



Relay Server

バージョン 12.0.1

2012 年 1 月

バージョン 12.0.1
2012 年 1 月

Copyright © 2012 iAnywhere Solutions, Inc. Portions copyright © 2012 Sybase, Inc. All rights reserved.

iAnywhere との間で書面による合意がないかぎり、このマニュアルは現状のまま提供されるものであり、その使用または記載内容の誤りに対して一切の責任を負いません。

次の条件に従うかぎり、このマニュアルの一部または全体を使用、印刷、複製、配布することができます。1) マニュアルの一部または全体にかかわらず、ここに示したものとそれ以外のすべての著作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) iAnywhere 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。

iAnywhere®、Sybase®、<http://www.sybase.com/detail?id=1011207> に示す商標は Sybase, Inc. またはその関連会社の商標です。® は米国での登録商標を示します。

このマニュアルに記載されているその他の会社名と製品名は各社の商標である場合があります。

目次

はじめに	v
Relay Server の概要	1
Relay Server のアーキテクチャー	1
Relay Server の配備	7
Windows Server 2003 上の Microsoft IIS 6.0 への Relay Server コンポー ネントの配備	7
Windows Server 2008/Windows Server 2008 R2 上の Microsoft IIS 7.0 または 7.5 への Relay Server コンポーネントの配備	11
Linux 上の Apache への Relay Server コンポーネントの配備	15
Relay Server ステイトマネージャー	21
サービスとしての Relay Server ステイトマネージャー	21
Relay Server ステイトマネージャーの自動起動	22
カスタマイズオプションを指定した Relay Server ステイトマネージャー の自動起動	22
Relay Server ステイトマネージャーのコマンドラインの構文	23
Relay Server 設定ファイル	25
Relay Server セクション	25
バックエンドファームセクション	26
バックエンドサーバーセクション	29
オプションセクション	30
Relay Server 設定ファイルのフォーマット	31
Outbound Enabler	33
サービスとしての Outbound Enabler	40
Relay Server ファーム設定の更新	43

Windows 上の Microsoft IIS 用の Relay Server 設定の更新	44
Linux 上の Apache 用の Relay Server 設定の更新	44
Sybase Central 用の Relay Server プラグイン	47
Sybase ホストのリレーサービス	53
Sybase ホストのリレーサービスの使用	53
Mobile Link で使用する Relay Server	55
クライアントから Relay Server ファームへの接続	55
サンプルシナリオ	56
索引	59

はじめに

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、Web サーバーを通じて通信する Afaria、Mobile Office、Mobile Link、SQL Anywhere、Unwired Server、Sybase Unwired Platform サーバーとモバイルデバイス間で、安全な通信を実現します。

Relay Server の概要

Relay Server は、Web サーバー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ー間で、安全で負荷分散された通信を実現します。サポートされるバックエンドサーバーには、Afaria、Mobile Office、Mobile Link、SQL Anywhere、Unwired Server、Sybase Unwired Platform があります。Relay Server には次の機能があります。

- バックエンドサーバーと通信するモバイルデバイス用の共通の通信アーキテクチャー。
- バックエンドサーバー用の負荷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。
- 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される方法で、モバイルデバイスとバックエンドサーバー間の通信に役立つ方法。

Relay Server のアーキテクチャー

Relay Server の配備環境は、次の要素で構成されます。

- 企業 LAN 内で実行されているバックエンドサーバーと通信する必要があるクライアントアプリケーションやサービスを実行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、オプションの負荷分散装置。
- 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役割を持っている少なくとも 1 つのバックエンドサーバー。次のバックエンドサーバーは、Relay Server とともに使用できます。
 - Afaria
 - Mobile Office
 - Mobile Link
 - SQL Anywhere
 - Unwired Server
 - Sybase Unwired Platform

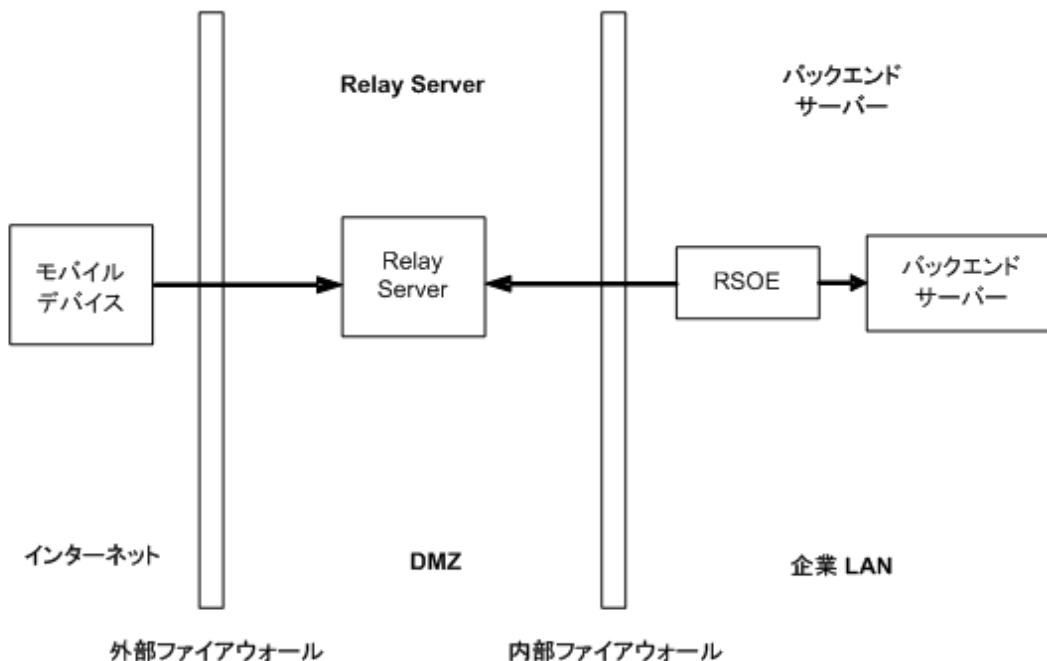
注意

Relay Server は、明確に定義された HTTP 要求と応答により通信を行う特定のバックエンドサーバーとクライアントでテストされます。SQL Anywhere を Web サーバー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、確実に Relay Server で動作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ー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「プラットフォーム別 SQL Anywhere コンポーネント」ページを参照してください。 http://www.iAnywhere.jp/tech/1061806-os_components.html を参照してください。

- Relay Server Outbound Enablers (RSOE) は、通常はバックエンドサーバーごとに1つしかありませんが、いくつかある場合もあります。Outbound Enabler は、バックエンドサーバーと Relay Server ファームとの間のあらゆる通信を管理します。

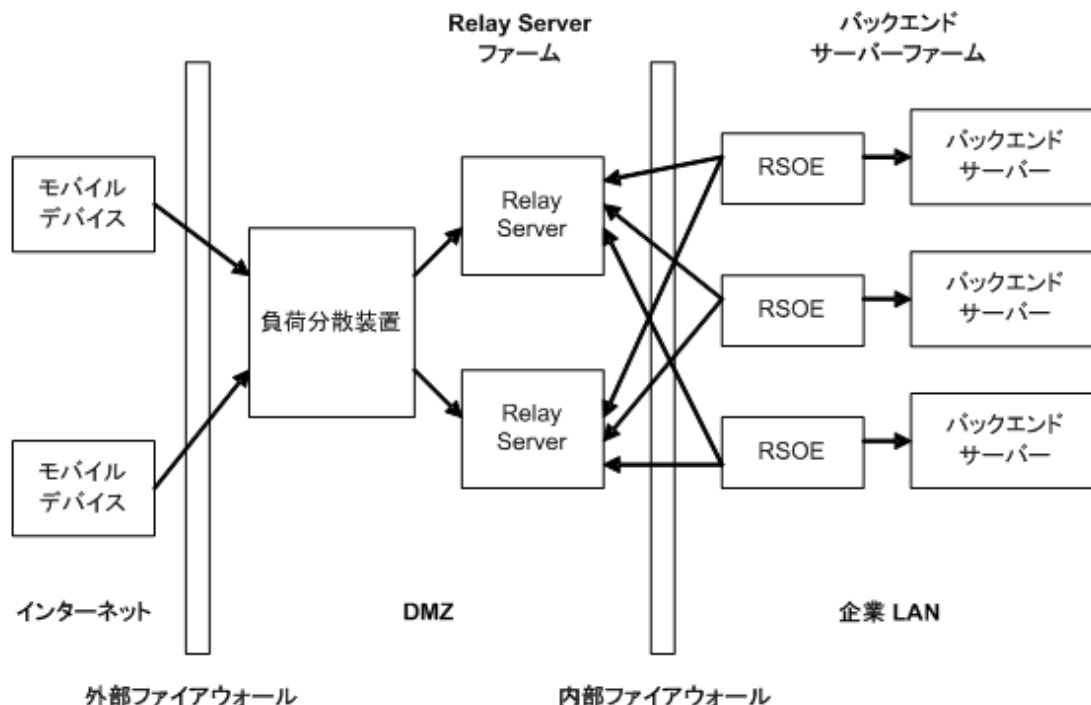
次の図は、単一の Relay Server を持つ Relay Server のアーキテクチャーを示しています。



注意

どのバックエンドサーバー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「プラットフォーム別 SQL Anywhere コンポーネント」ページを参照してください。 http://www.iAnywhere.jp/tech/1061806-os_components.html を参照してください。

次の図は、単一の Relay Server ファームと単一のバックエンドサーバーファームを持つ、より複雑なシステムの Relay Server のアーキテクチャーを示しています。

**注意**

どのバックエンドサーバー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「プラットフォーム別 SQL Anywhere コンポーネント」ページを参照してください。 http://www.ianywhere.jp/tech/1061806-os_components.html > を参照してください。

Relay Server は Web 拡張機能、ステータス情報を保持するバックグラウンドプロセス、Web サーバーのセットで構成されます。

Relay Server は Web サーバーで実行されている 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。これによって、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。

Relay Server は、クライアント拡張機能、サーバー拡張機能の 2 つの Web 拡張機能が含まれます。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。サーバー拡張機能は、バックエンドサーバーに代わって Outbound Enabler によって行われた要求を処理します。

共有メモリとセキュリティ

Relay Server は、クライアントとサーバー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 サーバーが HTTPS を HTTP に復号化し、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、WAP (ワイヤレスアプリケーションプロトコル) 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ー上の不良プロセスから守るには 2 つの方法があります。最初の方法は、エンドツーエンド暗号化をサポートしているクライアントとバックエンドサーバーを使用することです。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 サーバーとオペレーティングシステム (OS) 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 サーバーと OS を強化することです。この強化には、Web サーバーコンピューターの OS 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、コンピューター/VM を Relay Server と Web サーバーの実行だけに制限し、他は行わないことが理想的です。この目標は、コンピューター上でのプロセスの数を最小限にしなが、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ための強化を行うことです。

Relay Server ファーム

Relay Server ファームは、任意の数の Relay Server とフロントエンドの負荷分散装置から成ります。単一の Relay Server を持つ Relay Server ファームを設定することもできます。この場合、負荷分散装置は必要ありません。また、モバイルデバイスは Relay Server に直接接続できます。

バックエンドサーバーファーム

バックエンドサーバーファームは、同種のバックエンドサーバーで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ーファームを指定する必要があります。

負荷分散装置

負荷分散装置は、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

Mobile Link Relay Server Outbound Enabler は、バックエンドサーバーと同じコンピューター上で実行されます。主な機能は、バックエンドサーバーに代わって 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することです。Relay Server Outbound Enablers (RSOE) は、通常はバックエンドサーバーごとに 1 つしかありませんが、いくつかある場合があります。

参照

- [「Outbound Enabler」 33 ページ](#) を参照してください。

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、Web サーバー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 には、Web サーバーと組み合わせて安全な通信を実現するための次の機能が備えられています。

- サーバー側の証明書
- クライアント側の証明書
- バックエンドサーバーとバックエンドファームの設定
- RSOE MAC アドレスフィルタリングとトークン認証
- クライアントの暗号化テクノロジー(プロトコルレベルの暗号化)

サーバー側の証明書

Relay Server と通信するクライアントは、サーバー側の証明書を使用して、Relay Server を実行している Web サーバーが信頼できるかどうかを確認できます。クライアントに格納されているルート証明書を使用して、Web サーバー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。

クライアント側の証明書

Web サーバーは、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が信頼できるかどうかを確認できます。Web サーバーコンピューターの証明書マネージャー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ー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 Relay Server のピアリストを定義します。ファーム環境で稼働している場合は、バックエンドファームとバックエンドサーバーも設定します。環境内の各 Relay Server では、*rs.config* ファイルのコピーを保持する必要があります。

バックエンドファームとバックエンドサーバーを設定することによって、Relay Server が、Relay Server を設定済みのコンピューターとのみ通信することが保証されます。Relay Server が設定されていないコンピューターと通信しようとする、拒否されます。

バックエンドファームは、クライアントと RSOE からの要求を受け入れるときに通信セキュリティのレベルを指定するように設定できます。バックエンドファームでは、オプション `client_security` と `backend_security` を使用して、確立可能な通信のタイプを指定できます。このオプションは、次のように指定します。

client_security=on|off **On** は、クライアントが HTTPS を使用して接続する必要があることを示します。**Off** は、クライアントが HTTP を使用して接続する必要があることを示します。この設定はオプションです。値が指定されていない場合、クライアントは HTTP、HTTPS のどちらを使用しても接続できます。

backend_security=on|off **On** は、RSOE が HTTPS を使用して接続する必要があることを示します。**Off** は、RSOE が HTTP を使用して接続する必要があることを示します。この設定はオプ

ションです。値が指定されていない場合、RSOE は HTTP、HTTPS のどちらを使用しても接続できます。

RSOE MAC アドレスフィルタリングとトークン認証

RSOE では、1) 起動フェーズ、2) 準備完了フェーズ、3) 作業フェーズの 3 段階を使用して、バックエンドサーバーと Relay Server 間の接続が確立されます。

rs.config ファイルのバックエンドサーバーセクションでは、バックエンドファームに存在する各サーバーが、ID および関連するファーム名を使用して設定されます。ID はサーバー名に対応します。Relay Server には、RSOE で稼働しているコンピューターの 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ー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ーの 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ーセクションでは、バックエンドサーバー接続を認証するために、Relay Server で使用され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。

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ー間でデータが交換されるたびに再び暗号化されます。これは WAP ギャップと呼ばれます。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することをおすすめします。Mobile Link エンドツーエンド暗号化機能には、Mobile Link、Ultra Light、QAnywhere クライアントと Mobile Link サーバー間でのプロトコルレベルの暗号化が備えられています。RSA と ECC の暗号化タイプがどちらもサポートされています。TLS セキュリティは、エンドツーエンド暗号化と併用できません。

Outbound Enabler の HTTP 受信ポート

セキュリティのためにスタンドアロン Outbound Enabler を使用する場合は、バックエンドサーバーの HTTP 受信ポートは明示的にループバック IP アドレス (127.0.0.1) のみにバインドされる必要があります。

参照

- 「Relay Server 設定ファイル」 25 ページ
- Windows IIS <http://www.sybase.com/detail?id=1059277>
- Apache <http://www.sybase.com/detail?id=1065869>
- 「エンドツーエンド暗号化」『SQL Anywhere サーバー データベース管理』

Relay Server の配備

Windows Server 2003 上の Microsoft IIS 6.0 への Relay Server コンポーネントの配備

Relay Server ファームの各コンピューターに Relay Server ファイルを配備してから IIS 6.0 で Relay Server を実行する必要があります。

前提条件

Relay Server コンポーネントは、SQL Anywhere 12 のインストールの一部としてインストールされます。インストール処理では、Relay Server が実行されるコンピューターに、必要なすべてのファイルが自動的に配備されます。

デフォルトでは、すべてのファイルが %SQLANY12% にインストールされ、コンピューターのビット数に基づいています。

- %SQLANY12%\Bin32 と %SQLANY12%\Bin64 は、管理用の DLL と実行プログラムに使用されます。
- %SQLANY12%\RelayServer\IIS\Bin32 と %SQLANY12%\RelayServer\IIS\Bin64 は、該当するフォルダー (*Admin*、*Client*、*Monitor*、*Server* など) にある Relay Server 固有のファイルに使用されます。*Server* フォルダーには、*rshost.exe* と *rs.config* ファイルが含まれています。

内容と備考

対話型クイック設定機能

対話型クイック設定機能である *rs-setup.bat* は、この手順の代替手段として提供されます (クイック設定機能の詳細な説明については、*readme* ファイルを参照してください)。*rs-setup.bat* は、%SQLANY12%\RelayServer\IIS\quicksetup_iis6 ディレクトリに配置され、次のタスクを実行します。

1. デモアプリケーションを作成します
2. クイックリファレンスガイドを生成します

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

- *rs_client.dll*
- *rs_server.dll*
- *rs_monitor.dll*
- *rshost.exe*
- *dbngen12.dll*
- *dbsvc.exe*
- *dbfhide.exe*
- *dbtool12.dll*
- *dblib12.dll*
- *dbicu12.dll*
- *dbicudt12.dll*
- *dbsupport.exe*
- *dbghelp.dll*

サポートされている IIS のバージョンについては、http://www.iAnywhere.jp/tech/1061806-os_components.html を参照してください。

IIS 上の Relay Server 用設定スクリプトは、`%SQLANY12%\¥RelayServer¥IIS` ディレクトリにあります。

◆ Windows Server 2003 上の Microsoft IIS 6.0 への Relay Server コンポーネントの配備

1. Microsoft IIS マネージャーで、既定の Web サイトの下に Relay Server に使用する *rs* という仮想ディレクトリを作成します。仮想ディレクトリの物理的なロケーションは `%SQLANY12%\¥RelayServer¥IIS¥Bin.XX` であり、ここで *XX* は IIS サーバーのビット数に従い、32 と 64 のいずれかになります。
2. 次のガイドラインに従って、Relay Server 設定ファイル *rs.config* を作成します。
 - ファイルには、次の 4 つのセクションが必要です。
 - オプションセクション
 - 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ーセクション
 - 各セクションはセクションタグで開始します。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。
 - 各セクションに適切なプロパティを追加します。プロパティは、等号の左側にプロパティ名、等号の右側にその値を指定することによって定義されます。たとえば、*property name = value*。
 - 設定ファイルには、7 ビット ASCII 文字以外を含めることはできません。
3. アプリケーションプールを作成します。
 - a. Microsoft IIS マネージャーコンソールを起動します。
 - b. [アプリケーションプール] を右クリックし、RS_POOL などの新しいアプリケーションプールを作成します。

- c. 作成したアプリケーションプールのプロパティを編集します。
 - i. [リサイクル] タブをクリックし、すべてのリサイクルオプションをオフにします。
 - ii. [パフォーマンス] タブをクリックし、次の操作を行います。
 - A. [アイドルなワーカー プロセスの解放までの待ち時間] をオフにします。
 - B. ワーカープロセスの数をプロセッサコアの合計数に設定します。使用率やパフォーマンスの傾向に応じてこの値をさらに調整することができます。詳細については、Web ガーデンサイズに関する Microsoft IIS パフォーマンスの記述を参照してください。
4. デフォルト Web サイトの接続タイムアウトプロパティを最小の 60 秒に設定します。この値のデフォルト値は 120 秒であり、これで十分です。
5. IIS マネージャーコンソールを使用して rs のプロパティを編集し、Relay Server の Web 拡張機能を有効にします。
 - a. [ディレクトリ] タブをクリックし、次の操作を行います。
 - i. 実行のアクセス許可を [スクリプトおよび実行可能ファイル] に設定します。
 - ii. [アプリケーションの設定] の下の [作成] をクリックします。関連付けるアプリケーションプールとして手順 3 で作成したアプリケーションプールを選択します。
 - b. [ディレクトリ セキュリティ] タブをクリックし、次の操作を行います。
 - i. [認証とアクセス制御] で [編集] をクリックします。
 - ii. 匿名アクセスを有効にし、Administrators グループに属するユーザー名とパスワードを設定します。

または、設定を組み込みユーザー `IUSR_%computername%` のままにして、Microsoft IIS メタベースへのアクセスパーミッションを付与するために次のコマンドを実行する方法もあります。

```
C:\Windows\Microsoft.Net\Framework\<Version>\aspnet_regiis.exe -ga IUSR_%computername%
```
6. Microsoft IIS マネージャーの [Web サーバー拡張] で、`rs_server.dll`、`rs_client.dll`、`rs_monitor.dll` を新しい Web サービス拡張機能として追加します。拡張名には ISAPI を指定し、DLL では拡張ステータスを Allowed に設定する必要があります。
7. パフォーマンスのヒントを参考にして、最適なパフォーマンスを得ることができるようにします。
8. 次のようなコマンドラインを使用して、Relay Server ステイトマネージャーをサービスとして起動します。

```
dbsvc -as -s auto -t rshost -w RelayServer "%SQLANY12%\RelayServer\IIS\Bin\XX\Server\%rshost.exe" -q -qc -f "%SQLANY12%\RelayServer\IIS\Bin\XX\Server\rs.config" -o "c:\temp\ias_relay_server.log"
```


注意

ステイトマネージャーをサービスとして起動することをおすすめします。ただし、Relay Server によって自動的に起動することもできます。

9. Windows 上で Microsoft IIS 6.0 用の Relay Server 設定を更新します。
 - a. 更新中の Relay Server ファームに属するコンピューターごとに、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 `%SQLANY12%\RelayServer\IIS\Bin\XX\Server` ディレクトリにコピーします。自動起動を使用する場合は、設定ファイルを `rs.config` という名前にしてください。
 - b. `%SQLANY12%\RelayServer\IIS\Bin\XX\Server` 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. 更新中の Relay Server ファーム内のコンピューターごとに、前述の手順を繰り返します。

注意

IIS で Relay Server を設定した後、IIS サーバーまたはコンピューターを再起動することをおすすめします。

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ーに配備されます。

次の手順

なし

参照

- 「Relay Server 設定ファイル」 25 ページ
- 「パフォーマンスに関するヒント」 10 ページ
- 「Relay Server ステイトマネージャー」 21 ページ
- 「サービスとしての Relay Server ステイトマネージャー」 21 ページ
- 「Relay Server ステイトマネージャーのコマンドラインの構文」 23 ページ
- 「ファイル非表示ユーティリティ (dbfhide)」 39 ページ

パフォーマンスに関するヒント

Relay Server を Windows 上の Microsoft IIS に配備するときは、次の点に注意してください。

- Relay Server Web 拡張機能は、ASP.NET を使用しません。ASP.NET ISAPI フィルターを削除すると、パフォーマンスが向上します。標準の Microsoft IIS インストールでは、デフォルトでこのフィルターがオンになっています。フィルターをオフにするには、次の手順に従います。

1. Microsoft IIS マネージャーコンソールを起動します。

2. [既定の Web サイト] のプロパティを編集します。
 3. [ISAPI フィルター] タブで、ASP.NET フィルターを削除します。
- パフォーマンスを向上させるために、Microsoft IIS アクセスログをオフにすることができます。アクセスログをオフにするには、次の手順に従います。
 1. Microsoft IIS マネージャーコンソールを起動します。
 2. [既定の Web サイト] で `ias_relay_server` ディレクトリのプロパティを編集します。
 3. [ディレクトリ] タブで、[ログ アクセス] セクションをクリアします。
 - 運用環境では、Relay Server 設定ファイルで Relay Server の冗長性を 0 に設定できます。これにより、高負荷時のパフォーマンスが向上します。
 - Relay Server には、Web ガーデンサイズに関する制限がありません。1 つのワーカープロセスがすべての Outbound Enabler とすべてのクライアントからの要求を処理できます。ただし、プロセスで作成できるスレッド数は、スレッドの作成に使用できる残りのプロセスヒープ領域によって制限されます。Microsoft IIS によって作成されるスレッドのスタックサイズは、256 K です。コンピューターに十分なリソースがあり、サーバーが数千もの同時要求による負荷を受けているときに同時実行の制限に達するおそれがある場合は、プロセス数を増やすことを試してみてください。

Windows Server 2008/Windows Server 2008 R2 上の Microsoft IIS 7.0 または 7.5 への Relay Server コンポーネントの配備

Relay Server ファームの各コンピューターに Relay Server ファイルを配備してから IIS 7.0 または 7.5 で Relay Server を実行する必要があります。

前提条件

Microsoft IIS ISAPI Extensions 機能がインストールされます。

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 `%SQLANY12%` にインストールされ、コンピューターのビット数に基づいています。

- `%SQLANY12%\Bin32` と `%SQLANY12%\Bin64` は、管理用の DLL と実行プログラムに使用されます。
- `%SQLANY12%\RelayServer\IIS\Bin32` と `%SQLANY12%\RelayServer\IIS\Bin64` は、該当するフォルダー (*Admin*、*Client*、*Monitor*、*Server* など) にある Relay Server 固有のファイルに使用されます。*Server* フォルダーには、`rshost.exe` と `rs.config` ファイルが含まれています。

内容と備考

対話型クイック設定機能

対話型クイック設定機能である *rs-setup.bat* は、この手順の代替手段として提供されます (クイック設定機能の詳細な説明については、*readme* ファイルを参照してください)。 *rs-setup.bat* は、`%SQLANY12%/RelayServer/IIS/quicksetup_iis7%` ディレクトリに配置され、次のタスクを実行します。

1. IIS7 をインストールして、必要な IIS7 機能をオンにします
2. Relay Server 用の IIS7 を設定します
3. デモアプリケーションを作成します
4. クイックリファレンスガイドを生成します

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

- *rs_client.dll*
- *rs_server.dll*
- *rs_monitor.dll*
- *rshost.exe*
- *dbngen12.dll*
- *dbsvc.exe*
- *dbfhide.exe*
- *dbtool12.dll*
- *dblib12.dll*
- *dbicu12.dll*
- *dbicudt12.dll*
- *dbsupport.exe*
- *dbghelp.dll*

サポートされている IIS のバージョンについては、http://www.iAnywhere.jp/tech/1061806-os_components.html を参照してください。

Relay Server での IIS 7 設定スクリプトは、`%SQLANY12%\RelayServer\IIS\iis7_setup.txt` ディレクトリにあります。

◆ Windows Server 2008 / Windows Server 2008 R2 上の Microsoft IIS 7.0 または 7.5 への Relay Server コンポーネントの配備

1. `c:\Windows\System32\inetrv\config` フォルダにある IIS 設定ファイル *applicationHost.config* をバックアップします。
2. Relay Server のアプリケーションプールを追加するには、*applicationHost.config* ファイルを編集して、`<system.applicationHost>` » `<applicationPools>` セクションに次のコードを追加します。

```
<add name="RelayServer" queueLength="65535" autoStart="true" 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>
```

注意

以降の手順では、*applicationHost.config* ファイルで *%SQLANY12%\¥RelayServer¥IIS¥BinXX* ディレクトリを *%rs_dir%* としています。ただし、環境変数の展開は IIS 設定ファイルのすべてのセクションで完全にサポートされているわけではないので、*%rs_dir%* 変数を *applicationHost.config* ファイル内に追加するときには、完全に展開しておく必要があります。

3. Relay Server アプリケーションをデフォルトサイトに追加するには、*applicationHost.config* ファイルを編集して、**<system.applicationHost>** » **<applicationPools>** » **<sites>** » **<site name="Default Web Site">** セクションに次のコードを追加します。

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

4. Relay Server ISAPI 拡張機能を追加するには、*applicationHost.config* ファイルを編集して、**<system.webServer>** » **<security>** » **<isapiCgiRestriction>** セクションに次のコードを追加します。

```
<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. Relay Server 拡張機能へのアクセスを可能にするには、*applicationHost.config* ファイルを編集して次のコードを **<configuration>** セクションに追加します。

```
<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>
```

注意

Relay Server は、前述の手順を使用して匿名アクセス用に設定されます。業務の要件に基づいて、IIS と Relay Server に適切なセキュリティを設定する必要があります。

6. セキュリティの理由により、Relay Server 管理拡張機能への HTTPS アクセスを確保するには、*applicationHost.config* ファイルを編集して次のコードを <configuration> セクションに追加します。

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

7. これらの変更を *applicationHost.config* ファイルに保存します。
8. デフォルト Web サイトの接続タイムアウトプロパティを最小の 60 秒に設定します。この値のデフォルト値は 120 秒であり、これで十分です。
9. 次のガイドラインに従って、Relay Server 設定ファイル *rs.config* を作成します。
 - ファイルには、次の 4 つのセクションが必要です。
 - オプションセクション
 - 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ーセクション
 - 各セクションはセクションタグで開始します。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。
 - 各セクションに適切なプロパティを追加します。プロパティは、等号の左側にプロパティ名、等号の右側にその値を指定することによって定義されます。たとえば、*property name = value*。
 - 設定ファイルには、7 ビット ASCII 文字以外を含めることはできません。
10. *rs.config* ファイルを *%SQLANY12%\¥RelayServer¥IIS¥Bin¥Server* ディレクトリにコピーします。
11. パフォーマンスのヒントを参考にして、最適なパフォーマンスを得ることができるようにします。
12. 次のようなコマンドラインを使用して、Relay Server ステイトマネージャーをサービスとして起動します。

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

注意

ステイトマネージャーをサービスとして起動することをおすすめします。ただし、Relay Server によって自動的に起動することもできます。

13. Windows 上で Microsoft IIS 用の Relay Server 設定を更新します。
 - a. 更新中の Relay Server ファームに属するコンピューターごとに、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 *%SQLANY12%\¥RelayServer¥IIS*

`¥BinXX¥Server` ディレクトリにコピーします。自動起動を使用する場合は、設定ファイルを `rs.config` という名前にしてください。

- b. `%SQLANY12%¥RelayServer¥IIS¥BinXX¥Server` 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. 更新中の Relay Server ファーム内のコンピューターごとに、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ーに配備されます。

次の手順

なし

参照

- 「Relay Server 設定ファイル」 25 ページ
- 「パフォーマンスに関するヒント」 10 ページ
- 「Relay Server ステイトマネージャー」 21 ページ
- 「サービスとしての Relay Server ステイトマネージャー」 21 ページ
- 「Relay Server ステイトマネージャーのコマンドラインの構文」 23 ページ
- 「ファイル非表示ユーティリティ (dbfhide)」 39 ページ

Linux 上の Apache への Relay Server コンポーネントの配備

Relay Server ファームの各コンピューターに Relay Server ファイルを配備してから Apache で Relay Server を実行する必要があります。

前提条件

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。Linux では、Relay Server ファイルは、SQL Anywhere インストール環境の一部として `/opt/sqlanywhere12` ディレクトリにインストールされます。

内容と備考

対話型クイック設定機能

対話型クイック設定機能は、この手順の代替手段として提供されます。クイック設定機能：

- Relay Server 用の Web サーバーを設定します
- デモアプリケーションを作成します
- クイックリファレンスガイドを生成します

クイック設定機能の詳細な説明については `readme` ファイルを参照してください。クイック設定は 2 つの主要手順から構成されます。

1. Relay Server 用の Apache Web サーバーを設定します。この手順は、`install-dir/RelayServer/Apache/quicksetup_apache` ディレクトリの `ap-setup.sh` スクリプトを実行することにより実施できます。
2. Relay Server テストサービスを作成します。この手順は、`install-dir/RelayServer/Apache/quicksetup_apache` ディレクトリの `rs-test-setup.sh` スクリプトを実行することにより実施できます。

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`
- `dblgen12.res`
- `libdbtasks12.so`
- `libdbtasks12_r.so`
- `libdbicudt12.so`
- `libdbicu12_r.so`
- `libdblib12_r.so`
- `dbsupport`
- `dbfhide`
- `libdblib12.so`
- `mod_rs_ap_monitor.so`
- `mod_rs_ap_admin.so`

◆ Linux 上の Apache への Relay Server コンポーネントの配備

1. Relay Server 設定ファイル `rs.config` を作成します。
2. `rs.config` を `install-dir/relayservers/apache/bin??` ディレクトリにコピーします。サーバーモジュールは、`rshost` 実行プログラムが `rs.config` ファイルをコピーしたディレクトリと同じディレクトリにあることを想定します。
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、次の 4 つのセクションが必要です。
 - 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ーセクション
 - オプションセクション
 - 各セクションはセクションタグで開始します。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。
 - 各セクションに適切なプロパティを追加します。プロパティは、等号の左側にプロパティ名、等号の右側にその値を指定することによって定義されます。たとえば、`property name = value`。
 - 設定ファイルには、7 ビット ASCII 文字以外を含めることはできません。
4. LD_LIBRARY_PATH 環境変数には、Apache `install-dir/lib??` と `install-dir/relaysolver/apache/bin??` ディレクトリが含まれている必要があります。設定する `<apache-dir>/bin/envvars` ファイルを編集してから LD_LIBRARY_PATH をエクスポートします。
 5. Apache の `conf/httpd.conf` ファイルを編集します。
 - a. Relay Server のクライアントモジュールとサーバー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_client_module install-dir/relaysolver/apache/bin??/  
mod_rs_ap_client.so
```

```
LoadModule iarelayserver_server_module install-dir/relaysolver/apache/bin??/  
mod_rs_ap_server.so
```

注意

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。

- b. SQL Anywhere モニター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_monitor_module install-dir/relaysolver/apache/bin??/  
mod_rs_ap_monitor.so
```
 - 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/relaysolver/apache/bin??/  
mod_rs_ap_admin.so
```
 - d. クライアントモジュールの `<locationMatch>` セクションを作成するための、次の行を追加します。

```
<LocationMatch /cli/iarelayserver/* >  
  SetHandler iarelayserver-client-handler  
</LocationMatch>
```
 - e. サーバーモジュールの `<location>` セクションを作成するための、次の行を追加します。

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

```
RSConfigFile "/install-dir/relaysrver/apache/bin??.rs.config"
</Location>
```

注意

Relay Server 設定ファイル *rs.config* の場所を指定する **RSConfigFile** ディレクティブを指定してください。*rs.config* ファイルは、**rshost** 実行プログラムが配備されているディレクトリと同じディレクトリに存在している必要があります。

- f. SQL Anywhere モニターモジュールの `<location>` セクションを作成するための、次の行を追加します。

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

- g. Remote 管理モジュールの `<location>` セクションを作成するための、次の行を追加します。

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

- h. `TimeOut` ディレクティブが設定されている場合は、60 秒以上に設定されていることを確認します。

6. Linux の場合、Apache がプロセスを生成するときに `$TMP`、`$TMPDIR`、`$TEMP` のいずれかの環境変数がグローバルに設定されている場合、Apache をこれ以上設定する必要はありません。

これらの環境変数のいずれもグローバルに設定されていない場合、または既定の Relay Server ログファイルを特定のテンポラリディレクトリに保存する場合は (たとえばカスタマイズしないでステイトマネージャーを自動的に起動する場合)、ファイル `<apache-dir>/bin/envvars` を編集して `TMP` を設定し、エクスポートします。

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

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

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。

注意

Apache ユーザープロセスには、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。

7. Relay Server の起動中に Relay Server 設定を更新する場合は、次の手順に従います。
- 更新された設定ファイルを `install-dir/relaysrver/apache/bin??` ディレクトリにコピーします。自動起動を使用する場合は、設定ファイルを `rs.config` という名前にしてください。
 - `install-dir/relaysrver/apache/bin??` ディレクトリから、次のコマンドラインを実行して設定の更新を適用します。

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


- c. 複数のサーバー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ー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ーに配備されます。

次の手順

なし

Linux でサポートされている Apache のバージョンについては、http://www.ianywhere.jp/tech/1061806-os_components.html を参照してください。

参照

- 「Relay Server 設定ファイル」 25 ページ
- 「パフォーマンスに関するヒント」 10 ページ
- 「Relay Server ステイトマネージャー」 21 ページ
- 「サービスとしての Relay Server ステイトマネージャー」 21 ページ
- 「Relay Server ステイトマネージャーのコマンドラインの構文」 23 ページ
- 「ファイル非表示ユーティリティ (dbfhide)」 39 ページ

同時接続

Relay Server のバージョン

これは、Apache Web サーバー上の Relay Server 12.0.x のみに適用されます。これは、Apache Web サーバー上の Relay Server 11.0.x には適用されません。Relay Server 11.0.x については、次を参照してください。

Apache Web サーバーは、最大クライアントディレクティブを使用した同時接続 (同時要求) を制御します。デフォルトでは、最大クライアントディレクティブは、256 に設定されます。Apache Web サーバーで 256 を超えるの同時接続が確立された場合は、通常受信バックログディレクティブに基づき 256 の制限を越える接続がキューイングされます。デフォルトでは、受信バックログディレクティブは 511 に設定されています。

256 を超える同時接続を処理する Apache Web サーバーでは、最大クライアントディレクティブが `httpd.conf` ファイルに設定される必要があります。最大クライアントディレクティブが増えた場合、サーバー制限ディレクティブも変更して、Web サーバーの Apache プロセスの数を増やす必要があります。

Relay Server 12.0.x は、Relay Server が使用するセマフォを管理するためにセマフォマネージャーを組み込んでいます。その結果、最大クライアントとサーバー制限ディレクティブを変更する場合にシステムの「セマフォセット」を増やす必要がありません。

同時接続の数を増やすには、`httpd.conf` に次の行を追加します。

```
ServerLimit 1000
MaxClient 1000
```

ビジュー状態の Web サーバー内で調整できる他の Apache ディレクティブには次のものがあります。

MaxSpareServers
MinSpareServers
StartServers

Relay Server ステイトマネージャー

Relay Server ステイトマネージャーは、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ーは、Relay Server によって使用されるログファイルを管理する役割も持っています。ステイトマネージャーは、Relay Server によって自動的に起動することも、サービスとして起動することもできます。

デフォルトのログファイル名は、`ias_relay_server_host.log` です。Windows では、このファイルは、TEMP 環境変数によって指定されたディレクトリにあります。Linux では、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。これらの環境変数がいずれも設定されていない場合は、`/tmp` ディレクトリにログファイルが作成されません。

注意

Apache ユーザープロセスには、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ーは、ログファイルを `<yymmdd><nn>.log` 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です。

ステイトマネージャーをサービスとして起動することをおすすめします。ステイトマネージャーをコマンドラインから手動で起動することはサポートされていません。

ステイトマネージャーを起動するために、Relay Server によって使用されるオプションを指定することができます。オプションを変更するには、Relay Server 設定ファイルのオプションセクションで「start」プロパティを設定します。次に例を示します。

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

オプションの前に、Relay Server ステイトマネージャーの実行プログラムの名前 (`rshost`) を指定してください。

サービスとしての Relay Server ステイトマネージャー

サービスユーティリティ (`dbsvc`) を使用して、ステイトマネージャーをサービスとして起動できます。これには、Relay Server 設定ファイルのオプションセクションで `start` プロパティを「no」に設定する必要があります。

サービスユーティリティは、サービスを作成、変更、削除するときに使用されます。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。

Windows で自動起動するステイトマネージャーサービス RelayServer を設定する場合

```
dbsvc -as -s auto -t rshost -w RelayServer "%SQLANY12%\¥RelayServer¥IIS¥BinXX¥Server¥rshost.exe" -q -qc -f "%SQLANY12%\¥RelayServer¥IIS¥BinXX¥Server¥rs.config" -o "c:\¥temp¥ias_relay_server.log"
```

UNIX で自動起動するステイトマネージャーサービス RelayServer を設定する場合

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

備考

dbsvc の構文は、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 スイッチ引数の後にある最初のパラメーターとして指定しません。

フルパスのみを使用します。

UNIX では、Apache ユーザーのプロセスをステイトマネージャー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実行できるようにするために、(可能な場合は同じ) ユーザーアカウントを使用します。

サービスを開始する場合

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

サービスを停止する場合

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

サービスをアンインストールする場合

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

参照

- [「オプションセクション」 30 ページ](#)

Relay Server ステイトマネージャーの自動起動

最初の Outbound Enabler が Relay Server に接続すると、ステイトマネージャープロセスが自動的に開始されます。Relay Server 設定ファイルのオプションセクションで、start プロパティが指定されていないか、このプロパティが明示的に auto と指定されている場合、これがデフォルトの動作です。デフォルトのログファイルのロケーションは、%temp%\¥ias_relay_server_host.log です。

カスタマイズオプションを指定した Relay Server ステイトマネージャーの自動起動

自動起動が必要で、冗長性レベルやログファイルロケーションなど一部のデフォルト動作を上書きしたい場合は、Relay Server 設定ファイルのオプションセクションの start プロパティで、ステ

イトマネージャーのコマンドラインを明示的に指定することができます。この場合、`-f` オプションは使用できません。設定ファイルの名前は `rs.config` にして、サーバー拡張機能と同じディレクトリに配置する必要があります。

注意

IIS を使用している場合は、`wwwroot` ディレクトリの下にログファイルロケーションを指定しないでください。Microsoft IIS では、ワーカプロセスがパブリッシュされたツリーの下にファイルを作成することを許可していません。

参照

- 「オプションセクション」 30 ページ
- 「Relay Server ステイトマネージャーのコマンドラインの構文」 23 ページ

Relay Server ステイトマネージャーのコマンドラインの構文

`rshost [option]+`

パラメーター

オプション ステイトマネージャーを設定するために使用でき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
<code>-f filename</code>	Relay Server 設定ファイルのファイル名を示します。
<code>-o filename</code>	ログを取るために使用するファイルの名前を示します。
<code>-os size</code>	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。 <code>-os</code> が指定されている場合、 <code><yymmdd><nn>.olg</code> 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、コンピューター名、プロセッサのアーキテクチャー、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。
<code>-oq</code>	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。
<code>-q</code>	最小化ウィンドウで実行します。
<code>-qc</code>	完了時にウィンドウを閉じます。
<code>-u</code>	実行している Relay Server の設定を更新します。

rshost のオプション	説明
-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。

Relay Server 設定ファイル

Relay Server 設定ファイルは、Relay Server ファームと、Relay Server ファームに接続するバックエンドサーバーファームの両方を定義するために使用します。Relay Server 設定ファイルは、次のセクションに分かれています。

- 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ーセクション
- オプションセクション

各セクションは、セクションタグで始まります。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。たとえば [relay_server] は、Relay Server セクションの始まりを示します。

セクションタグの後には、定義されているセクションに関連するさまざまなプロパティを定義する多くの行が続きます。プロパティは、等号の左側にプロパティ名、等号の右側にその値を指定することによって定義されます。たとえば、**property name = value**。どのセクション名およびプロパティ名も、大文字と小文字が区別されません。コメントは、行の先頭のシャープ記号 (#) 文字によって示されます。

設定ファイルには、7 ビット ASCII 文字以外を含めることはできません。セクションは任意の順序で指定できます。

Relay Server 設定ファイルは、Sybase Central の Relay Server プラグインを使用して作成、インポート、配備できます。

参照

- [「Sybase Central 用の Relay Server プラグイン」 47 ページ](#)
- [「Relay Server ファーム設定の更新」 43 ページ](#)

Relay Server セクション

Relay Server セクションは、1 つの Relay Server を定義するために使用されるため、ファーム内の Relay Server ごとに Relay Server セクションが必要です。このセクションは、relay_server キーワードによって識別されます。

Relay Server セクションのプロパティ

Relay Server セクションでは、次のプロパティを指定できます。

- **enable** Relay Server ファームにこの Relay Server を含めるかどうかを指定します。可能な値は、次のとおりです。
 - **Yes** 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。

デフォルトは Yes です。このプロパティは省略可能です。

- **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。
- **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。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。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** カスタム説明を最大 2048 文字で入力します。このプロパティは省略可能です。

バックエンドファームセクション

バックエンドファームセクションは、バックエンドサーバーファームのプロパティを指定します。バックエンドサーバーファームは、同種のバックエンドサーバーで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ーファームを指定する必要があります。バックエンドサーバーファームごとに、1つのバックエンドファームセクションがあります。

このセクションは、`backend_farm` キーワードによって識別されます。

バックエンドファームセクションのプロパティ

バックエンドファームセクションでは、次のプロパティを指定できます。

- **active_cookie** クライアント/サーバーの結び付きを維持するために `cookie` を設定するかどうかを指定します。
 - **yes** これがデフォルト設定です。クライアント/サーバーセッションの結び付きを維持するために、Relay Server では標準の HTTP `set-cookie` コマンドの応答に独自の `cookie` 名が注入されます。
 - **no** アクティブな `cookie` は設定されません。バックエンドファームでセッションレスのブラウザアプリケーションを実行している場合は、このオプションを使用します。たとえば、バックエンドファームでセッションレスの SQL Anywhere Web サービスを提供している場合などがこれに該当します。

最適な結果を得るには、このコントロールを次のように設定します。

バックエンドサーバータイプ	active_cookie の設定	active_header の設定
Mobile Link	no	yes
SQL Anywhere	no	no

Mobile Link サーバーバックエンドの場合、active_cookie と active_header の両方を yes にします。ただし、両方を yes に設定すると、セッション中のすべての HTTP 要求/応答でセッション情報が冗長する可能性があります。送信中の累積バイト数を節約するため、いずれかの active_cookie を no に設定できます。選択した設定がすべての場合に動作することを確認するため、すべてのネットワークシナリオをテストする必要があります。

- **active_header** クライアント/サーバーの結び付きを維持するためにヘッダーを設定するかどうかを指定します。
 - **yes** これがデフォルト設定です。クライアント/サーバーセッションの結び付きを維持するために、Relay Server では、仲介者によって active_cookie が改ざんされた場合に備えて、応答に独自のヘッダーが注入されます。
 - **no** 独自のヘッダーは使用されません。バックエンドファームでブラウザアプリケーションのみを実行している場合や、バックエンドファームのすべてのクライアントで active_cookie が正しく機能している場合は、トラフィックボリューム削減のためにこのオプションを設定します。
- **backend_security** バックエンドサーバーファーム内の Outbound Enabler が Relay Server ファームに接続するのに必要なセキュリティレベルを指定します。値は次のいずれかです。
 - **on** バックエンドファームからのすべての接続を、HTTPS を使用して行う必要があることを示します。
 - **off** バックエンドファームからのすべての接続を、HTTP を使用して行う必要があることを示します。

このプロパティは省略可能です。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。
- **client_security** バックエンドサーバーファームがクライアントに要求するセキュリティレベルを指定します。値は次のいずれかです。
 - **on** クライアントが HTTPS を使用して接続する必要があることを示します。
 - **off** クライアントが HTTP を使用して接続する必要があることを示します。

このプロパティは省略可能です。値が指定されていない場合、クライアントは HTTP、HTTPS のどちらを使用しても接続できます。
- **description** カスタム説明を最大 2048 文字で入力します。このプロパティは省略可能です。

- **enable** このバックエンドサーバーファームからの接続を許可するかどうかを指定します。可能な値は、次のとおりです。
 - **Yes** このバックエンドサーバー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ーファームからの接続を禁止します。デフォルトは Yes です。このプロパティは省略可能です。
- **id** バックエンドサーバーファームに割り当てる最大 2048 文字の名前。
- **forward_x509_identity** SAP NetWeaver Gateway は、信頼できる仲介者を通じて転送する X.509 証明書など、クライアントを認証するいくつかの手段を提供します。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ーから転送されたクライアント ID 情報を抽出し、HTTP ヘッダーを使用してこれを 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** SAP 仲介者のチェーンが存在する場合、クライアント ID ヘッダーはすでに要求内に存在しているかもしれません。しかし、すべてのクライアントに、フォワーダーとして動作するパーミッションが付与されているとは限りません。したがって、デフォルトの動作は、既存のヘッダーをフォワーダーの ID と置き換えることです。フォワーダーとして他のクライアント ID を転送するパーミッションを付与するときには、**forwarder_certificate_issuer=<match-string>** と **forwarder_certificate_subject=<match-string>** を設定します。ここで、<match-string> は、証明書内の対応する複合名の直列化された形式に対してチェックを行います。任意の文字と一致させるには '?'、任意の文字列と一致させるには '*' が使用できます。'?','*','または'¥'を文字どおり一致させる必要がある場合には、それらの先頭エスケープ文字として '¥' を使用します。

例：

```
forwarder_certificate_issuer = 'CN = quicksigner, OU = security department, O = my org, L = my city, S = my state, C = my country'
```
- **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 ヘッダーはすでに要求内に存在しているかもしれません。しかし、すべてのクライアントに、フォワーダーとして動作するパーミッションが付与されているとは限りません。したがって、デフォルトの動作は、既存のヘッダーをフォワーダーの ID と置き換えることです。フォワーダーとして他のクライアント ID を転送するパーミッションを付与するときには、**forwarder_certificate_issuer=<match-string>** と **forwarder_certificate_subject=<match-string>** を設定します。ここで、<match-string> は、証明書内の対応する複合名の直列化された形式に対してチェックを行います。任意の文字と一致させるには '?'、任意の文字列と一致させるには '*' が使用できます。'?','*','または'¥'を文字どおり一致させる必要がある場合には、それらの先頭エスケープ文字として '¥' を使用します。

例：

```
forwarder_certificate_subject = 'CN = mySapWD???.my.com, OU = Sybase, O = SAP, *'
```
- **verbosity** verbosity は次のレベルに設定できます。
 - **0** エラーのみをログに取ります。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** 要求レベルロギング。すべての HTTP 要求がログファイルに書き込まれます。
- **2** 要求レベルロギング。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。

バックエンドサーバーセクション

バックエンドサーバーセクションは、バックエンドサーバー接続を定義します。このセクションは、Outbound Enabler がバックエンドサーバーに代わって Relay Server ファームに接続するときを使用する情報を指定します。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ーセクションがあります。バックエンドサーバーセクションは、バックエンドサーバーをバックエンドサーバーファームに割り当てることも行います。

次のバックエンドサーバーは、Relay Server とともに使用できます。

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

注意

どのバックエンドサーバー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「プラットフォーム別 SQL Anywhere コンポーネント」ページを参照してください。 http://www.iAnywhere.jp/tech/1061806-os_components.html を参照してください。

このセクションは、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ーセクションのプロパティ

バックエンドサーバーセクションでは、次のプロパティを指定できます。

- **description** カスタム説明を最大 2048 文字で入力します。このプロパティは省略可能です。
- **enable** このバックエンドサーバーからの接続を許可するかどうかを指定します。可能な値は、次のとおりです。

Yes このバックエンドサーバーからの接続を許可します。

No このバックエンドサーバーからの接続を禁止します。

デフォルトは **Yes** です。このプロパティは省略可能です。

● **farm** このバックエンドサーバーが属するバックエンドサーバーファームの名前。

● **id** バックエンドサーバー接続に割り当てる最大 2048 文字の名前。

● **MAC** 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ーの MAC アドレス。アドレスは、IEEE 802 MAC-48 フォーマットを使用して指定します。正しいフォーマットの MAC アドレスを取得するには、Mobile Link Relay Server Outbound Enabler のコンソールまたはログを確認します。このプロパティは省略可能です。指定しない場合、MAC アドレスのチェックは行われません。

● **token** Relay Server がバックエンドサーバー接続を認証するために使用する、最大 2048 文字のセキュリティトークン。このプロパティは省略可能です。

● **verbosity** verbosity は次のレベルに設定できます。

0 エラーのみをログに取ります。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 要求レベルロギング。すべての HTTP 要求がログファイルに書き込まれます。

2 要求レベルロギング。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。

オプションセクション

オプションセクションは、ファーム内の各 Relay Server に適用するプロパティを指定します。オプションセクションは 1 つのみ使用できます。

このセクションは、options キーワードによって識別されます。

オプションセクションのプロパティ

オプションセクションでは、次のプロパティを指定できます。

● **start** ステイトマネージャーを起動する方法。値は次のいずれかです。

auto ステイトマネージャーは、ステイトマネージャーのコマンドラインのデフォルトを使用して自動的に起動されます。

no ステイトマネージャーは、Windows サービスとして外部で起動されます。

- **full path** ステイトマネージャーの実行プログラム (*rshost*) へのフルパスを指定します。デフォルトは `auto` です。このプロパティは省略可能です。
- **shared_mem** 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。デフォルトは 10 メガバイトです。このプロパティは省略可能です。
- **verbosity** `verbosity` は次のレベルに設定できます。
 - **0** エラーのみをログに取ります。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** 要求レベルロギング。すべての HTTP 要求がログファイルに書き込まれます。
 - **2** 要求レベルロギング。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。

Relay Server 設定ファイルのフォーマット

Relay Server 設定ファイルの基本的なフォーマットは、次のとおりです。

```
#
# 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 back-end
# server farm
#
[backend_farm]
# List of properties for a back-end 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 back-end server connection
property = value
```

Outbound Enabler

Outbound Enabler は、バックエンドサーバーと同じコンピューター上で実行されます。その目的は、次のとおりです。

- 企業 LAN 内で稼働しているコンピューターから 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く。
- 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ーに転送し、バックエンドサーバーからの応答を Relay Server 経由でクライアントに転送する。

Outbound Enabler は、起動されると、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。これは、Relay Server の Web サーバー拡張機能コンポーネントにマッピングするサーバー URL を使用して行われます。サーバー URL は、Relay Server に直接マッピングすることも、負荷分散装置にマッピングすることもできます。サーバー URL が負荷分散装置にマッピングされている場合、負荷分散装置は、要求をファーム内で実行されている Relay Server の 1 つに転送します。Outbound Enabler からの要求を受信した Relay Server は、ファーム内のすべての Relay Server の接続情報を返します。次に、Outbound Enabler は、返された各 Relay Server へのアウトバウンド接続 (チャンネルと呼ばれます) を 2 つ作成します。1 つは上りチャンネルと呼ばれ、基本的に応答の長さが無限である HTTP 要求を使用して作成されます。応答は、Relay Server から Outbound Enabler へのクライアント要求から構成される連続的なストリームです。2 つ目のチャンネルは下りチャンネルと呼ばれ、基本的にコンテンツの長さが無限である HTTP 要求を使用して作成されます。要求は、クライアント要求に対するサーバー応答の連続的なストリームから構成されます。

Outbound Enabler は、接続している Relay Server のいずれかから上りチャンネルでクライアント要求を受信すると、それを Outbound Enabler が処理しているバックエンドサーバーに転送します。応答は、バックエンドサーバーから受信されると、下りチャンネルを使用して対応する要求を受信した Relay Server に転送されます。

注意

次のバックエンドサーバーは、Relay Server とともに使用できます。

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

どのバックエンドサーバー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「プラットフォーム別 SQL Anywhere コンポーネント」ページを参照してください。 http://www.iAnywhere.jp/tech/1061806-os_components.html を参照してください。

Outbound Enabler の構文

`rsoe [option]+`

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

パラメーター

オプション Outbound Enabler では次のオプションを使用できます。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)、およびサーバー (-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。

rsoe のオプション	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。設定ファイル内のパスワードなどの情報を保護する場合は、ファイル難読化ユーティリティを使用して、設定ファイルの内容を難読化できます。「 ファイル非表示ユーティリティ (dbfhide) 」 39 ページ を参照してください。

rsoe のオプション	説明
<p>-cr <i>"connection-string"</i></p>	<p>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、名前と値のペアがセミコロンで区切られたリストです。名前と値のペアは、次の要素で構成されます。</p> <ul style="list-style-type: none"> ○ 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。 「host」『Mobile Link クライアント管理』を参照してください。 ○ port Relay Server が受信しているポート。このオプションは必須です。 「port」『Mobile Link クライアント管理』を参照してください。 ○ http_userid 認証用のユーザー ID です。省略可能です。HTTP 認証の設定方法については、Web サーバー (またはプロキシ) のマニュアルを参照してください。 「http_userid」『Mobile Link クライアント管理』を参照してください。 ○ http_password 認証用のパスワードです。省略可能です。HTTP 認証の設定方法については、Web サーバー (またはプロキシ) のマニュアルを参照してください。 「http_password」『Mobile Link クライアント管理』を参照してください。 ○ http_proxy_userid プロキシ認証用のユーザー ID です。省略可能です。HTTP 認証の設定方法については、Web サーバー (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」『Mobile Link クライアント管理』を参照してください。 ○ http_proxy_password プロキシ認証用のパスワードです。省略可能です。HTTP 認証の設定方法については、Web サーバー (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」『Mobile Link クライアント管理』を参照してください。 ○ proxy_host プロキシサーバーのホスト名または IP アドレスを指定します。省略可能です。 「proxy_host」『Mobile Link クライアント管理』を参照してください。

rsoe のオプション	説明
	<p>○ proxy_port プロキシサーバーのポート番号を指定します。省略可能です。</p> <p>「proxy_port」『Mobile Link クライアント管理』を参照してください。</p> <p>○ url_suffix Relay Server のサーバー拡張機能への URL パス。必須。</p> <p>デフォルトでは、rsoe で url_suffix を指定する必要があります。</p> <p>「url_suffix」『Mobile Link クライアント管理』を参照してください。</p> <p>○ https 0 - HTTP (デフォルト)</p> <p>1 - HTTPS</p> <p>デフォルトでは、Mobile Link は TCPIP コミュニケーション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、必ずお客様の RSOE 設定で必要なコミュニケーション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。「-x mlsrv12 オプション」『Mobile Link サーバー管理』を参照してください。</p> <p>https=1 パラメーターが -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。</p> <p>https=1 の場合は、次のオプションも指定できます。</p> <p>○ tls_type RSA または ECC</p> <p>○ certificate_name 証明書の通称フィールド。</p> <p>○ certificate_company 証明書の組織名フィールド。</p> <p>○ certificate_unit 証明書の組織単位フィールド。</p> <p>○ identity Outbound Enabler とバックエンドサーバー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ーに必要です。</p> <p>○ identity_password Outbound Enabler とバックエンドサーバー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ーに必要です。</p>

rsoe のオプション	説明
	<p>○ fips Yes または No。</p> <p>○ trusted_certificates 信頼できるルート証明書のリストを含むファイル。</p> <p>バックエンドサーバー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。</p> <p>trusted_certificates=backend_server_public_cert_filename</p> <p>Windows については、trusted_certificate 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。</p> <p>詳細については、「Mobile Link クライアントネットワークプロトコルオプション」『Mobile Link クライアント管理』を参照してください。</p>
-cs "connection-string"	<p>バックエンドサーバーへの接続に使用するホストとポートを設定します。デフォルトは "host=localhost;port=80" です。</p> <p>定期的なバックエンドサーバステータス要求を有効にするには、status_url パラメーターを -cs に追加します。status_url パラメーターは、status_url=/your-status-url のフォーマットで指定します。</p> <p>次の例は、-cs で status_url を指定する方法を示します。</p> <p>-cs "host=localhost;port=80;status_url=/getstatus/"</p> <p>バックエンドサーバステータス要求の頻度を指定するには、-d オプションを使用します。</p>
-d seconds	<p>バックエンドサーバーの活性 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。</p>
-dl	<p>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。</p>
-f farm	<p>バックエンドサーバーが属するファームの名前を指定します。</p>
-id id	<p>バックエンドサーバーに割り当てられた名前を指定します。</p>
-o file	<p>出力メッセージのログの記録先ファイルを指定します。</p>
-oq	<p>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。</p>

rsoe のオプション	説明
-os	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	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のログを取ります。
-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	rsoe をシェルモードで起動します。これはデフォルトです。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。
-ud	rsoe をデーモンとして実行するように指示します。このオプションが適用されるのは、UNIX プラットフォームのみです。
-ui	使用可能な表示がない場合は、rsoe 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、サーバー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、使用可能な表示が見つからなかった場合は、rsoe はシェルモードで起動されます。
-ux	Linux の場合に、メッセージを表示する、rsoe のメッセージウィンドウを開きます。 -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。 Windows では、rsoe のメッセージウィンドウが自動的に表示されません。

rsoc のオプション	説明
-v level	<p>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 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、主に技術サポートに使用されます)。</p> <ul style="list-style-type: none"> ○ 0 エラーのみをログに取ります。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 要求レベルロギング。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。 <p>レベル 1 と 2 では、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 スイッチを使用します。</p>

ファイル非表示ユーティリティ (dbfhide)

ファイル難読化ユーティリティ (dbfhide) は、単純暗号化を使用して、設定ファイルと初期化ファイルの内容を難読化します。

構文

dbfhide original-configuration-file encrypted-configuration-file

オプション	説明
original-configuration-file	元のファイルの名前を指定します。
encrypted-configuration-file	難読化された新しいファイルの名前を指定します。

Relay Server と Outbound Enabler は、dbfhide を使用して設定ファイルが難読化されたことを検出して、処理します。

このユーティリティは、設定ファイルからオプションを読み込む @data パラメーターを受け入れません。

統合化 Outbound Enabler (MobiLink が対象)

mlsrv12 の -x オプションに oe プロトコルを使用すると、rsoc コマンドで呼び出されたスタンドアロン Outbound Enabler の代わりに統合化 Outbound Enabler を使用できます。統合化 Outbound Enabler の使用には、次の利点があります。

- システムリソース、特にソケットの使用が削減されます。

- 単一の統合化ログファイルが提供されます。統合化 Outbound Enabler から Mobile Link サーバーログに出力される行にプレフィクス <OE> が付けられます。
- 配備が簡略化されます。
- Outbound Enabler と Mobile Link サーバー間の活性チェックが省略されます。

統合化 Outbound Enabler の使用法の詳細については、「[-x mlsrv12 オプション](#)」『[Mobile Link サーバー管理](#)』を参照してください。

配備に関する考慮事項

Outbound Enabler を使用しているときは、次の考慮事項に注意してください。

- **サービスとしての Outbound Enabler** サービスユーティリティを使用して、Outbound Enabler をサービスとして設定および管理することもできます。
- **認証** 単純認証やダイジェスト認証は使用できません。*rsoe.exe* は、Web サーバーの種類やオペレーティングシステムにかかわらず、Web サーバーでの単純認証やダイジェスト認証はサポートしていません。

参照

- [「サービスとしての Outbound Enabler」 40 ページ](#)

サービスとしての Outbound Enabler

サービスユーティリティ (*dbsvc*) を使用して、Outbound Enabler をサービスとして起動できます。サービスユーティリティは、サービスを作成、変更、削除するときに使用されます。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。

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 rsoe -w oes "%SQLANY12%\BinXX\rsoe.exe"  
-cr "host=relayserver.sybase.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 <some-user-account> -t rsoe -w oes @/<full-dir-path>/oe.config
```

備考

dbsvc の構文は、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* スイッチ引数の後にある最初のパラメーターとして指定しません。

フルパスのみを使用します。

UNIX では、Outbound Enabler パラメーターをコマンドラインでのみ指定します。コマンドラインスイッチを *setup dbsvc* コマンドで使用しないでください。

サービスを開始する場合

`dbsvc.exe -u oes`

サービスを停止する場合

`dbsvc.exe -x oes`

サービスをアンインストールする場合

`dbsvc.exe -d oes`

参照

- [「SQL Anywhere Web サービスの高可用性とスケールアウトソリューション」『SQL Anywhere サーバー データベース管理』](#)

Relay Server ファーム設定の更新

Relay Server ファーム設定は、Relay Server 設定ファイルの内容によって定義されます。Relay Server ファーム内の各 Relay Server は、同じ Relay Server 設定ファイルを共有します。そのため、Relay Server ファーム設定を更新するときは、ファーム内の Relay Server ごとに Relay Server 設定ファイルを更新する必要があります。更新には、次の操作が含まれます。

- 新しい Relay Server を Relay Server ファームに追加する。
- 新しいバックエンドサーバーファームを作成して、Relay Server ファームへのアクセスを許可する。
- 新しいバックエンドサーバーを既存のバックエンドサーバーファームに追加する。
- Relay Server、バックエンドサーバーファーム、またはバックエンドサーバーのプロパティを変更する。
- オプションを変更する。

Relay Server 設定を更新する 1 つの方法は、すべての Relay Server をシャットダウンし、Relay Server 設定ファイルを更新したバージョンで置き換えて、すべての Relay Server を再起動することです。ただし、Relay Server をシャットダウンして再起動するということは、Relay Server のユーザーがサービスの中断を被る可能性があります。

Relay Server 設定を更新する際の推奨方法は、Relay Server ファームが稼働している状態でサービスを中断せずに、Relay Server ステイトマネージャーを使用して設定を更新することです。

Relay Server 設定を更新するには、次のコマンドラインフォーマットを使用して、Relay Server ステイトマネージャーの新しいインスタンスを起動します。

```
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ーに指示します。
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。

次に、Relay Server ファーム設定を更新するために必要な手順の概要を示します。

1. Relay Server 設定ファイルのマスターコピーに変更を加えます。
2. 更新中の Relay Server ファームに属し、Relay Server のインスタンスを実行しているコンピューターごとに、次の操作を行います。
 - a. 古い設定ファイルを更新した設定ファイルで置き換えます。
 - b. 更新された設定ファイルを使用して Relay Server ステイトマネージャーを実行します。

参照

- [「Relay Server ステイトマネージャー」 21 ページ](#)

Windows 上の Microsoft IIS 用の Relay Server 設定の更新

時々には Relay Server 設定ファイルを更新して、Relay Server または Relay Server ファームを追加または変更したり、サーバーとファームのプロパティとオプションを変更したりする必要があるかもしれません。

前提条件

既存の Relay Server ファーム用 Relay Server 設定ファイル

内容と備考

次のとおりです。

◆ Windows 上の Microsoft IIS 用の Relay Server 設定を更新する。

1. 更新中の Relay Server ファームに属するコンピューターごとに、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 `%SQLANY12%\¥RelayServer¥IIS¥BinXX¥Server` ディレクトリにコピーします。自動起動を使用する場合は、設定ファイルを `rs.config` という名前にしてください。
2. `%SQLANY12%\¥RelayServer¥IIS¥BinXX¥Server` 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
```

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

結果

Relay Server 設定が更新されました。

次の手順

なし

Linux 上の Apache 用の Relay Server 設定の更新

時々には Relay Server 設定ファイルを更新して、Relay Server または Relay Server ファームを追加または変更したり、サーバーとファームのプロパティとオプションを変更したりする必要があるかもしれません。

前提条件

既存の Relay Server ファーム用 Relay Server 設定ファイル

内容と備考

次のとおりです。

◆ Linux 上の Apache 用の Relay Server 設定を更新する。

1. 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。自動起動を使用する場合は、設定ファイルを `rs.config` という名前にしてください。
2. `/Apache-install/modules` ディレクトリから、次のコマンドラインを実行して設定の更新を適用します。

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

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

結果

Relay Server 設定が更新されました。

次の手順

なし

Sybase Central 用の Relay Server プラグイン

Sybase Central 用の Relay Server プラグインには、Relay Server を使用する簡単な方法があります。Relay Server プラグインを使用すると、次のことを実行できます。

- Relay Server 設定ファイルを作成、インポート、配備する。
- Relay Server 設定ファイルのプロパティを表示する。
- Relay Server、Relay Server ファーム、バックエンドサーバー、バックエンドサーバーファームを追加する。
- Relay Server、Relay Server ファーム、バックエンドサーバー、バックエンドサーバーファームを表示および編集する。

Relay Server 設定ファイルの操作 (Sybase Central)

Sybase Central を使用して、Relay Server 設定ファイルを操作できます。Sybase Central から、次の操作を実行できます。

- Relay Server 設定ファイルを作成する。
- Relay Server 設定ファイルを開く。
- Relay Server 設定ファイルをインポートする。
- Relay Server 設定ファイルを配備する。

◆ Relay Server 設定ファイルを作成する。

1. Sybase Central の [フォルダー] ビューで、[Relay Server 12] を右クリックし、[新規] » [設定ファイル] をクリックします。
2. Sybase Central が稼働しているコンピューターで、設定ファイルを保存するディレクトリを参照します。これは、配備場所と同じではありません。
3. [ファイル名] フィールドに、設定ファイルの名前を入力します。通常は、*rs.config* です。
4. [保存タイプ] フィールドで、*.config* 拡張子が選択されていることを確認します。
5. [保存] をクリックします。必要な Relay Server とバックエンドサーバーを追加できる Relay Server ファームが自動的に作成されます。

◆ Relay Server 設定ファイルを開く。

1. Sybase Central の [フォルダー] ビューで、[Relay Server 12] を右クリックし、[設定ファイルを開く] をクリックします。
2. 設定ファイルがあるディレクトリを参照し、そのファイルをクリックし、[開く] をクリックします。

◆ **Relay Server 設定ファイルをインポートする。**

1. Sybase Central の [フォルダー] ビューで、[Relay Server 12] を右クリックし、[設定ファイルのオープン] を選択します。
2. 既存の Relay Server の URL を入力します。
3. Relay Server に認証が必要な場合は、[ユーザー名] と [パスワード] に値を入力し、[OK] をクリックします。

注意

HTTPS 通信を必要とする Relay Server の場合は、サーバーのルート証明書を Java キーと証明書管理ユーティリティに格納しておく必要があります。これには、Java の keytool ユーティリティを使用します。通信にルート証明書が必要な場合、Sybase Central は Java キーと証明書管理ユーティリティにアクセスします。

◆ **Relay Server 設定ファイルを配備する。**

1. [フォルダー] ビューで、配備する Relay Server 設定ファイルを右クリックし、[展開] をクリックします。
2. Relay Server の URL を入力します。
3. Relay Server に認証が必要な場合は、[ユーザー名] と [パスワード] に値を入力し、[OK] をクリックします。

注意

HTTPS 通信を必要とする Relay Server の場合は、サーバーのルート証明書を Java キーと証明書管理ユーティリティに格納しておく必要があります。これには、Java の keytool ユーティリティを使用します。通信にルート証明書が必要な場合、Sybase Central は Java キーと証明書管理ユーティリティにアクセスします。

4. [サーバーリスト] ページに、既存の Relay Server が表示されます。1 つまたは複数の Relay Server に設定ファイルを配備するには、リストからサーバーを選択して、[追加] をクリックします。

リストから Relay Server を削除するには、サーバーを選択して、[削除] をクリックします。

Relay Server と Relay Server ファームの管理 (Sybase Central)

Sybase Central を使用して、Relay Server と Relay Server ファームを管理できます。Sybase Central から、次の操作を実行できます。

- Relay Server を Relay Server ファームに追加する。
- Relay Server のプロパティを表示または編集する。
- Relay Server ファームのプロパティを表示または編集する。

◆ Relay Server をファームに追加する。

1. 操作する設定ファイルの下にある **[フォルダー]** ウィンドウ枠で、Relay Server を追加する Relay Server ファームを右クリックし、**[新規]** » **[Relay Server]** をクリックします。
2. **[この Relay Server を有効にする]** が選択されていることを確認します。
3. 接続先となる **[ホスト]** のパス情報を入力します。指定したホストに接続できるかどうかをチェックする場合は、**[ping]** をクリックします。
4. 使用する通信プロトコルを選択します。**[HTTP]** または **[HTTPS]** のいずれかです。
5. 選択したプロトコルに使用するポートを指定します。
6. 必要に応じて、Relay Server の説明を **[説明]** フィールドに入力します。
7. Relay Server を引き続き追加する場合は **[適用]** をクリックし、Relay Server を追加して **[Relay Server の作成]** ウィンドウを閉じる場合は **[OK]** をクリックします。

◆ Relay Server のプロパティを表示または編集する。

1. **[フォルダー]** ウィンドウ枠で、操作する Relay Server を含む Relay Server ファームをクリックします。そのファームに含まれる Relay Server が、右ウィンドウ枠にリストされます。
2. 編集または表示する Relay Server を右クリックし、**[プロパティ]** をクリックします。
3. 必要に応じて Relay Server のプロパティを変更し、**[適用]** または **[OK]** をクリックします。

◆ Relay Server ファームのプロパティを表示または編集する。

1. **[フォルダー]** ウィンドウ枠で、操作する Relay Server ファームを右クリックし、**[プロパティ]** をクリックします。
2. 必要に応じて Relay Server ファームのプロパティを変更し、**[適用]** または **[OK]** をクリックします。

バックエンドサーバーとバックエンドサーバーファームの管理

Sybase Central を使用して、バックエンドサーバーとバックエンドサーバーファームを管理できます。Sybase Central から、次の操作を実行できます。

- バックエンドサーバーファームを作成し、バックエンドサーバーをそれに追加する。
- バックエンドサーバーのプロパティを表示または編集する。
- バックエンドサーバーファームのプロパティを表示または編集する。

◆ バックエンドサーバーファームを作成する。

1. 左ウィンドウ枠で、操作する Relay Server 設定ファイルを右クリックし、**[新規]** » **[バックエンドサーバーファーム]** をクリックします。
2. **[このバックエンドサーバーファームを有効にする]** が選択されていることを確認します。

3. 新しいバックエンドサーバーファームに関連付けられている名前を入力します。
4. **[クライアントセキュリティ]** で、バックエンドサーバーファームへの接続に使用するクライアントのプロトコルを選択します。
5. **[バックエンドセキュリティ]** で、バックエンドサーバーファームへの接続に使用する Relay Server Outbound Enabler (rsoe) のプロトコルを選択します。
6. クライアント/サーバーの結び付きについて、使用するサーバータイプを選択します。スタンドアロン Outbound Enabler を使用した Mobile Link HTTP サーバー、または埋め込み Outbound Enabler を使用した Mobile Link サーバーの場合、**[Mobile Link]** をクリックします。一般的な SQL Anywhere Web サービスの場合、**[SQL Anywhere]** をクリックします。サーバータイプに **[カスタム]** をクリックすると、高度なカスタム設定を使用できます。**[カスタム]** サーバータイプを選択すると、次の結び付き設定を完全に制御できます。
 - a. Relay Server で標準の HTTP set-cookie コマンドを使用してクライアント/サーバーの結び付きを維持する場合は、**[アクティブな cookie]** オプションをオンにします。
 - b. Relay Server で独自のヘッダーを使用してクライアント/サーバーの結び付きを維持する場合は、**[アクティブなヘッダー]** オプションをオンにします。
7. 必要に応じて、バックエンドサーバーファームの説明を **[説明]** フィールドに入力します。
8. バックエンドサーバーファームを引き続き追加する場合は **[適用]** をクリックし、バックエンドサーバーファームを追加して **[バックエンドサーバーファームの作成]** ウィンドウを閉じる場合は **[OK]** をクリックします。

◆ **バックエンドサーバーファームのプロパティを表示または編集する。**

1. **[フォルダー]** ウィンドウ枠で、操作するバックエンドサーバーファームを右クリックし、**[プロパティ]** をクリックします。
2. 必要に応じてバックエンドサーバーファームのプロパティを変更し、**[適用]** または **[OK]** をクリックします。

◆ **サーバーをバックエンドサーバーファームへ追加する。**

1. **[フォルダー]** ウィンドウ枠で、操作するバックエンドサーバーファームを右クリックし、**[新規]** » **[バックエンドサーバー]** をクリックします。
2. **[このバックエンドサーバーを有効にする]** が選択されていることを確認します。
3. 新しいバックエンドサーバーファームに関連付けられている名前を入力します。
4. MAC アドレスチェックを強制的に行うには、**[MAC アドレスの強制チェック]** チェックボックスをクリックします。
5. MAC アドレスチェックを選択した場合は、IEEE 802 MAC-48 フォーマットを使用して RSOE の MAC アドレスを入力します。正しいフォーマットの MAC アドレスを取得するには、Mobile Link Relay Server Outbound Enabler のコンソールまたはログを確認します。バックエンドサーバーマシン上で複数のアダプターが現在アクティブである場合は、感嘆符 (!) で区切られた複数の MAC アドレスが、Outbound Enabler によってレポートされます。Relay Server

でチェックする対象として、最も永続的な MAC アドレスを選択します。Windows 上で `ipconfig /all` コマンドを実行すると、ネットワークアダプターとそれに関連付けられた MAC アドレスの詳細なリストが表示されます。

6. Relay Server で使用されるセキュリティトークンを指定して、バックエンドサーバー接続を認証します。最大 2048 文字まで使用できます。
7. 必要に応じて、バックエンドサーバーの説明を **[説明]** フィールドに入力します。
8. バックエンドサーバーを引き続き追加する場合は **[適用]** をクリックし、バックエンドサーバーを追加して **[バックエンドサーバーの作成]** ウィンドウを閉じる場合は **[OK]** をクリックします。

◆ **バックエンドサーバーのプロパティを表示または編集する。**

1. **[フォルダー]** ウィンドウ枠で、操作するバックエンドサーバーを含むバックエンドサーバーファームをクリックします。そのファームに含まれるバックエンドサーバーが、右ウィンドウ枠にリストされます。
2. 編集または表示するバックエンドサーバーを右クリックし、**[プロパティ]** をクリックします。
3. 必要に応じてバックエンドサーバーのプロパティを変更し、**[適用]** または **[OK]** をクリックします。

Sybase ホストのリレーサービス

Sybase ホストのリレーサービスは、Sybase をホストとする Relay Server のファームです。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を容易にすること、また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することを目的としています。特に、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 側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。

Sybase 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に Relay Server をインストールしておく必要があります。

Sybase ホストのリレーサービスの使用

◆ Sybase ホストのリレーサービスのサブスクライブ

Sybase ホストのリレーサービスを使用するには、先にサブスクライブする必要があります。

1. 次の URL に移動してください。 <http://relayserver.sybase.com/account> これにより、Sybase ホストのリレーサービスのホームページが開きます。
2. **[Register]** をクリックしてアカウントを作成します。
3. **[Subscription ID]** と **[Password]** を指定し、自分および組織の連絡先情報を入力し、**[Hosted Relay Service Terms of Service]** に同意するように求められます。**[Subscription ID]** には組織でユニークな ID を選択してください。**[Submit]** をクリックします。

登録に成功したら、登録確認の電子メールが送られてきます。

◆ Sybase ホストのリレーサービスへのログイン

1. **[Log In]** をクリックして新しく作成したアカウントにログインします。
2. 登録処理中に入力した **[Subscription ID]** と **[Password]** を入力します。ログインすると、**[Account Information]** ページが開きます。アカウント情報のページでは、サブスクライバー情報を変更したり、このサービスにアクセスするバックエンドサーバーファームを指定したりすることができます。

◆ サーバーファームの追加

1. 追加したいファームのタイプをクリックします。次のものから選択します。
 - 新しい Mobile Link ファームの追加
 - 新しい SQL Anywhere Web Read-Write ファームの追加
 - 新しい SQL Anywhere Web Read ファームの追加
 - 新しい SQL Anywhere Web Read-Only ファームの追加
 - 新しい SQL Remote Message Server ファームの追加
 - 新しい Afaria ファームの追加
 - 新しい Anywhere Mobile Office ファームの追加
 - 新しい Sybase Unwired Platform ファームの追加
2. サーバーファームを表すユニークな **[Farm Name]** を入力します。
3. ファーム内の各サーバーに対してユニークな名前を指定します。最大で2つのサーバーを指定できます。
4. **[Create Farm]** をクリックします。ファームが正常に追加された場合は確認が表示されます。
5. **[Configuration Instructions]** をクリックして、サービスの使用方法の詳細を確認します。表示される指示内容は、入力した情報によって異なります。
6. 完了したら、**[Log Out]** をクリックします。

Mobile Link で使用する Relay Server

以降の項では、Relay Server と Mobile Link を一緒に使用する場合について説明します。

参照

- Relay Server でサポートされるオペレーティングシステムとブラウザについては、<http://www.ianywhere.jp/sas/os.html> を参照してください。
- Outbound Enabler の配備の詳細については、「[Mobile Link サーバーの配備](#)」『[Mobile Link サーバー管理](#)』を参照してください。

クライアントから Relay Server ファームへの接続

Relay Server ファームが適切に設定されたら、クライアントは次の URL を使用して Relay Server ファームに接続します。

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

オプション

オプション	説明
<code><Relay Server client extension URL></code>	<p>Windows 上の Microsoft IIS の場合、<code><domain name><relayserver.sybase.com>/ias_relay_server/client/rs_client.dll</code> です。</p> <p>Linux 上の Apache の場合、<code><domain name>/cli/iarelayserver</code> です。</p> <p>誰でも使用できる Sybase ホストのリレーサービスを使用している場合は、<code><domain name></code> に <code>relayserver.sybase.com</code> を使用します。サービスに対するサブスクリプトおよびバックエンドサーバーの設定手順の詳細については、「Sybase ホストのリレーサービスの使用」53 ページを参照してください。</p>
<code><farmname></code>	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 (バックエンドサーバーのグループ) を識別します。

SQL Anywhere Mobile Link クライアントの接続の例

サーバーファーム F1 に接続するには、SQL Anywhere Mobile Link クライアントで次のオプションを指定する必要があります。

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

HTTPS の場合は、http を https に変更します。

Ultra Light/Ultra Light J Mobile Link クライアントの接続の例

サーバーファーム **F1** に接続するには、Ultra Light/Ultra Light J Mobile Link クライアントで ULSyncParms クラスの次のプロパティを設定する必要があります。

- ストリームタイプを HTTP または HTTPS に設定します。
- ストリームパラメーターを次のように設定します。

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

QAnywhere クライアントの接続の例

サーバーファーム **F1** に接続するには、QAnywhere クライアントで次のオプションを指定する必要があります。

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

サンプルシナリオ

適切な設定で設定ファイルを設定し配備してから、Mobile Link をファームに接続します。

前提条件

このシナリオのため、Relay Server の Microsoft IIS バージョンを使用します。

内容と備考

ABC 社がモバイルアプリケーションの開発を完了し、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を設定したいと仮定します。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。したがって、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、顧客は次のような設定が必要であると判断しました。

- 2 つの Mobile Link サーバー
- 2 つの Relay Server
- 1 つの負荷分散装置
- 各 Relay Server は、専用のコンピューター上に配備されます。ホスト名が **rs1.abc.com** と **rs2.abc.com** の 2 つのコンピューターを使用します。
- 各 Mobile Link サーバーは、専用のコンピューター上に配備されます。2 つの Mobile Link サーバーは、**m11** と **m12** 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ーファームに属しています。
- 負荷分散装置は、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 サーバーには既知の認証局 (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ーコンピューターには対応する信頼できるルート証明書が標準の証明書ストアに保存されていると仮定します。

◆ Relay Server ファームの設定

1. 初めに、Relay Server 設定ファイルを作成します。

設定を含むファイルの名前は、*rs.config* にしてください。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 back-end 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. Relay Server を実行している 2 つのコンピューターに、Relay Server コンポーネントとともに設定ファイル *rs.config* を配備します。
3. Integrated Outbound Enabler を使用して Mobile Link サーバーを実行する 2 つのコンピューターで、Mobile Link サーバーを起動します。

ID が ml1 の Mobile Link サーバーを実行しているコンピューターで、次のコマンドを実行します。

```
mlsrv12 -x oe<config=oe1.txt> -zs ml1 <other ML options>
```

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

ID が ml2 の Mobile Link サーバーを実行しているコンピューターで、次のコマンドを実行します。

```
m1srv12 -x oe<config=oe2.txt> -zs ml2 <other ML options>
```

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

[「-x m1srv12 オプション」](#)『[Mobile Link サーバー管理](#)』を参照してください。

結果

すべてのサーバーと 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_client.dll/abc.mobilink

次の手順

なし

参照

- 「[エンドツーエンド暗号化](#)」『[SQL Anywhere サーバー データベース管理](#)』

索引

A

Apache

- Relay Server の配備, 15
- 同時接続、Relay Server 用に増やす, 19

H

HTTP 負荷分散装置

- Relay Server, 4

M

Microsoft IIS

- Relay Server、パフォーマンスに関するヒント, 10

Microsoft IIS 6.0

- Relay Server の配備, 7

Microsoft IIS 7.x

- Relay Server の配備, 11

Mobile Link

- Relay Server の使用, 55

O

Outbound Enabler

- Relay Server ファーム, 4
- 構文, 34
- サービスの開始, 40
- 説明, 33
- 配備に関する考慮事項, 40

R

Relay Server (参照 Relay Server)

- Mobile Link の使用, 55
- Mobile Link 用のサンプルシナリオ, 56
- Outbound Enabler, 33
- Outbound Enabler の構文, 34
- Outbound Enabler の配備, 40
- Relay Server ファーム, 4
- rshost の構文, 23
- rsoe の構文, 34
- Web サーバーを経由した同期, 1
- アーキテクチャー, 1
- ステイトマネージャー, 21
- ステイトマネージャーのコマンドラインの構文, 23

設定の更新, 43

設定ファイル, 25

設定ファイル、Sybase Central, 47

設定ファイルのフォーマット, 31

説明, 1

配備, 7

バックエンドサーバーファーム, 4

ホストサービス, 53

Relay Server ステイトマネージャー

Relay Server の配備、Apache, 15

Relay Server の配備、Microsoft IIS 6.0, 10

Relay Server の配備、Microsoft IIS 7.x, 15

rshost.exe, 23, 43

Windows サービスとして起動, 21

カスタマイズオプションを指定して自動的に起動, 22

コマンドラインの構文, 23

自動的に起動, 22

説明, 21

Relay Server セクション

Relay Server 設定ファイル, 25

Relay Server 設定の更新、Linux

rshost.exe, 44

Relay Server 設定の更新、Windows

rshost.exe, 44

Relay Server 設定ファイル

Relay Server セクション, 25

オプションセクション, 30

更新, 43

更新手順、Apache, 44

更新手順、Microsoft IIS, 44

説明, 25

バックエンドサーバーセクション, 29

バックエンドファームセクション, 26

ファーム設定, 43

フォーマット, 31

Relay Server の Outbound Enabler (参照 Outbound Enabler)

Relay Server の Web 拡張機能

説明, 1

配備、Microsoft IIS 6.0, 10

配備、Microsoft IIS 7.x, 15

Relay Server の設定

rshost.exe を使用した更新, 43

Relay Server の配備

Linux 上の Apache, 15

Linux 用のファイル, 15

rshost.exe, 7

Web サーバー拡張機能、Apache, 15
Web サーバー拡張機能、Microsoft IIS 6.0, 10
Web サーバー拡張機能、Microsoft IIS 7.x, 15
Windows IIS 6.0 用のファイル, 7
Windows 上の Microsoft IIS 6.0, 7
Windows 上の Microsoft IIS 7.x, 11
アプリケーションプール、Microsoft IIS 6.0, 10
同時接続、増やす, 19
Relay Server のパフォーマンスに関するヒント
Microsoft IIS, 10
Relay Server のホスティングサービス
(参照 Sybase ホストのリレーサービス)
Relay Server のホスト
サブスクリプト, 53
サーバーファームの追加, 54
ログイン, 53
Relay Server ファーム
HTTP 負荷分散装置, 4
Mobile Link, 4
Outbound Enabler, 4
クライアントの接続, 55
モバイルデバイスの接続, 55
Relay Server ファーム設定
更新, 43
rshost (参照 Relay Server ステイトマネージャー)
rshost.exe
Relay Server ステイトマネージャー, 43
Relay Server 設定の更新, 44
Relay Server 設定の更新、Linux, 44
Relay Server の配備, 7
コマンドラインの構文, 23
ロケーション, 7, 11
rsoe (参照 Outbound Enabler)
rsoe.exe, 40
構文, 34
rsoe.exe
サービスの開始, 40

S

Sybase ホストのリレーサービス
サブスクリプト, 53
サーバーファームの追加, 54
説明, 53
ログイン, 53

W

Web 拡張機能

Relay Server, 1
Web サーバーを経由した同期
Relay Server, 1

あ

アプリケーションプール
作成, 10
アーキテクチャー
Relay Server, 1

お

オプションセクション
Relay Server 設定ファイル, 30

く

クライアント
Relay Server ファームへの接続, 55

こ

構文
Outbound Enabler, 34

さ

サーバーファーム
Relay Server, 4
サービス
サービスとしてのステイトマネージャーの実
行, 21

す

ステイトマネージャー
オプション, 23
コマンドラインの構文, 23
ステータス管理
Relay Server, 21

せ

接続
Relay Server、Apache 用に同時接続を増やす,
19
設定ファイル
Relay Server, 25
Relay Server のフォーマット, 31

は

配備

Relay Server, 7
バックエンドサーバーセクション
Relay Server 設定ファイル, 29
バックエンドサーバーファーム
Relay Server, 4
バックエンドファームセクション
Relay Server 設定ファイル, 26

ふ

ファーム (参照 Relay Server ファーム)
負荷分散装置
HTTP, 4

も

モバイルデバイス
Relay Server ファームへの接続, 55
