変更します。
- オプションを変更します。

`rstool.jar` 内の AdminChannel または Linux Relay Server 用 Relay Server State Manager を使用して Relay Server 設定のオンライン更新を実行します。更新は、Relay Server を再起動することなしに実行できます。

このセクションの内容:

関連情報

Relay Server ステイトマネージャ (Linux) [18 ページ]

1.8.1 Microsoft Windows 上の Microsoft IIS 用 Relay Server ファーム設定の更新 (コマンドライン)

Relay Server または Relay Server ファームを追加または変更したり、サーバおよびファームのプロパティとオプションを変更するには、Relay Server 設定ファイルを更新します。

前提条件

- 既存の Relay Server ファーム用 Relay Server 設定ファイルが存在します。
- `Admin¥rs.dll` の URL が公開されています。設定スクリプトでは、セキュリティ強化なしにこれを実行します。典型的なセキュリティ強化としては、認証要件の追加などがあります。

手順

1. 新しい Relay Server 設定ファイルに更新を保存します。
2. 各 Relay Server に対して下記のコマンドを実行し、設定の更新を適用します。

```
java -cp %sqlany%¥java¥rstool.jar com.sap.relayserver.AdminChannel -url https://rs.my.com/rs17/admin/rs.dll -uid me -pwd myPassword -setRSConfig new-rs-config-file
```

3. 更新中の Relay Server ファーム内のコンピュータごとに、前述の手順を繰り返します。

結果

Relay Server ファームの設定が更新されます。

1.8.2 Linux 上の Apache 用 Relay Server 設定の更新 (コマンドライン)

Relay Server または Relay Server ファームを追加または変更したり、サーバおよびファームのプロパティとオプションを変更するには、Relay Server 設定ファイルを更新します。

前提条件

Relay Server ファーム用 Relay Server 設定ファイルがすでに存在します。

コンテキスト

Relay Server ステイトマネージャは、サービスを中断することなく、実行中の Relay Server ファームの設定を更新できます。

手順

1. Relay Server 設定ファイルのマスターコピーを更新します。
2. 更新した設定ファイルを Apache インストールディレクトリにある `/modules` ディレクトリにコピーします。
3. `/Apache-install/modules` ディレクトリから、次のコマンドを実行して設定の更新を適用します。

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

   -u オプションは、更新操作を実行することを Relay Server ステイトマネージャに指示します。-f オプションは、更新された設定が含まれる設定ファイルの名前を指定します。
4. 更新中の Relay Server ファーム内のコンピュータごとに、前述の手順を繰り返します。

結果

Relay Server の設定が更新されます。

1.9 Relay Server のロギングとログの管理

Relay Server のログファイルは、情報、警告、およびエラーメッセージを表示します。

情報

現在のセッションについての基本的な情報

警告

発生したアクションについての警告メッセージ

エラー

失敗したアクションについてのエラーメッセージ

Windows の場合、デフォルトの Relay Server ログファイルの名前とディレクトリは、*%SQLANY17%*`¥RelayServer¥IIS¥Bin64¥Log¥rs.log` です。

Relay Server のロギングでは次の機能もサポートされています。

- Relay Server と Outbound Enabler のログは、ミリ秒間隔でタイムスタンプを記録します。タイムスタンプは、RFC 822 local differential format (+/- hhmm) でレポートされます。
- Relay Server と Outbound Enabler は、SAP パスポートヘッダを伝送する HTTP 要求を処理するときに、要求処理の冗長レベルを増加して、パスポートのトレースレベル要件を満たすとともに、関連するログの行にサフィックスを追加して、パスポートのキー情報をリストします。

冗長性

Relay Server のログの冗長性を、次のいずれかのレベルに設定します。

0

エラーのみをログに記録します。配備には、このログレベルを使用してください。これはデフォルトです。

1

要求ロギング。HTTP 要求の概要が省略形式で Relay Server ログファイルに書き込まれます。

2

要求ロギング。HTTP 要求のより詳細な表示を提供します。

3 以上

詳細ロギング。主に、技術サポートのために使用されます。

エラーは指定されたログレベルに関係なく表示され、警告はログレベルが 1 以上の場合に表示されます。Relay Server Record は、冗長性が 1 以上に設定されている場合に Relay Server のログの一部として生成されます。

このセクションの内容:

Relay Server のロギングと SAP パスポート [46 ページ]
SAP パスポートは、クライアントからバックエンドサーバまでの要求をトレースします。

Relay Server Record [47 ページ]

Relay Server Record (RSR) は、Relay Server の処理について簡潔にまとめたもので、内容は、要求、タイミング、アフィニティ情報、要求のステータス、データボリュームなどです。RSR はリレー障害の診断やパフォーマンス特性の調査に使用できます。

Outbound Enabler Record [50 ページ]
Outbound Enabler Record (OER) は、リレー障害の診断やパフォーマンスの調査に使用します。OER は、特定の要求に対する Outbound Enabler の処理について簡潔にまとめたもので、内容は要求、タイミング、重要なデバッグ情報、要求のステータス、データボリュームなどです。

AdminChannel を使用したリモート管理 (Microsoft Windows) [52 ページ]
`rstool.jar` の `AdminChannel` クラスを通じて、Relay Server の設定とログファイルをリモート管理できます。

関連情報

Relay Server ステイトマネージャ (rshost) のコマンドラインの構文 (Linux) [19 ページ]

1.9.1 Relay Server のロギングと SAP パスポート

SAP パスポートは、クライアントからバックエンドサーバまでの要求をトレースします。

SAP パスポートにロギングの冗長性を拡大するディレクティブを入れて各サーバに流すことができます。Relay Server と Relay Server Outbound Enabler はいずれもこのディレクティブに従います。Relay Server または Outbound Enabler の冗長レベルがパスポートで示唆されるレベルよりも高く設定されている場合は、高い方のレベルが有効になります。Relay Server/Outbound Enabler の管理者が Relay Server/Outbound Enabler の設定でロギングの冗長レベルを高く設定すれば、ユーザが低いトレースレベルのパスポートを使用してペイロードロギングを抑制することはできません。

Relay Server は、SAP パスポートのバージョン 2 とバージョン 3 をサポートしています。

パスポートのトレースレベル	RS/OE の冗長レベル	説明
低	1	アクセスレベルロギング
中	4	パケットロギングによるデバッグロギング (Relay Server 側では要求ヘッダロギングも実行)
高	5	ペイロード全体でのデバッグロギング

Relay Server と Outbound Enabler のログのうち、SAP パスポートを含む要求に関連付けられている行には、次の表の "要求の説明" カラムに示すようなパスポートが含まれます。

SAP パスポートのバージョン	要求の説明	ログ行の例
2	R{#SAP-PPK#V2#<Transaction-uuid>}	I. 2015-05-05 14:38:06.898-0400 J{RSTEST01#F0#S0#1} R{1#SAP-PPK#V2#8fa46833ea42b94a8181b5bc8da3a33c001e3700e6331ee1b4b2ac6ff58a2de0#001e3700e6331ee1b4b2a7d926ce4de0#001e3700e6331ee1b4b2ac6ff58a2de0#1} M{Relaying header}
3	R{#SAP-PPK#V3#<Transaction-uuid>#Root-context-uuid#Connection-uuid#Connection-counter}	I. 2015-05-05 14:38:06.898-0400 J{RSTEST01#F0#S0#1} R{2#SAP-PPK#V3#001e3700e6331ee1b4b2ac6ff58a2de0#001e3700e6331ee1b4b2a7d926ce4de0#001e3700e6331ee1b4b2ac6ff58a2de0#1} M{Relaying headers}

1.9.2 Relay Server Record

Relay Server Record (RSR) は、Relay Server の処理について簡潔にまとめたもので、内容は、要求、タイミング、アフィニティ情報、要求のステータス、データボリュームなどです。RSR はリレー障害の診断やパフォーマンス特性の調査に使用できます。

RSR は、冗長性が 1 以上に設定されている場合に Relay Server のログの一部として生成されます。

RSR は、Relay Server ログ内の単一行であり、1 つの HTTP 要求の概要を示す値が含まれています。Relay Server Record の内容を解釈しやすくするため、Relay Server ログファイルには RSR 値を示すヘッダが付けられています。

シンボル	データ型
b:	バイト数。
c:	その他の数。
i:	ID または数値コード。
m:	ミリ秒単位の時間。
x:	16 進数。
name:string	可変長の名前付き文字列値 (Relay Server エラー名、Relay Server エラーパラメータ、および SAP パスポートの情報を含む)。
oe	Outbound Enabler がレポートした要素。
up	Relay Server からバックエンドへの要求 (応答を除く) に関連付けられている要素。

シンボル	データ型
rtp	往復トランスポートおよび、最後の上りパケットと最初の下りパケットの処理に関連付けられている要素。
dn	バックエンドから Relay Server への要求 (応答を除く) に関連付けられている要素。
in	入力の読み込み待機時間。
out	入力の書き込み待機時間。
A.B	B が A の子コンポーネント、サブプロセス、または一部であることを示します。
pkt/packet	Relay Server か Outbound Enabler がジャンクション経由の通信で作成したパケット。

記号は (フィールド名として) 連結され、特定の種類のデータを表します。たとえば、*m:up.out* は、リレー期間内にジャンクション (*m:*) でパケットの書き込み (*up*) に費やした合計時間 (*.out*) を表します。さらにドット記号 (.) は、*out* が *up* のサブプロセスであることを示しています。

フィールド名	データ型	値
flag[0]	アフィニティの決定。	**n=new** 新しいアフィニティを示します **c=continue** アフィニティが確立されているセッションの後続の要求を示します **h=homed** バックエンドサーバが指定された新しいアフィニティを示します **r=renew** 競合があるためにアフィニティ情報が更新されて新しいセッションが開かれていることを示します **x=expired** アフィニティ cookie の期限が切れたことを知らせる信号が Relay Server からクライアントに送信されたことを示します
flag[1]	要求の永続性。	p=persistent、n=non-persistent
flag[2]	要求転送のエンコード。	k=chunked、l=content-length
flag[3]	応答の永続性。	p=persistent、n=non-persistent、u=unknown
flag[4]	応答転送のエンコード。	k=chunked、l=content-length、u=unknown
b:up	要求のサイズ (バイト)。	
b:dn	応答のサイズ (バイト)。	
c:up:pkt	Outbound Enabler に送信された上りの要求パケット数。	
m:up	クライアントからの読み込み、ジャンクションへの書き込み、および要求パケット化のインターリーブに関連した要求リレー時間。	
m:up.in	要求リレー時間内にクライアントからの要求ペイロードを待機した合計時間。	

フィールド名	データ型	値
m:up.out	リレー時間内にジャンクションへのパケットの書き込みに費やした合計時間。	
m:rtp	Relay Server とバックエンドサーバの間の、最後の要求パケット送信から、最初の応答パケット受信までの往復処理時間 (バックエンド処理時間を含む)。	
m:oe:rtp	Outbound Enabler とバックエンドサーバの間の、最後の要求パケット送信から、最初の応答パケット受信までの往復処理時間 (バックエンド処理時間を含む)。	
m:rtp:kpi	Relay Server と Outbound Enabler の間の、最後の要求パケットと最初の応答パケットのリレー処理と転送に要した往復時間 (m:rtp - m:oe.rtp)。	
c:dn.pkt	Outbound Enabler から受信した下りの応答パケット数。	
m:dn	ジャンクションからの読み込みとクライアントへの書き込みのインターリーブに関連した応答リレー時間。	
m:dn.in	応答リレー時間内にジャンクションからの応答パケットの待機に費やされた合計時間。	
m:dn.out	応答リレー時間内にクライアントへの応答ペイロードの書き込み待機に費やされた合計時間。	
m:oe.dn	Outbound Enabler で認識された応答受信時間。この時間は、m:dn.in および m:dn.out の時間と重複します。	
m:close	m:dn の終了時から rs_client 拡張機能の終了までの経過時間。	
i:dn.stts	HTTP 応答のステータス。	
i:err	エラー ID。	
i:warn	警告 ID。	
err	エラーの名前。	
warn	警告の名前。	
oe.err	Outbound Enabler のエラーの名前。	
oe.err.p0	Outbound Enabler からの最初のエラーパラメータ。	
oe.err.p1	Outbound Enabler からの 2 番目のエラーパラメータ。	
oe.err.p2	Outbound Enabler からの 3 番目のエラーパラメータ。	
up.ua	要求のユーザエージェントヘッダ。	

フィールド名	データ型	値
up.uq	`name=value` ペア形式での URL クエリパラメータ。これは、SAP パスポートを使用できない場合に要求内のタグ情報として使用できます。	
up.AfHdr	アフィニティ。	
up.cookie	要求の Cookie ヘッダ。	

label	複合要求のプレフィクス (各行の最後に配置)
	`<label: J{relayserver-host#backend-farm-name#backend-server-name#junction-index} R{request-number}>`
	`<BEFarmName>`
	`<BeServerName>`
	次に例を示します。
	`<11436.4592.F0B0S0R0> <RSTEST02.F0> <S0>`	

Relay Server Record の例を示します。

```
I. 2015-05-01 18:34:26.059-0400 RSR header 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:oe.dn|m:close|
i:dn.stts  i:err i:warn ...other-variable-length-elements...
I. 2015-05-01 18:34:30.286-0400 RSR data nnlnl 1530 1242 | 2 0 0 0 | 150 149 1 | 2
0 0 0 0 | 1 | 200  0 0 <err:> <warn:> <oe.err:> <oe.err.parameters:>
<up.ua:RSTestClient> <up.uq:> <up.AfHdr:> <up.cookie:> <label: J{RSTEST01#F0#S0#0}
R{1}>
```

関連情報

アフィニティ [10 ページ]

1.9.3 Outbound Enabler Record

Outbound Enabler Record (OER) は、リレー障害の診断やパフォーマンスの調査に使用します。OER は、特定の要求に対する Outbound Enabler の処理について簡潔にまとめたもので、内容は要求、タイミング、重要なデバッグ情報、要求のステータス、データボリュームなどです。

OER は、冗長性が 1 以上に設定されている場合に Outbound Enabler のログの一部として生成されます。各 HTTP 要求に OER が 1 行作成されます。

OER は、データ型 (シンボル表現) と値で構成されます。Outbound Enabler Record の内容を解釈しやすくするため、Outbound Enabler ログファイルには OER 値を示すヘッダが付けられています。ヘッダ行は、ファイル内に記述されているシンボルで構成されています。ヘッダ行シンボルの規則は次のとおりです。

シンボル	データ型
b:	バイト数。
i:	ID または数値コード。
m:	ミリ秒単位の時間。
up	バックエンドへの要求のリレー (応答を除く) に関連付けられている要素。
rtp	往復トランスポートおよび、最後の上りパケットと最初の下りパケットの処理に関連付けられている要素。
dn	バックエンドから Relay Server への要求のリレー (応答を除く) に関連付けられている要素。

記号は (フィールド名として) 連結され、特定の種類のデータを表します。たとえば、*m:up* は要求フェーズの期間に対応し、*b:dn* は応答のバイト数に対応します。

フィールド名	データ型
m:up	最初の要求パケットの送信から最後の要求パケットの送信までの要求リレー時間
m:rtp	Outbound Enabler とバックエンドサーバの間の、最後の要求パケット送信から、最初の応答パケット受信までの往復処理時間 (バックエンド処理時間を含む)
m:dn	最初の応答パケットの到達から最後の応答パケットの到達までの応答リレー時間
m:close	最後の応答パケットの到達からガーベジコレクションまたはアフィニティコンテキストの再利用までのアイドル時間
b:up	バックエンドサーバに送信したバイト数
b:dn	バックエンドサーバから受信したバイト数
i:err	エラー ID
err	エラーの名前
err.parameters	エラーのパラメータ

Outbound Enabler Record の例を示します。

```
I. 2015-05-01 18:34:26.158-0400 T{000022a4} J{RS-host#farm#server#junction-index}
R{request-number} M{OER HEADER m:up m:rtp m:dn m:close | b:up b:dn | i:err err
err.parameters}
I. 2015-05-01 18:34:30.294-0400 T{00000480} J{RSTEST01#F0#S0#0} R{1} M{OER DATA 0
149 0 0 | 1575 1242 | 0 OK ()}
```

関連情報

アフィニティ [10 ページ]

1.9.4 AdminChannel を使用したリモート管理 (Microsoft Windows)

`rstool.jar` の `AdminChannel` クラスを通じて、Relay Server の設定とログファイルをリモート管理できます。

コマンドラインの使用方法:

```
java com.sap.relayserver.AdminChannel [{options}]...
```

オプション	説明
-url `rsAdminUrl`	rs_admin 拡張機能へのポインタ。
-uid `user` -pwd `password`	rs_admin へのアクセスを目的とした HTTP 認証用のクレデンシャルを提供。
-ping	rs_admin 拡張機能への ping を実行。
-getRSConfig	Relay Server の設定を取得。
-setRSConfig `new-config-file`	Relay Server の設定の更新およびバックアップ。
-hello	管理プロトコルバージョンのネゴシエーションと ping を実行。
-archiveRSLog	現在オンラインの Relay Server ログファイルのトランケートとアーカイブ。
-xol `outFile`*none* | `beginTime`*none* | `endTimen`Regex "`regex`"...	ローカル Outbound Enabler のアーカイブログかオンラインログからログ行を抽出。Relay Server が稼動している必要はありません。この機能が実行するのはローカルの抽出だけです。 • `*Time` は yyyy-MM-dd HH:mm:ss.SSS 形式のローカルタイムスタンプ。 • `nRegex` は正規表現。
-xrl `outFile`*none* | `beginTime`*none* | `endTimen`Regex "`regex`"...	Relay Server のアーカイブログかオンラインログからリモートでログ行を抽出。 • `*Time` は yyyy-MM-dd HH:mm:ss.SSS 形式のローカルタイムスタンプ。 • `nRegex` は正規表現。
?| -?| /?| -h | /h	この使用方法の説明を出力。

例

```
java.exe -cp rstool.jar com.sap.relayserver.AdminChannel -url https://rs.my.com/
rs17/admin/rs.dll -uid me -pwd passwd -hello
java.exe -cp rstool.jar com.sap.relayserver.AdminChannel -url https://rs.my.com/
rs17/admin/rs.dll -uid me -pwd passwd -xrl rr.xrl none none 1 "RSR (element|
header|data)"
```

1.10 Mobile Link で使用する Relay Server

クライアントと Relay Server ファームを接続するには、Mobile Link を使用します。

Mobile Link とともに Relay Server を使用するには、次のタスクを実行します。

1. Mobile Link クライアントに向けて Relay Server を設定します。
2. Outbound Enabler を実行します。
3. ステータスページを使用して設定をテストします。
4. Mobile Link クライアントを実行します。

このセクションの内容:

関連情報

Relay Server Outbound Enabler の構文 [33 ページ]
Relay Server のステータスページ [10 ページ]

1.10.1 Mobile Link への Relay Server ファームの設定 (コマンドライン)

Mobile Link クライアントがファームに接続できるよう、設定ファイルを適切に設定し、配備します。

手順

1. Relay Server 設定ファイルを作成します。設定ファイルには `rs.config` という名前を付ける必要があります。
2. Relay Server を実行している 2 つのコンピュータに、Relay Server コンポーネントとともに `rs.config` ファイルを配備します。
3. 各バックエンドサーバ上で、Mobile Link サーバと Outbound Enabler を起動します。

結果

すべてのサーバと Outbound Enablers が実行を開始すると、Mobile Link クライアントをファームに接続できます。

例

ABC 社がモバイルアプリケーションを開発し、モバイルアプリケーションにサービスを提供する配備ランタイムの設定を希望していると仮定します。モバイル配備は、最初は 10000 個のデバイスから構成され、将来さらに増加します。顧客は、現在の負荷を処理でき、将来より多くのモバイル配備を処理できるように拡張しやすい、フォールトトレラントで負荷分散された環境を要求しています。モバイルアプリケーションのデータ同期特性に基づいて、顧客は、2 つの Mobile Link サーバ、2 つの Relay Server、および 1 つのロードバランサといった設定が必要であると判断しました。

- 各 Relay Server は、専用のコンピュータ上に配備されます。ホスト名が rs1.abc.com と rs2.abc.com である 2 つのコンピュータを使用します。
- 各 Mobile Link サーバは、専用のコンピュータ上に配備されます。2 つの Mobile Link サーバは、ml1 と ml2 という名前が割り当てられ、abc.mobilink という名前のバックエンドサーバファームに属しています。
- ロードバランサは、ホスト名 www.abc.com を使用して指定できます。
- 最高レベルのセキュリティを得るため、Relay Server に接続するすべてのクライアントと Outbound Enabler では HTTPS が使用されます。すべての Web サーバには既知の認証局 (CA) による証明書が配置され、すべてのバックエンドサーバコンピュータには対応する信用されたルート証明書が標準の証明書ストアに保存されています。

この例では、Relay Server の Microsoft IIS バージョンを使用します。

1. Relay Server 設定ファイルを作成します。
   設定ファイルは `rs.config` と名付けられます。このシナリオでは、次の設定ファイルを使用しています。

   ```
   #
   # 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 will connect to the Relay Server farm
   #
   [backend_server]
   farm = abc.mobilink
   id = ml1
   token = mltoken1
   [backend_server]
   farm = abc.mobilink
   id = ml2
   token=mltoken2
   ```

2. Relay Server を実行している 2 つのコンピュータに、Relay Server コンポーネントとともに `rs.config` ファイルを配備します。

3. ID ml1 で Mobile Link サーバを実行しているコンピュータで、下記のコマンドを実行します。

```
rsoe2 -f abc.mobilink -id ml1 -t mltoken1 -cr
"host=www.abc.com;port=443;https=1" -cs "host=localhost;port=80"
```

```
mlsrv17 -x http(port=80)
```

4. ID ml2 で Mobile Link サーバを実行しているコンピュータで、下記のコマンドを実行します。

```
mlsrv17 -x http(port=80)
```

```
rsoe2 -f abc.mobilink -id ml2 -t mltoken2 -cr
"host=www.abc.com;port=443;https=1" -cs "host=localhost;port=80"
```

5. すべてのサーバと Outbound Enabler が実行中になったら、Mobile Link クライアントは次の接続情報を使用してファームに接続できます。

   HTTPS

   protocol

   host

   www.abc.com

   url_suffix

   /rs/client/rs.dll/abc.mobilink

関連情報

Relay Server 設定ファイル [20 ページ]

1.11 Relay Server ファームへのクライアント接続

いったん Relay Server ファームが設定されると、クライアントはそれに接続できます。

クライアントは下記の URL を使用して接続します。

```
http://Relay-Server-client-extension-URL/farm-name
```

オプション

オプション	説明
`Relay-Server-client-extension-URL`	Windows 上の Microsoft IIS の場合`<domain name><relayserver.sap.com>/rs/client/rs.dll` Linux 上の Apache の場合`<domain name>/cli/iarelayserver`
`farm-name`	この値は、Relay Server がクライアント要求を転送するバックエンドファームを示します。

SQL Anywhere Mobile Link クライアントの接続の例

次のコマンドは、SQL Anywhere HTTP を介して、Mobile Link クライアントとサーバファーム F1 とを接続します。

```
-e "ctp=http;
     adr='host=relayserver.sap.com;
        url_suffix=/rs17/client/rs.dll/F1'"
```

Ultra Light Mobile Link クライアント接続の例

ULSyncParms クラスプロパティは、Ultra Light Mobile Link クライアントとサーバファーム F1 とを接続します。

- ストリームタイプ (HTTP または HTTPS) を設定します。
- ストリームパラメータを次の値に設定します。

  ```
  "host=my_rs.my_corp.com;url_suffix=/rs17/client/rs.dll/F1"
  ```

1.12 このマニュアルの印刷、再生、および再配布

次の条件に従うかぎり、このマニュアルの全部または一部を使用、印刷、再生、配布することができます。

1. ここに示したものとそれ以外のすべての版権と商標の表示をすべてのコピーに含めること。
2. マニュアルに変更を加えないこと。
3. SAP 以外の人間がマニュアルの著者または情報源であるかのように示す一切の行為をしないこと。

ここに記載された情報は事前の通知なしに変更されることがあります。

重要免責事項および法的情報

コードサンプル

この文書に含まれるソフトウェアコード及び / 又はコードライン / 文字列 (「コード」) はすべてサンプルとしてのみ提供されるものであり、本稼動システム環境で使用することが目的ではありません。「コード」は、特定のコードの構文及び表現規則を分かりやすく説明及び視覚化することのみを目的としています。SAP は、この文書に記載される「コード」の正確性及び完全性の保証を行いません。更に、SAP は、「コード」の使用により発生したエラー又は損害が SAP の故意又は重大な過失が原因で発生させたものでない限り、そのエラー又は損害に対して一切責任を負いません。

アクセシビリティ

この SAP 文書に含まれる情報は、公開日現在のアクセシビリティ基準に関する SAP の最新の見解を表明するものであり、ソフトウェア製品のアクセシビリティ機能の確実な提供方法に関する拘束力のあるガイドラインとして意図されるものではありません。SAP は、この文書に関する一切の責任を明確に放棄するものです。ただし、この免責事項は、SAP の意図的な違法行為または重大な過失による場合は、適用されません。さらに、この文書により SAP の直接的または間接的な契約上の義務が発生することは一切ありません。

ジェンダーニュートラルな表現

SAP 文書では、可能な限りジェンダーニュートラルな表現を使用しています。文脈により、文書の読者は「あなた」と直接的な呼ばれ方をされたり、ジェンダーニュートラルな名詞 (例: 「販売員」又は「勤務日数」) で表現されます。ただし、男女両方を指すとき、三人称単数形の使用が避けられない又はジェンダーニュートラルな名詞が存在しない場合、SAP はその名詞又は代名詞の男性形を使用する権利を有します。これは、文書を分かりやすくするためです。

インターネットハイパーリンク

SAP 文書にはインターネットへのハイパーリンクが含まれる場合があります。これらのハイパーリンクは、関連情報を見いだすヒントを提供することが目的です。SAP は、この関連情報の可用性や正確性又はこの情報が特定の目的に役立つことの保証は行いません。SAP は、関連情報の使用により発生した損害が、SAP の重大な過失又は意図的な違法行為が原因で発生したものでない限り、その損害に対して一切責任を負いません。すべてのリンクは、透明性を目的に分類されています (http://help.sap.com/disclaimer を参照)。

go.sap.com/registration/contact.html