



Postgres Enterprise Manager

リリース **7.10**

Upgrade and Migration Guide

2019 年 10 月 21 日

目次

第 1 章	PEM インストールのアップグレード	2
1.1	Windows ホストでの PEM インストールのアップグレード	2
1.1.1	Windows ホストでの PEM エージェントのアップグレード	4
1.1.2	Windows ホストでの PEM サーバーのアップグレード	10
1.1.3	Windows ホストでの SQL プロファイラーのアップグレード	19
1.2	RPM パッケージと共に Linux ホストにインストールされた PEM のアップグレード	20
1.2.1	RPM を使用してインストールされた PEM エージェントのアップグレード	20
1.2.2	RPM を使用してインストールされた PEM サーバーのアップグレード	21
1.2.3	Linux ホストでの SQL プロファイラーのアップグレード	23
1.3	隔離されたネットワークでの PEM リポジトリの作成	24
1.4	Linux ホスト上のグラフィカルインストーラーでインストールされた PEM のアップグレード	25
1.4.1	グラフィカルインストーラーでインストールされた PEM サーバーのアップグレード	25
1.4.2	グラフィカルインストーラーでインストールされた PEM エージェントのアップグレード	29
1.4.3	Linux ホストでの SQL プロファイラーのアップグレード	32
第 2 章	Backing Postgres データベースのアップグレード	33
第 3 章	Postgres Enterprise Manager™ サーバーの移動	43
第 4 章	トラブルシューティング	54
4.1	pem.alert テーブルが復元に失敗する	54
第 5 章	Postgres Enterprise Manager™ のアンインストール	56
5.1	Windows ホストからの PEM のアンインストール	56
5.2	CentOS または RHEL ホストからの PEM のアンインストール	56
5.3	Debian または Ubuntu ホストからの PEM のアンインストール	57
5.4	SLES ホストからの PEM のアンインストール	57

第 6 章 結論	58
索引	59

このガイドは、Postgres Enterprise Manager (PEM) サーバーのアップグレードに関する詳細情報を提供します。

- **PEM インストールのアップグレード**-このセクションでは、PEM サーバーをあるメジャーバージョンから別のメジャーバージョン（つまり、6.0 から 7.5）にアップグレードする方法について説明します。
- **バックアップデータベースのアップグレード**-このセクションでは、同じバージョンの PEM サーバーを維持しながら、バックアップデータベースのアップグレードに関する詳細情報を提供します。
- **PEM サーバーの移動**-このセクションでは、あるホストから別のホストへの PEM サーバーの移動に関する詳細情報を提供します。
- **トラブルシューティング**-このセクションでは、PEM のアップグレード中に発生する可能性のあるエラーのトラブルシューティングに関する詳細情報を提供します。
- **PEM のアンインストール**-このセクションでは、PEM コンポーネントのアンインストールに関する詳細情報を提供します。

このドキュメントでは、* Postgres *という用語を使用して、PostgreSQL または Advanced Server データベースを意味します。

PEM インストールのアップグレード

PEM インストールのアップグレードプロセスはプラットフォーム固有です。Windows 用の PEM グラフィカルインストーラーを使用して、Windows ホスト上の PEM エージェントまたはサーバーを更新できます。PEM 7.8 リリースより前は、グラフィカルインストーラーまたは RPM を使用して、PEM エージェントまたはサーバーを Linux にインストールできました。PEM バージョン 7.8 以降、Linux 用の PEM グラフィカルインストーラーは廃止されています。Linux ホスト上の PEM エージェントまたはサーバーを下位バージョンから PEM 7.9 に更新するには、RPM を使用する必要があります。

PEM および SQL Profiler インストーラーと RPM へのリンクは、“EnterpriseDB Web サイト<<https://www.enterprisedb.com/enterprise-postgres/edb-postgres-enterprise-manager-pem/>>’_。

1.1 Windows ホストでの PEM インストールのアップグレード

Windows ホスト上の PEM コンポーネントソフトウェアをアップグレードするには、次の順序で新しいバージョンの PEM コンポーネントインストーラーを呼び出すだけです。

1. PEM サーバーノードを除く各監視対象ノードで PEM エージェントインストーラーを呼び出します。
2. PEM サーバーインストーラーを起動します。このインストーラーは、PEM サーバーと PEM サーバーノードにある PEM エージェントの両方をアップグレードします。

インストール中に、コンポーネントインストーラーは既存のインストールを自動的に検出し、アップグレードを実行します。PEM エージェントとサーバーをアップグレードした後、SQL プロファイラーをアップグレードする必要があります。このステップはプラットフォーム固有です。

次のセクションでは、Windows ホストでのアップグレードプロセスについて順を追って説明します。

1.1.1 Windows ホストでの PEM エージェントのアップグレード

PEM エージェントによって現在監視されているシステムをより新しいバージョンの PEM エージェントにアップグレードするには、エージェントが監視しているシステムでプラットフォーム固有の PEM エージェントインストーラーの新しいバージョンをダウンロードして呼び出します。

ダウンロードしたインストーラーのアイコンを右クリックして Run as Administrator を選択すると、インストーラーを起動できます。PEM Agent Setup Wizard が開き、歓迎されます。



図 1 The PEM Agent installer welcome window

続行するには、Next をクリックする前に `License Agreement` を読んで受け入れます。

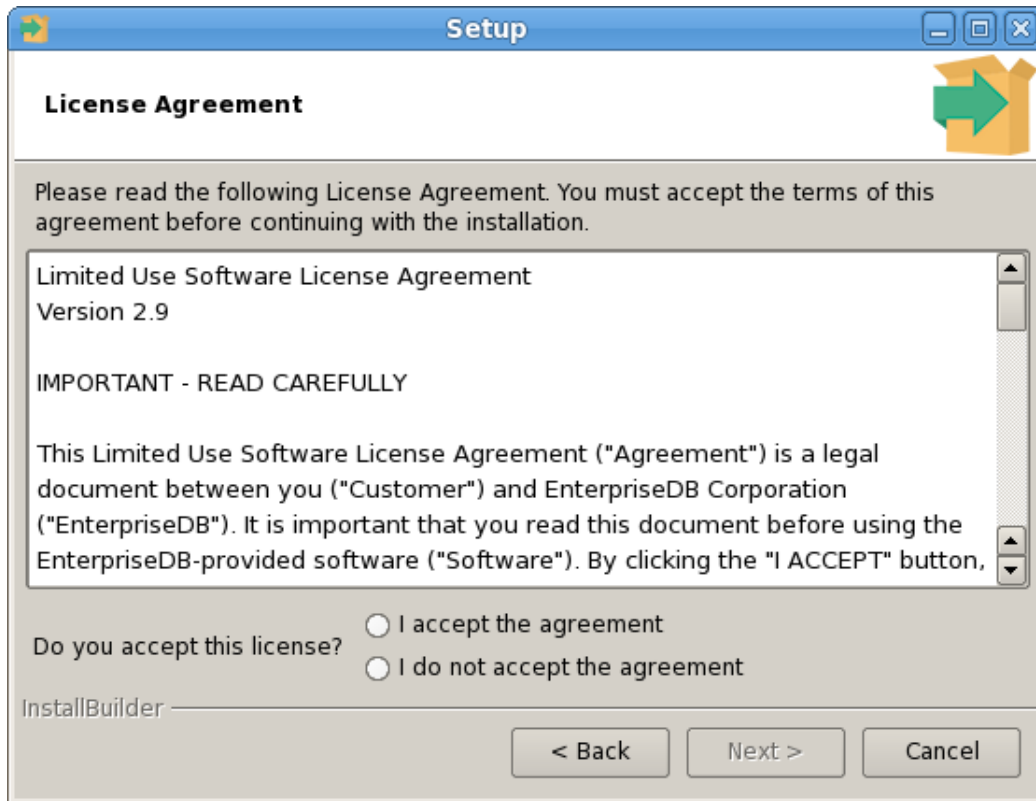


図 2 The PEM license agreement

セットアップウィザードは、既存のエージェントを自動的に検出し、インストールされているバージョンをアップグレードします。Next をクリックして続行します。

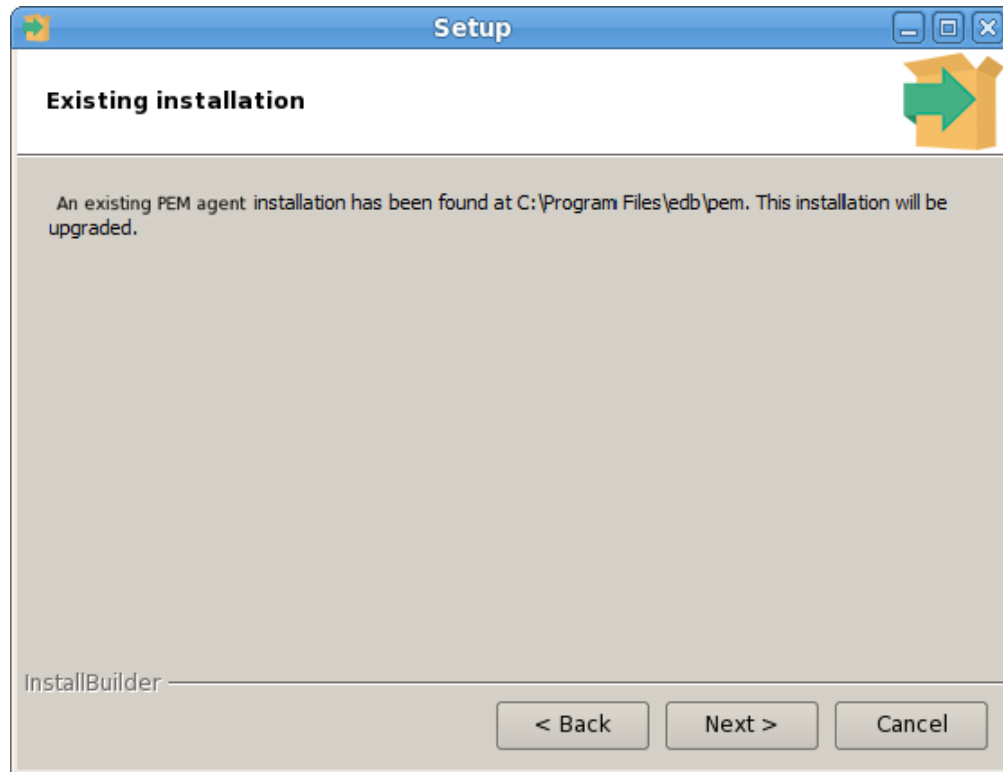


図 3 インストーラーは既存のインストールを検出します

Ready to Install ダイアログがインストールを開始しようとしていることを通知したら、Next をクリックして続行します。

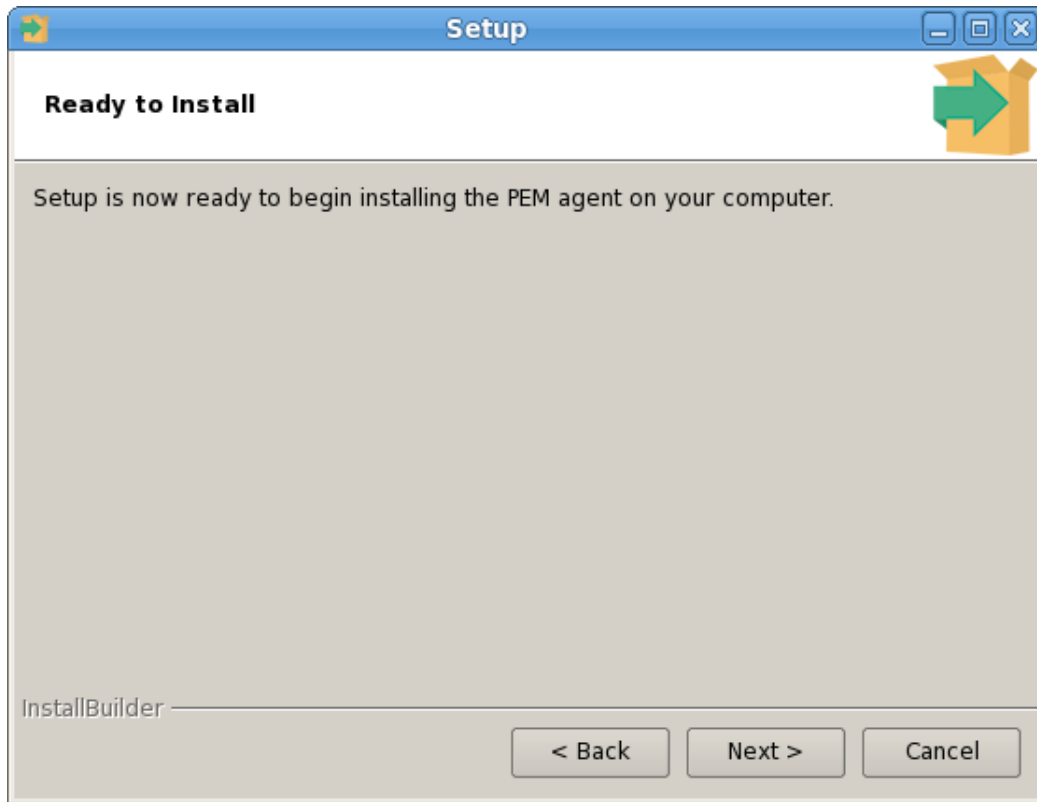


図 4 The PEM Agent will perform the upgrade

セットアップウィザードに進行状況バーが表示され、インストールされている各コンポーネントが通知されます。

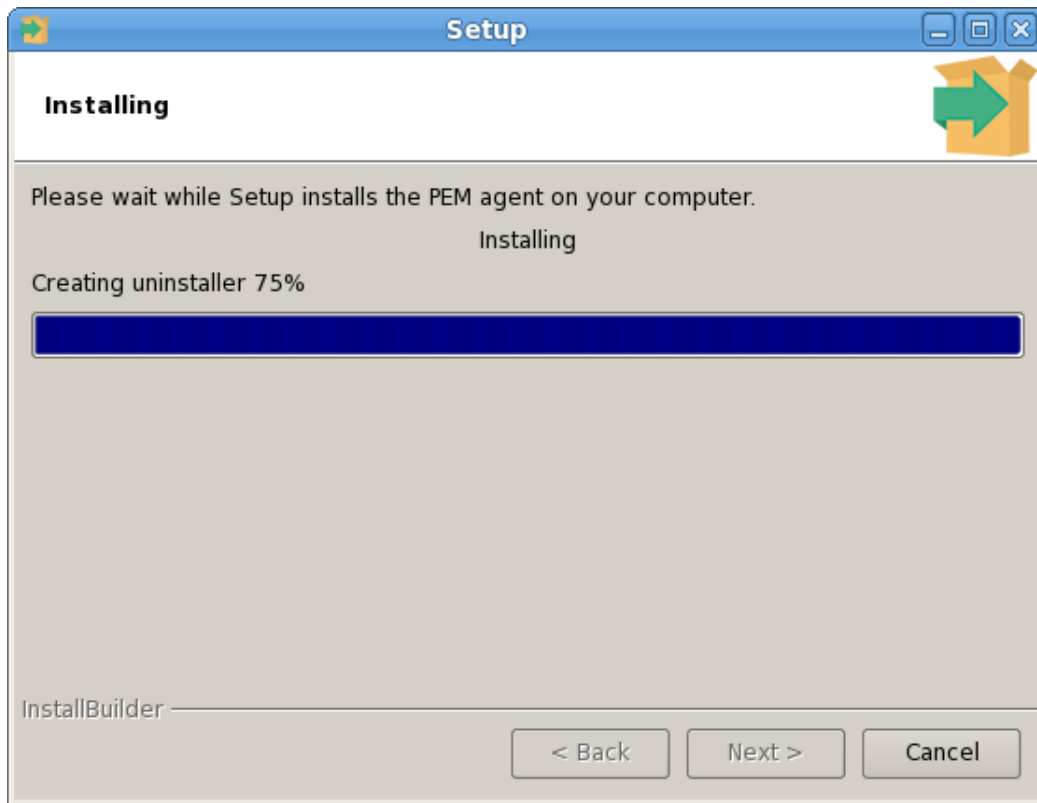


図 5 The PEM Agent will perform the upgrade

インストールが完了すると、PEM Agent Setup Wizard が通知されます。Finish をクリックしてウィザードを終了し、ウィンドウを閉じます。

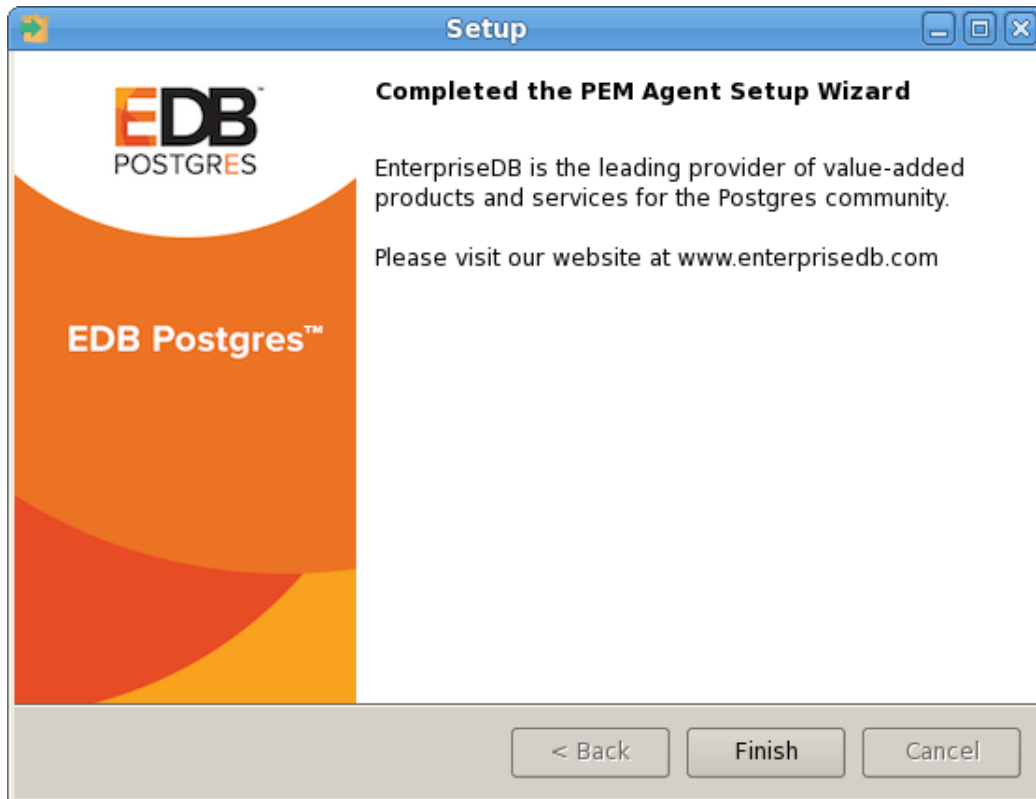


図 6 The PEM Agent Setup Wizard has finished the update

1.1.2 Windows ホストでの PEM サーバーのアップグレード

PEM サーバーインストーラーを使用すると、PEM サーバーのメジャーバージョン間で直接アップグレードできます（たとえば、最初にバージョン 6.0 にアップグレードしなくても、バージョン 5.0 からバージョン 7.5 に直接アップグレードできます）。

ダウンロードしたインストーラーのアイコンを右クリックして Run as Administrator を選択すると、インストーラーを起動できます。

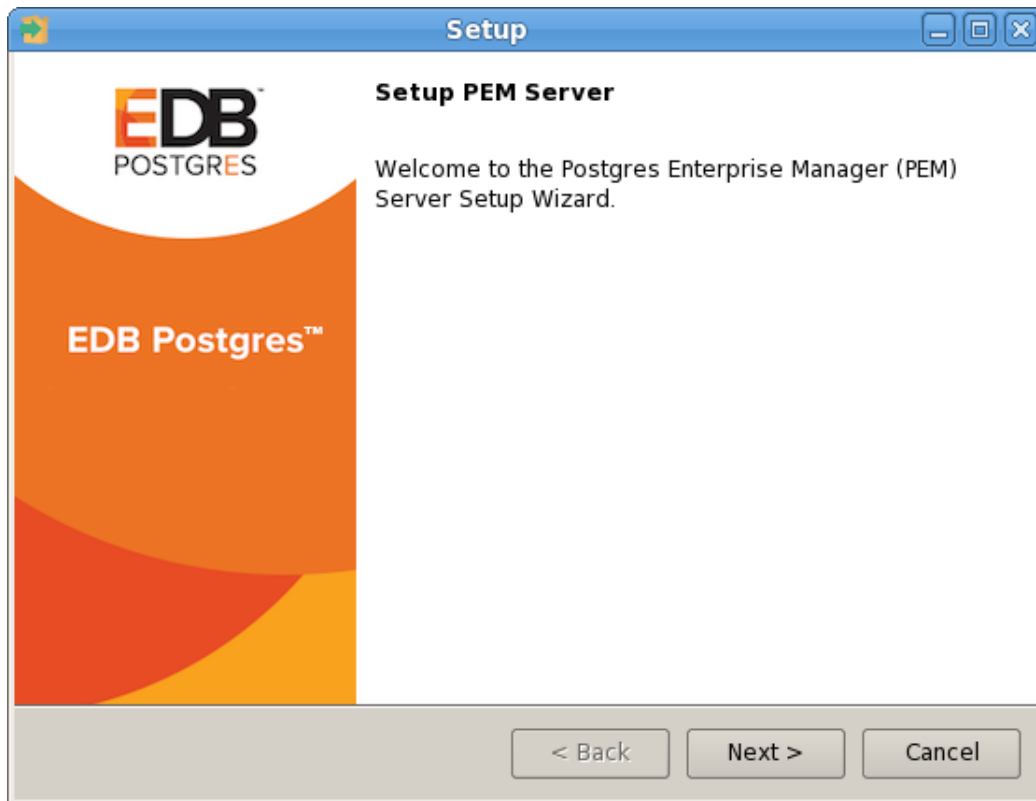


図 7 The PEM Server Setup Wizard

画像に示すように、PEM Server Setup Wizard はあなたを歓迎します。Next をクリックして、License Agreement に進みます。

PEM server setup wizard は License Agreement を受け入れるように促します。ライセンス契約を確認した後、I accept the agreement の横のラジオボタンをチェックし、Next をクリックして Existing installation dialog に進みます。

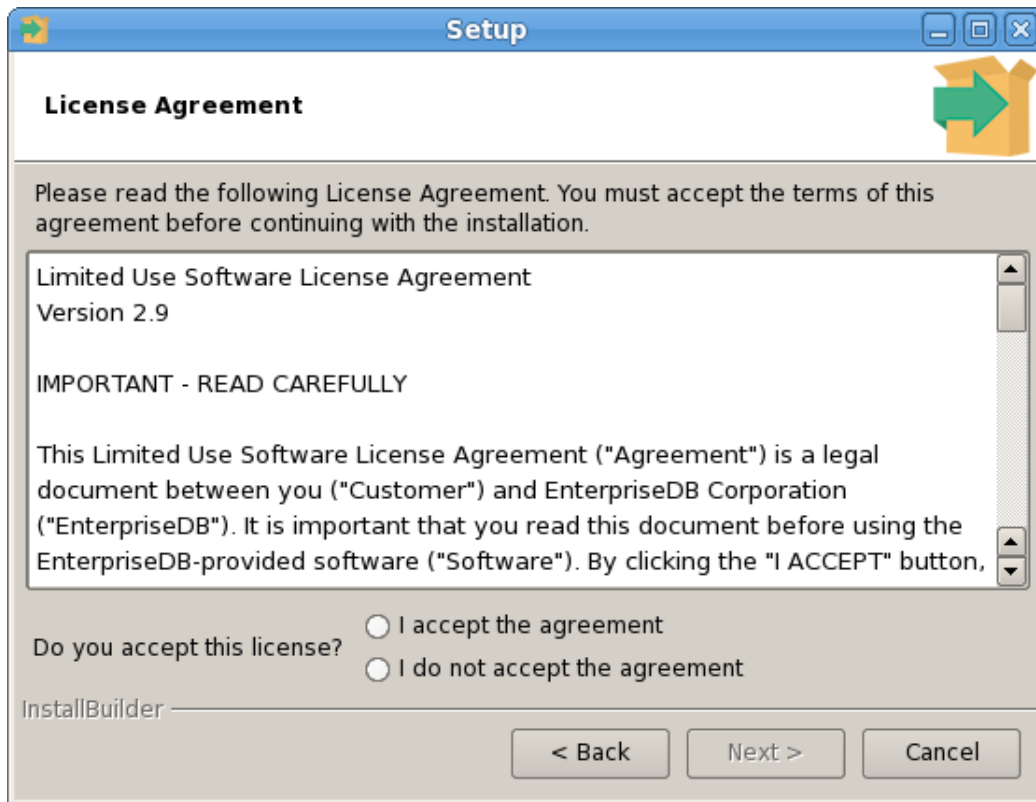


図 8 続行するにはライセンス契約に同意します

ウィザードは、既存の PEM サーバーのインストールについて PEM サーバーホストを確認します。ウィザードがインストールを見つけると、アップグレードを実行します。Next をクリックして続行します。

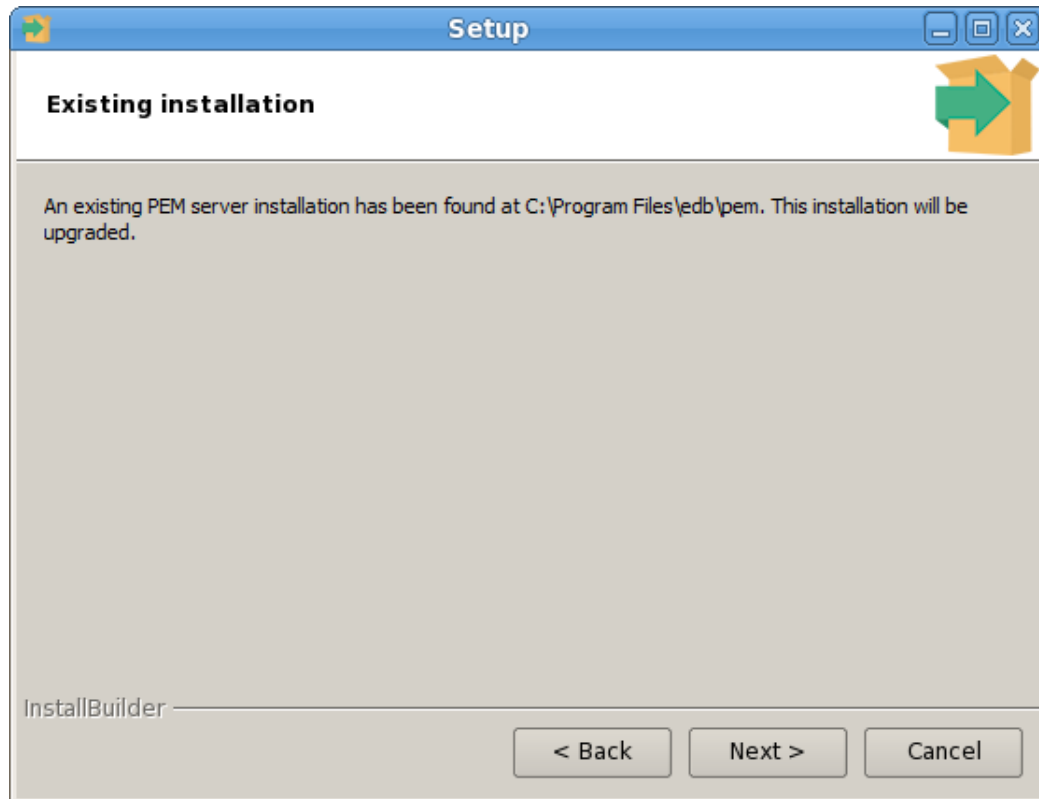


図 9 The PEM server installer detects an existing PEM server Installation

PEM サーバーをアップグレードする前に、ウィザードは新しい PEM サーバーの要件が存在することを確認します。サポートするコンポーネントが見つからない場合、または新しい PEM インストールをサポートしないバージョンの場合、PEM インストールウィザードは依存関係をアップグレードする必要があることを通知し、必要なインストーラーを起動します。

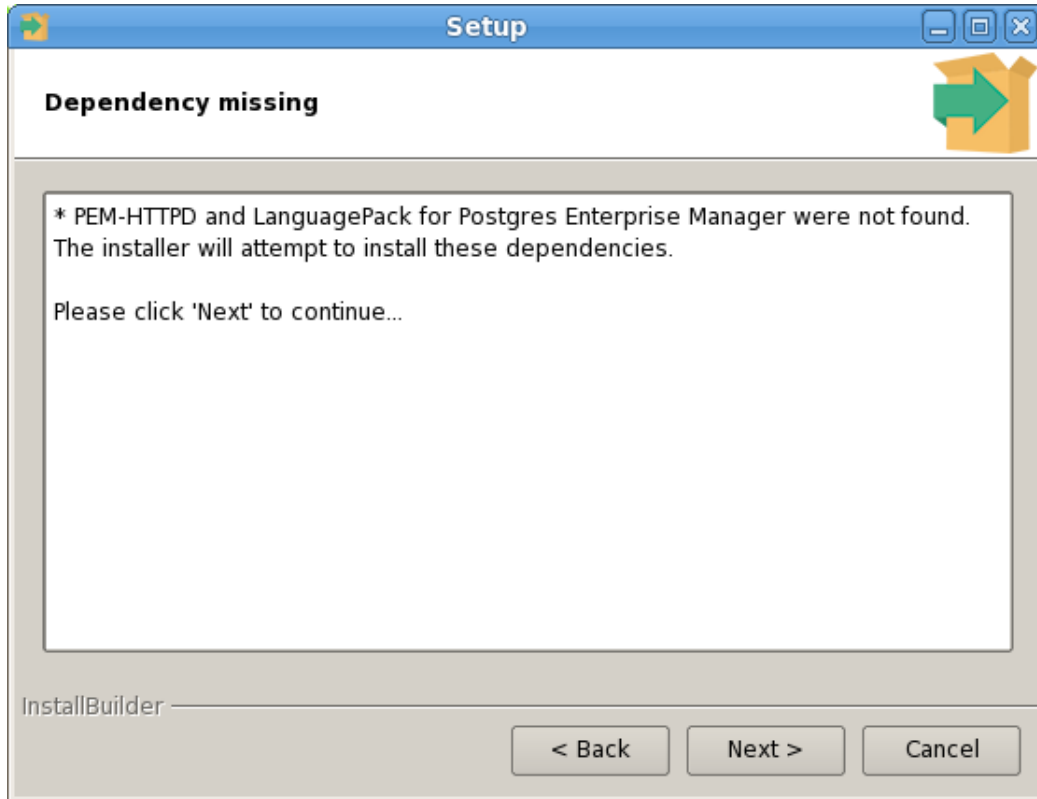


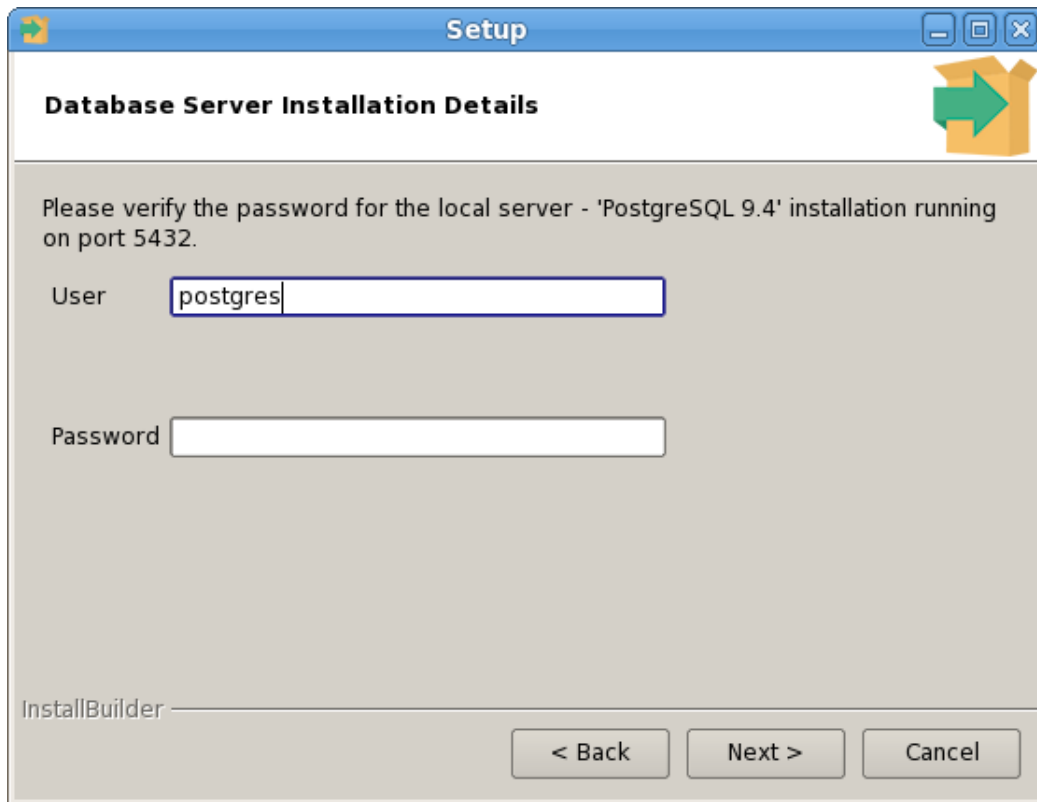
図 10 インストールには依存関係のアップグレードが必要です

インストールウィザードが依存関係のアップグレードを完了すると、フォーカスは PEM サーバーセットアップウィザードに戻ります。

次に、ウィザードは Database Server Installation Details ダイアログを開き、PEM バックアップデータベースのデータベーススーパーユーザーの接続資格情報の入力を求めます。提供:

- User フィールドのデータベーススーパーユーザーの名前。
- Password フィールドのデータベーススーパーユーザーに関連付けられたパスワード。

Next をクリックして続行します。



The screenshot shows a Windows-style dialog box titled "Setup" with a subtitle "Database Server Installation Details". The main text inside the dialog reads: "Please verify the password for the local server - 'PostgreSQL 9.4' installation running on port 5432." Below this text are two input fields. The first is labeled "User" and contains the text "postgres". The second is labeled "Password" and is currently empty. At the bottom of the dialog, there are three buttons: "< Back", "Next >", and "Cancel". The "Next >" button is highlighted. In the bottom left corner, the text "InstallBuilder" is visible.

図 11 バックアップデータベースの接続情報を提供

pemAgent service account ダイアログは、PEM エージェントサービスが実行されるアカウントのパスワードを要求する場合があります。プロンプトが表示されたら、パスワードを入力し、Next をクリックして続行します。

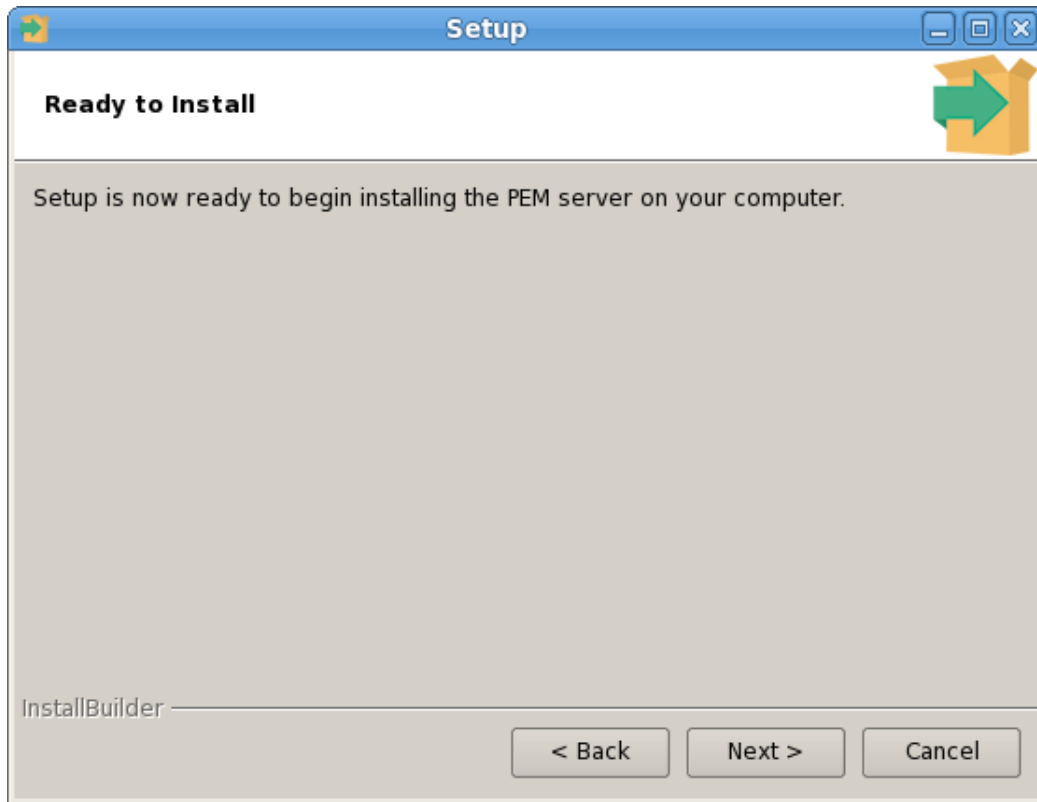


図 12 The PEM Setup Wizard is ready to install the PEM server

Ready to Install ダイアログは、セットアップウィザードがインストールを実行する準備ができたことを通知します。Next をクリックしてインストールを開始します。

インストール中、進行状況バーに更新の進行状況が表示されます。

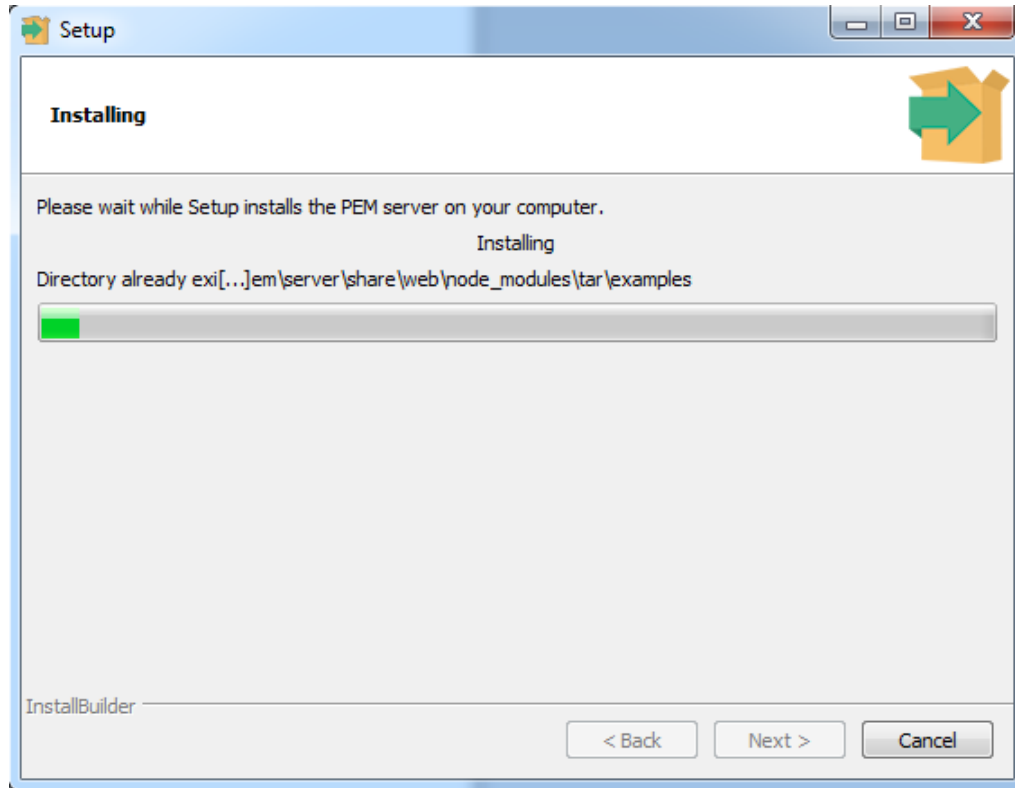


図 13 進行中の PEM サーバーのインストール

PEM サーバー（および PEM サーバーと同じホスト上にあるエージェント）を更新し、Web サービスを構成した後、PEM セットアップウィザードはサービスがリッスンしているポートを通知します。PEM クライアントで PEM サーバーに接続するときに、このポート番号を使用します。

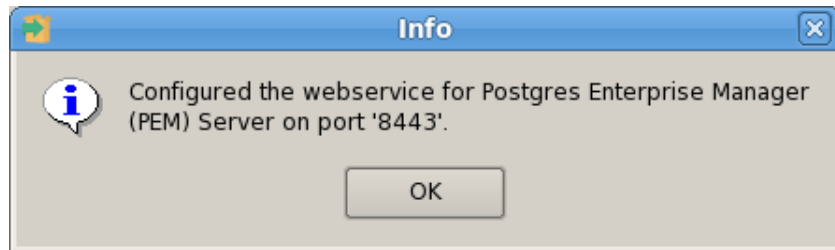


図 14 セットアップウィザードは *PEM Web* サービスを設定します

OK をクリックして、Info ポップアップを閉じます。PEM サーバーセットアップウィザードは、インストールが完了したことを通知します。

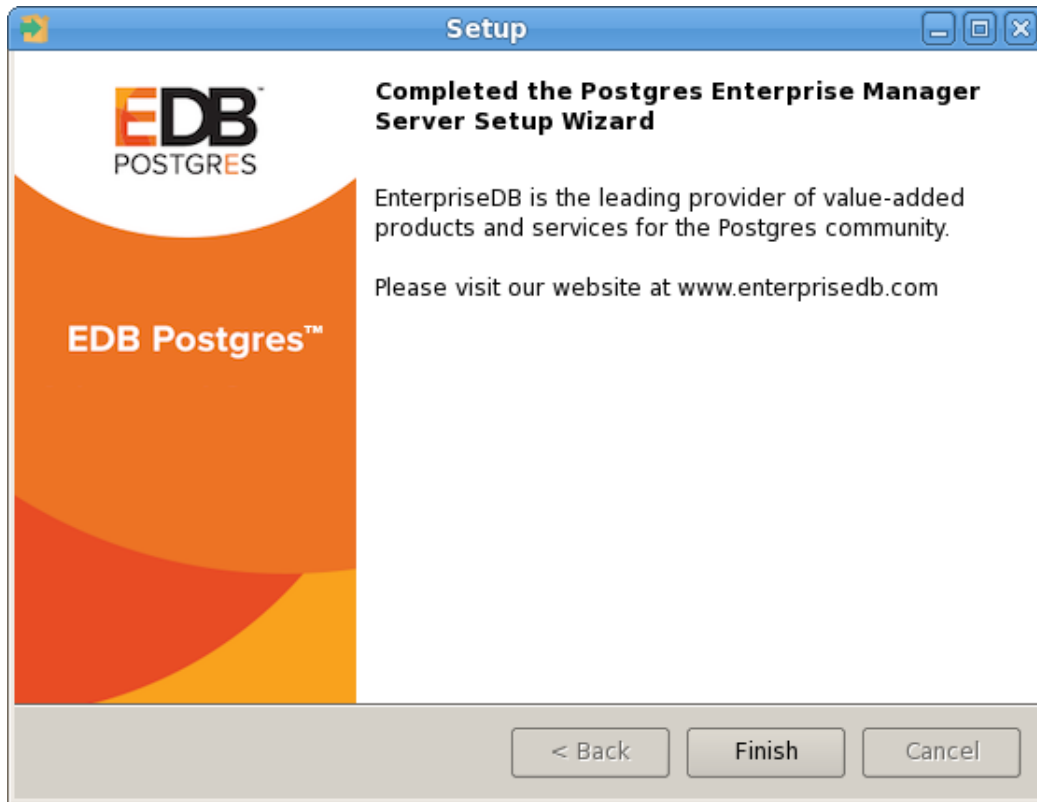


図 15 The PEM Server upgrade is complete

PEM バックエンドデータベースサーバーと PEM-HTTPD を異なるホストにインストールした場合、PEM サーバーインストーラーを 2 回（各ホストに 1 回）実行する必要があります。言語パックインストーラーを抽出し、PEM インストーラーを起動する前に PEM-HTTPD のホストにインストールします。インストーラーを呼び出して言語パックを抽出する場合、次のキーワードを含めます。

```
--extract-languagepack <path>
```

ここで、<path> は、言語パックインストーラーを抽出するための既存のパスを指定します。

PEM サーバーをアップグレードした後、バックアップデータベースをより新しいバージョンにアップグレードすることができます。バックアップデータベースのアップグレードについては、[バックアップデータベースのアップグレードを参照してください](#)。

1.1.3 Windows ホストでの SQL プロファイラーのアップグレード

Windows ホストで SQL プロファイラーを使用している場合、Windows は実行またはメモリにロードされたファイルをロックします。ロックされたファイルを解放するには、アップグレードを実行する前に Postgres サーバーを停止する必要があります。

Windows では、Services ダイアログを使用してサービスを制御できます。Services ダイアログを開くには、Control Panel から System and Security menu までナビゲートします。Administrative Tools を選択し、Services アイコンをダブルクリックします。Services ダイアログが開いたら、リスト内のサービス名を強調表示し、ダイアログに表示されるオプションを使用して Stop サービスを選択します。

Postgres サーバーを停止した後:

1. `uninstall-sql-profiler.sql` スクリプトを呼び出して、各ノードの既存の SQL プロファイラーのエリセットを削除します。

デフォルトでは、スクリプトは Advanced Server または PostgreSQL インストールの下の `share/contrib` ディレクトリにあります。

2. プロファイリングする各ノードで新しい SQL Profiler インストーラーを呼び出します。

次に、Postgres サーバーを再起動して、PEM クライアントからノードのプロファイリングを再開します。PEM コンポーネントを更新したら、バックアップデータベースを更新できます。

1.2 RPM パッケージと共に Linux ホストにインストールされた PEM のアップグレード

以下のセクションでは、Linux ホストでのアップグレードプロセスについて順を追って説明します。

前提条件:

バージョン 10 よりも低いバージョンの Postgres または Advanced Server を使用している場合、アップグレードの前に、PEM サーバーがインストールされているシステムにバージョン 10 以上の `libs` パッケージをインストールする必要があります。以下のコマンドを使用して、`libs` バージョン 10 以降をインストールします。

- **Advanced Server の場合:** `yum install edb-as<version>-server-libs`
- **Postgres の場合:** `yum install postgresql<version>-libs`

ここで、`* version *`は、`libs` パッケージをインストールする Postgres または Advanced Server のバージョンです。

1.2.1 RPM を使用してインストールされた PEM エージェントのアップグレード

RPM パッケージを使用して、パッケージによって最初にインストールされた既存のエージェントをアップグレードできます。アップグレードプロセスでは、PEM エージェントの構成ファイルは更新されません。新しいエージェントをインストールした後、既存のエージェントの構成ファイルを新しいインストール場所に手動でコピーする必要があります。

RPM パッケージを使用して PEM エージェントをアップグレードするには、次のコマンドを使用する必要があります。

```
yum upgrade pem-agent
```

ここで、`* version *`は、`libs` パッケージをインストールする Postgres または Advanced Server のバージョンです。

PEM サーバーに登録されている CentOS 6.x 上の下位バージョンの PEM エージェントが 7.10 にアップグレードされると、次の警告が表示される場合があります。

```
warning:%postun(edb-pem-agent-7.9.0-3.rhel6.x86_64) scriptlet failed, exit
status 1
```

```
Non-fatal POSTUN scriptlet failure in rpm package edb-pem-agent
```

上記の警告にもかかわらず、アップグレードは正常に完了し、アップグレード後にエージェントが自動的に再起動されます。

1.2.2 RPM を使用してインストールされた PEM サーバーのアップグレード

RPM パッケージで PEM サーバーをインストールした場合、RPM を使用して PEM サーバーをアップグレードできます。

分離されたネットワークのマシンにインストールされている PEM サーバーをアップグレードする場合は、PEM サーバーをアップグレードする前に、そのマシンに PEM リポジトリを作成する必要があります。分離されたネットワークで PEM リポジトリを作成する方法の詳細については、“分離されたネットワークでの PEM リポジトリの作成”を参照してください。。

RPM パッケージを使用して既存の RPM インストールをアップグレードするには、以下を行う必要があります。

1. パッケージマネージャーを使用して、インストールされている PEM サーバーのバージョンをアップグレードします。

```
yum upgrade edb-pem edb-pem-server
```

2. アップグレードされたサーバーを構成します。

```
/usr/edb/pem/bin/configure-pem-server.sh
```

構成スクリプトを呼び出すときに、構成プロパティを指定するコマンドラインオプションを含めることができます。スクリプトは、コマンドラインで省略した値の入力を求めます。受け入れられるオプションは次のとおりです。

オプション	説明
-acp	PEM エージェントの証明書パスを定義します。デフォルトは/root/.pem です。
-ci	エージェントがサーバーに接続する、サーバーの pg_hba.conf ファイルに追加される CIDR 形式のネットワークアドレス範囲。たとえば、192.168.1.0 / 24。デフォルトは 0.0.0.0/0 です。
-dbi	データベースサーバーのインストールディレクトリ。たとえば、Advanced Server の場合は /usr / edb / as10、PostgreSQL の場合は /usr / postgresql-10 です。
-ds	PEM データベースサーバーのユニットファイル名。Advanced Server の場合、デフォルトのファイル名は edb-as-10 です。PostgreSQL の場合、postgresql-10 です。
-ホ	PEM データベースサーバーのホストアドレス。
-p	PEM データベースサーバーのポート番号。
-ps	pemagent のサービス名。デフォルト値は pemagent です。
-sp	PEM データベースサーバーのスーパーユーザーパスワード。この値は必須です。
-su	PEM データベースサーバーのスーパーユーザー名。
-t	インストールの種類:構成が Web サービスとバックアップデータベース用の場合は 1、Web サービスを構成する場合は 2、バックアップデータベースを構成する場合は 3 を指定します。3 を指定する場合、データベースはローカルホストに存在する必要があることに注意してください。
-un	PEM サーバーをアンインストールします。
-h	ヘルプを表示します。

コマンドラインで構成プロパティを指定しない場合、スクリプトによって値の入力が求められます。スクリプト関連のヘルプを表示するには、次のコマンドを使用します。

```
/usr/edb/pem/bin/configure-pem-server.sh --help
```

PEM サーバー設定ファイルを実行した後、バージョン固有のサービス制御コマンドを使用して、httpd サービスを再起動します。

RPM パッケージを使用して PEM サーバーまたは PEM エージェントをインストールまたは構成する方法の詳細については、次の場所にある* PEM インストールガイド*を参照してください。

<http://www.enterprisedb.com/products-services-training/products/documentation/enterpriseedition>

1.2.3 Linux ホストでの SQL プロファイラーのアップグレード

Linux ホストにある SQL Profiler インストールをアップグレードするには:

1. `uninstall-sql-profiler.sql` スクリプトを呼び出して、各ノードの既存の SQL プロファイラーのエリセットを削除します。

デフォルトでは、グラフィカルインストーラーでインストールされた Linux ホストで Advanced Server を使用している場合、スクリプトは Advanced Server インストールの `share/contrib` ディレクトリにあります。

Linux ホストで PostgreSQL インストールを使用している場合、スクリプトは PostgreSQL インストールの下の `share/postgresql/contrib` ディレクトリにあります。

2. プロファイリングするノードで次のコマンドを使用します。

```
yum install postgresql<pg_version>-sqlprofiler-<sql_profiler_version>
```

ここで、* `pg_version` は *postgres* のバージョンであり、`sqlprofiler_version` * は SQL プロファイラーのバージョンです。

PEM コンポーネントを更新したら、バックアップデータベースを更新できます。

1.3 隔離されたネットワークでの PEM リポジトリの作成

PEM をインストールするサーバーが EnterpriseDB リポジトリに直接アクセスできない場合、PEM RPM パッケージのホストとして機能するローカルリポジトリを作成できます。これは、必要な手順の概要であることに注意してください。個々のネットワークのプロセスを変更する必要がある場合があります。ローカルリポジトリを作成して使用するには、以下を行う必要があります。

1. 次のコマンドを使用して、インターネットアクセスが可能なマシンに PEM に必要なすべての依存関係をダウンロードします。

```
yum install yum-plugin-downloadonly

mkdir /tmp/<pem_dir>

yum install --downloadonly --downloadaddir=/tmp/<pem_dir>/ edb -pem edb-pem-server

mkdir /tmp/<epel_dir>

yum install --downloadonly --downloadaddir=/tmp/<epel_dir>/ epel-release*
```

ここで、<pem_dir> と <epel_dir> は RPM をダウンロードするために作成するローカルディレクトリです。

2. ディレクトリ /tmp/<pem_dir> および /tmp/<epel_dir> を分離されたネットワーク内のマシンにコピーします。
3. リポジトリを作成します。

```
createrepo /tmp/<pem_dir>

createrepo /tmp/<epel_dir>
```

5. 以下を指定する接続情報を使用して、/etc/yum.repos.d/pem.repo というリポジトリ構成ファイルを作成します。

```
[pemrepo]
name=PEM Repository
baseurl=file:///tmp/pem78/
enabled=1
gpgcheck=0
```

6. 以下を指定する接続情報を使用して、/etc/yum.repos.d/epel.repo というリポジトリ構成ファイルを作成します。

```
[epelrepo]
name=epel Repository
baseurl=file:///tmp/pem78/
```

(次のページに続く)

(前のページからの続き)

```
enabled=1
gpgcheck=0
```

ローカルリポジトリの場所と接続情報を指定したら、yum コマンドを使用して PEM サーバーをインストールまたはアップグレードできます。

- PEM サーバーをインストールするには:

```
yum install edb-pem edb-pem-server
```

- PEM サーバーをアップグレードするには:

```
yum upgrade edb-pem edb-pem-server
```

ローカル yum リポジトリの作成の詳細については、<https://wiki.centos.org/HowTos/CreateLocalRepos> にアクセスしてください。

1.4 Linux ホスト上のグラフィカルインストーラーでインストールされた PEM のアップグレード

以下のセクションでは、Linux ホスト上のグラフィカルインストーラーを介して実行された PEM インストールのアップグレードプロセスについて説明します。

1.4.1 グラフィカルインストーラーでインストールされた PEM サーバーのアップグレード

前提条件:

バージョン 10 よりも低いバージョンの Postgres または Advanced Server を使用している場合、アップグレードの前に、PEM サーバーがインストールされているシステムにバージョン 10 以上の `libs` パッケージをインストールする必要があります。以下のコマンドを使用して、`libs` バージョン 10 以降をインストールします。

- Advanced Server の場合: `yum install edb-as<version>-server-libs`
- Postgres の場合: `yum install postgresql<version>-libs`

ここで、* `version` *は、`libs` パッケージをインストールする Postgres または Advanced Server のバージョンです。

グラフィカルインストーラーによってインストールされたときの PEM サーバーのデフォルトのインストール場所は `/opt/edb/pem` です。次の例では、サーバーのインストール場所を `<PEM_installation_path>` に置き換えます。

1. PEM からログアウトします。

2. PEM サーバーで PEMHTTPD サービスを停止します。PEM サーバーと Web サーバーが 2 つの異なるシステム上にある場合、Web サーバーでコマンドを実行します。

```
systemctl stop PEMHTTPD
```

3. PEM サーバーに epel パッケージをインストールします。PEM サーバーと Web サーバーが 2 つの異なるシステムにある場合、PEM サーバーと Web サーバーでコマンドを実行します。

```
yum install epel-release/*
```

4. EDB リポジトリ構成ファイルをインストールします。

```
yum install http://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.  
rpm
```

5. リポジトリ構成ファイルのインストールが完了したら、dependencies および tools リポジトリ定義を変更し、リポジトリ定義が有効になっていることを確認し、正しいリポジトリ資格情報を提供します。ファイルの場所は次のとおりです。

```
/etc/yum.repos.d/edb.repo
```

6. yum makecache コマンドを使用して、現在有効になっているリポジトリのメタデータをダウンロードします。コマンドが完了したら、利用可能なパッケージをチェックして、リストに最新の PEM サーバーが含まれていることを確認します。

```
yum makecache
```

```
yum list edb-pem-server
```

7. PEM サーバーに PEM サーバー RPM をインストールします。PEM サーバーと Web サーバーが 2 つの異なるシステムにある場合、PEM サーバーと Web サーバーでコマンドを実行します。インストールが完了したら、yum info コマンドを使用してインストールの詳細を確認します。

```
yum install edb-pem-server
```

```
yum info edb-pem-server
```

8. インストール後、agent.cfg ファイルを現在の場所（グラフィカルインストーラーに必要な場所）から RPM パッケージに必要な場所にコピーします。

```
cp /<PEM_installation_path>/agent/etc/agent.cfg /usr/edb/pem/agent/etc/  
agent.cfg
```

9. 次の場所にあるエージェント構成ファイルを開きます。

```
/usr/edb/pem/agent/etc/agent.cfg
```

次に、ca_file パラメータの値を設定します:

```
ca_file=/usr/libexec/libcurl-pem/share/certs/ca-bundle.crt
```

10. pem.db ファイル（およびその他の必要なファイル）を RPM インストール場所にコピーし、ファイルの所有権を変更します。PEM サーバーと Web サーバーが 2 つの異なるシステムにある場合、Web サーバーで以下のコマンドを実行します。

```
cp -r /<PEM_installation_path>/server/share/pemhome/.pem/* /var/lib/
pemhome/.pem/

chown -R pem:pem /var/lib/pemhome/.pem/
```

11. グラフィカルインストーラーによって識別された場所から RPM の場所に passwd ファイルのホームディレクトリを変更します。PEM サーバーと Web サーバーが 2 つの異なるシステムにある場合、PEM サーバーと Web サーバーでコマンドを実行します。

```
usermod -m -d /var/lib/pemhome pem

cat /etc/passwd | grep pem
```

12. PEM サービスファイルとエージェント証明書のバックアップを作成します。

```
cp /usr/lib/systemd/system/pemagent.service /usr/lib/systemd/system/
pemagent.service_bkp

mv /root/.pem/agent1.key /root/.pem/agent1.key.bkp

mv /root/.pem/agent1.crt /root/.pem/agent1.crt.bkp
```

13. PEM サーバーおよび Web サーバーマシンからグラフィカルアンインストーラーを使用して PEM サーバーをアンインストールします。

```
/<PEM_installation_path> /server/uninstall-pemserver
```

14. PEM サーバーおよび Web サーバーで PEM RPM 構成スクリプトを実行します。プロンプトが表示されたら、バックアップデータベースの詳細を指定します。エラーを生成せずにスクリプトを実行する必要があります。

```
/usr/edb/pem/bin/configure-pem-server.sh
```

15. サービスファイルのバックアップとエージェント証明書を PEM サーバーの元の場所に復元します。

```
cp /usr/lib/systemd/system/pemagent.service_bkp /usr/lib/systemd/system/
pemagent.service

mv /root/.pem/agent1.crt.bkp /root/.pem/agent1.crt

mv /root/.pem/agent1.key.bkp /root/.pem/agent1.key
```

16. PEM サーバーと Web サーバーで pemagent サービスを有効にします。Web サーバーで pemagent および httpd サービスを開始します。

```
systemctl enable pemagent
```

```
systemctl start pemagent
```

```
systemctl start httpd
```

17. PEM Web インターフェイスを起動します。サーバーとエージェントをチェックして、**PEM** バージョン、サーバーステータス、およびスキーマバージョンを確認します。この時点で、すべてが稼働しているはずです。使用されなくなった PEMHTTD サービスを Web サーバーからアンインストールできるようになりました。

1.4.2 グラフィカルインストーラーでインストールされた PEM エージェントのアップグレード

グラフィカルインストーラーによってインストールされたときの PEM サーバーのデフォルトのインストール場所は `/opt/edb/pem` です。次の例では、サーバーのインストール場所を `<PEM_installation_path>` に置き換えます。

1. バージョン固有のコマンドを使用して、`pemagent` サービスを停止します。

- RHEL または CentOS 7.x ホストの場合:

```
systemctl stop pemagent
```

- RHEL または CentOS 6.x ホストの場合:

```
/etc/init.d/pemagent stop (RHEL -6)
```

2. サポートする `epel` パッケージをインストールします。

```
yum install epel-release/*
```

3. EDB リポジトリ構成ファイルをインストールします。

```
yum install http://yum.enterprisedb.com/edbrepos/edb-repo-latest.noarch.rpm
```

4. リポジトリ構成ファイルのインストールが完了したら、`dependencies` および `tools` リポジトリ定義を変更し、リポジトリ定義が有効になっていることを確認し、正しいリポジトリ資格情報を提供します。ファイルの場所は次のとおりです。

```
/etc/yum.repos.d/edb.repo
```

5. The `yum makecache` command downloads the metadata for the currently enabled repositories; when the command completes, check the available packages to confirm that the list includes the latest PEM agent:

```
yum makecache
```

```
yum list edb-pem-agent
```

6. PEM エージェント RPM をインストールします。インストールが完了したら、`yum info` コマンドを使用して PEM エージェントのインストール情報を確認できます。

```
yum install edb-pem-agent
```

```
yum info edb-pem-agent
```

7. インストール後、PEM エージェント設定ファイル (`agent.cfg`) を以前場所から RPM インストーラーに必要な場所にコピーします。

```
cp /PEM_installation_path/agent/etc/agent.cfg /usr/edb/pem/agent/etc/agent.cfg
```

8. 次の場所にあるエージェント構成ファイルを開きます。

```
/usr/edb/pem/agent/etc/agent.cfg
```

次に、`ca_file` パラメータの値を設定します:

```
ca_file=/usr/libexec/libcurl-pem/share/certs/ca-bundle.crt
```

9. サービスファイルとエージェント証明書のバックアップを作成します。

- RHEL または CentOS 7.x では、次のコマンドを使用してサービスファイルをバックアップします。

```
cp /usr/lib/systemd/system/pemagent.service /usr/lib/systemd/system/pemagent.service_bkp
```

- RHEL または CentOS 6.x では、次のコマンドを使用してサービスファイルをバックアップします。

```
cp /etc/init.d/pemagent_bkp /etc/init.d/pemagent
```

次に、エージェント証明書をコピーします。次のコマンドでは、`<agent_id>` はエージェント識別子 (`agent2` または `agent3` など) を指定する必要があります。

```
mv /root/.pem/<agent_id>.key /root/.pem/<agent_id>.key.bkp
```

```
mv /root/.pem/<agent_id>.crt /root/.pem/<agent_id>.crt.bkp
```

10. `bitrock uninstaller` を使用して PEM エージェントをアンインストールします。

```
/PEM_installation_path/agent/uninstall-pemagent
```

11. バージョン固有のコマンドを使用して、サービスファイルのバックアップとエージェント証明書を元の場所に復元します。例:

- RHEL または CentOS 7.x ホストの場合:

```
cp /usr/lib/systemd/system/pemagent.service_bkp /usr/lib/systemd/system/pemagent.service
```

- RHEL または CentOS 6.x ホストの場合:

```
cp /etc/init.d/pemagent /etc/init.d/pemagent_bkp
```

次に、エージェント証明書ファイルを移動します。次のコマンドでは、`<agent_id>` はエージェント識別子 (`agent2` または `agent3` など) を指定する必要があります。

```
mv /root/.pem/<agent_id>.key.bkp /root/.pem/<agent_id>.key
```

```
mv /root/.pem/<agent_id>.crt.bkp /root/.pem/<agent_id>.crt
```

12. **pemagent** サービスを有効にして、**pemagent** と **httpd** を開始します。

- RHEL または CentOS 7.x ホストでは、次のコマンドを使用します。

```
systemctl enable pemagent
```

```
systemctl start pemagent
```

- RHEL または CentOS 6.x ホストの場合:

```
/etc/init.d/pemagent start
```

この時点で、**PEM** エージェントが稼働しているはずです。 **PEM Web** インターフェイスを使用して、エージェントのバージョンとステータスを確認できます。

1.4.3 Linux ホストでの SQL プロファイラーのアップグレード

Linux ホスト上の SQL Profiler インストールをアップグレードするには:

1. `uninstall-sql-profiler.sql` スクリプトを呼び出して、各ノードの既存の SQL プロファイラーのエリセットを削除します。

デフォルトでは、グラフィカルインストーラーでインストールされた Linux ホストで Advanced Server を使用している場合、スクリプトは Advanced Server インストールの `share/contrib` ディレクトリにあります。

Linux ホストで PostgreSQL インストールを使用している場合、スクリプトは PostgreSQL インストールの下の `share/postgresql/contrib` ディレクトリにあります。

2. 次に、プロファイリングする各ノードで次のコマンドを使用します: `yum install postgresql-<sql_version>-sqlprofiler-<sqlprofiler_version>`

ここで、`*sql_version*` は postgres のバージョンであり、`<sqlprofiler_version>` SQL プロファイラーのバージョンです。

PEM コンポーネントを更新したら、バックアップデータベースを更新できます。

第 2

Backing Postgres データベースのアップグレード

PEM コンポーネントと PEM バックアップデータベースの両方を更新する場合、バックアップデータベースを更新する前に PEM コンポーネントの更新（サーバー、エージェント、およびクライアント）を実行する必要があります。PEM コンポーネントソフトウェアの更新の詳細については、[PEM インストールのアップグレード](#)を参照してください。

このセクションで説明する更新プロセスでは、pg_upgrade ユーティリティを使用して、バックアップサーバーのあるバージョンからより新しいバージョンに移行します。pg_upgrade は、Postgres の任意のバージョン（バージョン 9.3 以降）と、同じプラットフォームでサポートされる Postgres の後続のリリースとの間の移行を容易にします。

pg_upgrade は同じタイプのサーバー間のデータ転送をサポートします。たとえば、pg_upgrade を使用して、PostgreSQL 9.6 バックアップデータベースから PostgreSQL 10 バックアップデータベースにデータを移動できますが、Advanced Server 10 バックアップデータベースには移動できません。別の種類のバックアップデータベースに（つまり、PostgreSQL サーバーから Advanced Server に）移行する場合は、[Postgres Enterprise Manager™ サーバーの移動](#)を参照してください。

pg_upgrade の使用に関する詳細については、次を参照してください。

<http://www.postgresql.org/docs/current/static/pgupgrade.html>

ステップ 1-更新されたバックアップデータベースインストーラーをダウンロードして呼び出します

PostgreSQL および Advanced Server のインストーラーは、EnterpriseDB Web サイトから入手できます。

<http://www.enterprisedb.com>

アップグレードするサーバーバージョンのインストーラーをダウンロードした後、PEM サーバーのホストでインストーラーを起動します。インストールウィザードの画面上の指示に従って、Postgres サーバーを構成およびインストールします。

オプションで、カスタム構築の PostgreSQL サーバーを PEM バックアップデータベースのホストとして使用できます。ポート 5432 でリッスンする PostgreSQL バックアップデータベースからアップグレードする場合、新しいサーバーは別のポートでリッスンするように設定する必要があることに注意してください。

ステップ 2-新しいサーバーで SSL ユーティリティを構成する

新しいバックアップデータベースは、現在のバックアップデータベースが実行されているのと同じバージョンの `sslutils` を実行している必要があります。EnterpriseDB インストーラーで使用される SSL Utils パッケージは、次の場所からダウンロードできます。

<http://www.enterprisedb.com/downloads/component-source-code>

Advanced Server を新しいバックアップデータベースとして使用する場合、`sslutils` 拡張子を手動で追加する必要はありません。`sslutils` を設定するプロセスはプラットフォーム固有です。

Linux で

Linux を使用している場合、アーカイブされた SSL Utils ファイルのバージョンを次からダウンロードできます。

<http://www.enterprisedb.com/downloads/component-source-code>

ダウンロードが完了したら、`sslutils` フォルダを抽出し、アップグレード先の Postgres バージョンの Postgres インストールディレクトリに移動します。

コマンドラインを開き、スーパーユーザー権限を引き継ぎ、PATH 環境変数の値を設定して、`make` が `pg_config` プログラムを見つけられるようにします。

```
export PATH=$PATH:/opt/<Postgres>/<x.x>/bin/
```

どこで:

Postgres specifies either of the following:

- PostgreSQL サーバーにアップグレードする場合は PostgreSQL。
- Advanced Server サーバーにアップグレードする場合は PostgresPlus。
- `x.x` は、移行先の Postgres のバージョンを指定します。

次に、`yum` を使用して `sslutil` の依存関係をインストールします。

```
yum install openssl-devel
```

`sslutils` フォルダに移動し、次を入力して `sslutils` パッケージをビルドします。

```
make USE_PGXS=1
```

```
make USE_PGXS=1 install
```

Windows で

sslutils は、元のバックアップデータベースで sslutils をコンパイルするために使用したのと同じコンパイラーを使用して、新しいバックアップデータベースでコンパイルする必要があります。PostgreSQL ワンクリックインストーラー（EnterpriseDB から）または Advanced Server インストーラーを使用してインストールされた Postgres データベースに移動する場合は、Visual Studio を使用して sslutils をビルドします。PostgreSQL 9.3 以降にアップグレードする場合は、Visual Studio 2010 を使用してください。

Windows で特定のバージョンの Postgres をビルドする方法の詳細については、そのバージョンのコアドキュメントを参照してください。コアドキュメントは、PostgreSQL プロジェクトの Web サイトで入手できます。

<http://www.postgresql.org/docs/>

または次の EnterpriseDB Web サイトで:

<http://www.enterprisedb.com/products-services-training/products/documentation/enterprisedition>

プロセスの具体的な詳細はプラットフォームとコンパイラーによって異なりますが、各プラットフォームの基本的な手順は同じです。次の例は、32 ビット Windows システムでの PostgreSQL の OpenSSL サポートのコンパイルを示しています。

OpenSSL 拡張機能をコンパイルする前に、ご使用の Windows バージョンの OpenSSL を見つけてインストールする必要があります。OpenSSL インストーラーを呼び出す前に、前提条件の再配布可能ファイル（vcredist_x86.exe など）をダウンロードしてインストールする必要がある場合があります。

OpenSSL をインストールした後、次から入手できる sslutils ユーティリティパッケージをダウンロードして解凍します。

<http://www.enterprisedb.com/downloads/component-source-code>

解凍した sslutils フォルダーを Postgres インストールディレクトリ（つまり C:/ProgramFiles/PostgreSQL/9.<x>）にコピーします

Visual Studio コマンドラインを開き、sslutils ディレクトリに移動します。次のコマンドを使用して、sslutils を作成します:

```
SET USE_PGXS=1

SET GETTEXT_PATH=/ <path_to_gettext>

SET OPENSSL_PATH=/ <path_to_openssl>

SET PG_PATH=/ <path_to_pg_installation_dir>

SET ARCH=x86

msbuild sslutils.proj /p:Configuration=Release
```

どこで:

`path_to_gettext` は GETTEXT ライブラリとヘッダーファイルの場所を指定します。

`path_to_openssl` は openssl ライブラリとヘッダーファイルの場所を指定します。

`path_to_pg_installation_dir` は Postgres インストールの場所を指定します。

たとえば、次のコマンドセットは、OpenSSL サポートを PostgreSQL 10 サーバーに組み込みます。

```
SET USE_PGXS=1

SET OPENSSL_PATH=C:/OpenSSL-Win32

SET GETTEXT_PATH="C:/Program Files/PostgreSQL/10/"

SET PG_PATH="C:/Program Files/PostgreSQL/10/"

SET ARCH=x86

msbuild sslutils.proj /p:Configuration=Release
```

ビルドが完了すると、`sslutils` ディレクトリに次のファイルが含まれます。

- `sslutils--1.1.sql`
- `sslutils--unpackaged--1.1.sql`
- `sslutils--pemagent.sql.in`
- `sslutils.dll`

コンパイルされた `sslutils` ファイルをインストールに適したディレクトリにコピーします。例えば:

```
COPY sslutils*.sql /"%PG_PATH%/share/extension/"

COPY sslutils.dll /"%PG_PATH%/lib/"
```

ステップ 3-サービスの停止

古いバックアップデータベースと新しいバックアップデータベースの両方のサービスを停止します。

-RHEL または CentOS 6.x で、コマンドラインを開き、スーパーユーザーの ID を引き継ぎます。次のコマンドを入力します。

```
/etc/init.d/<service_name> stop
```

-RHEL または CentOS 7.x で、コマンドラインを開き、スーパーユーザーの ID を引き継ぎます。次のコマンドを入力します。

```
systemctl/<service_name> stop
```

ここで、`service_name` は Postgres サービスの名前を指定します。

Windows では、Services ダイアログを使用してサービスを制御できます。Services ダイアログを開くには、Control Panel から System and Security メニューに移動します。Administrative Tools を選択し、Services アイコンをダブルクリックします。Services ダイアログが開いたら、リスト内のサービス名を強調表示し、ダイアログに表示されるオプションを使用してサービスを停止します。

ステップ 4-`pg_upgrade` を使用してサーバーを更新

`pg_upgrade` ユーティリティを使用して、古いバックアップデータベースと新しいバックアップデータベース間で既存のデータのインプレース転送を実行できます。サーバーが md5 認証を実施するように設定されている場合、データベーススーパーユーザーの接続プロパティ（およびパスワード）を指定する `.pgpass` ファイルにエントリを追加するか、`pg_hba.conf` `pg_upgrade` を呼び出す前に信頼接続を許可するファイル。 `.pgpass` ファイルにエントリを作成する方法の詳細については、次の場所にある PostgreSQL コードドキュメントを参照してください。

<http://www.postgresql.org/docs/current/static/libpq-pgpass.html>

アップグレードプロセス中に、`pg_upgrade` は一連のログファイルを書き込みます。クラスターの所有者は、書き込み権限があるディレクトリから `pg_upgrade` を呼び出す必要があります。アップグレードが正常に完了すると、`pg_upgrade` はアップグレードの完了時にログファイルを削除します。アップグレードログファイルを削除しないように `pg_upgrade` を指示するには、`pg_upgrade` を呼び出すときに `--retain` キーワードを含めます。

`pg_upgrade` を呼び出すには、クラスター所有者の ID を想定し、クラスター所有者が書き込み権限を持つディレクトリに移動して、コマンドを実行します。

```
<path_to_pg_upgrade> pg_upgrade
-d <old_data_dir_path>
-D <new_data_dir_path>
-b <old_bin_dir_path> -B <new_bin_dir_path>
-p <old_port> -P <new_port>
-u <user_name>
```

どこで:

Postgres ディレクトリの下で “`path_to_pg_upgrade`” specifies the location of the `pg_upgrade` utility. By default, `pg_upgrade` is installed in the “`bin`” ディレクトリ。

`old_data_dir_path` は、古いバックアップデータベースのデータディレクトリへの完全なパスを指定します。

`new_data_dir_path` は、新しいバックアップデータベースのデータディレクトリへの完全なパスを指定します。

`old_bin_dir_path` は、古いバックアップデータベースの `bin` ディレクトリへの完全なパスを指定します。

`new_bin_dir_path` は、古いバックアップデータベースの `bin` ディレクトリへの完全なパスを指定します。

`old_port` は、古いサーバーがリスンしているポートを指定します。

`new_port` は、新しいサーバーがリスンするポートを指定します。

`user_name` はクラスター所有者の名前を指定します。

たとえば、次のコマンド:

```
C:/> "C:/Program Files/PostgreSQL/10/bin/pg_upgrade.exe/"  
-d "C:/Program Files/PostgreSQL/9.6/data/"  
-D "C:/Program Files/PostgreSQL/10/data/"  
-b "C:/Program Files/PostgreSQL/9.6/bin/"  
-B "C:/Program Files/PostgreSQL/10/bin/"  
-p 5432 -P 5433  
-u postgres
```

Windows システムで PEM データベースを PostgreSQL 9.6 から PostgreSQL 10 に移行するように `pg_upgrade` に指示します (バックアップデータベースがデフォルトの場所にインストールされている場合)。

呼び出されると、`pg_upgrade` はデータを新しいバックアップデータベースに移動する前に一貫性チェックを実行します。アップグレードが終了すると、`pg_upgrade` はアップグレードが完了したことを通知します。

`pg_upgrade` オプションの使用またはアップグレードプロセスのトラブルシューティングの詳細については、以下を参照してください。

<http://www.postgresql.org/docs/current/static/pgupgrade.html>

ステップ 5-証明書ファイルを古いデータベースから新しいデータベースにコピーします

次の証明書ファイルを古いバックアップデータベースの `data` ディレクトリから新しいバックアップデータベースの `data` ディレクトリにコピーします。

- `ca_certificate.crt`
- `ca_key.key`
- `root.crt`
- `root.crl`
- `server.key`
- `server.crt`

ターゲットサーバーに配置したら、ファイルには以下で説明する（プラットフォーム固有の）権限が必要です。

Linux での許可と所有権

ファイル名	所有者	許可
ca_certificate.crt	postgres	-rw ——
ca_key.key	postgres	-rw ——
root.crt	postgres	-rw ——
root.crl	postgres	-rw ——
server.key	postgres	-rw ——
server.crt	postgres	-rw-r--r--

Linux では、証明書ファイルは postgres によって所有されている必要があります。コマンドラインで次のコマンドを使用して、ファイルの所有権を変更できます。

```
chown postgres <file_name>
```

ここで、file_name は証明書ファイルの名前を指定します。

server.crt ファイルはファイルの所有者のみが変更できますが、すべてのユーザーが読み取ることができます。次のコマンドを使用して、server.crt ファイルのファイル許可を設定できます：

```
chmod 644 server.crt
```

他の証明書ファイルは、ファイルの所有者のみが変更または読み取ることができます。次のコマンドを使用して、ファイルのアクセス許可を設定できます。

```
chmod 600 <file_name>
```

ここで、file_name はファイルの名前を指定します。

Windows での権限と所有権

Windows では、ソースホストから移動された証明書ファイルは、ターゲットホストで PEM サーバーとバックアップデータベースのインストールを実行したサービスアカウントが所有している必要があります。Run as Administrator オプション（インストーラーのコンテキストメニューから選択）を使用して PEM サーバーと Postgres インストーラーを呼び出した場合、証明書ファイルの所有者は Administrators になります。

Windows でファイルのアクセス許可を確認および変更するには、ファイル名を右クリックし、Properties を選択します。

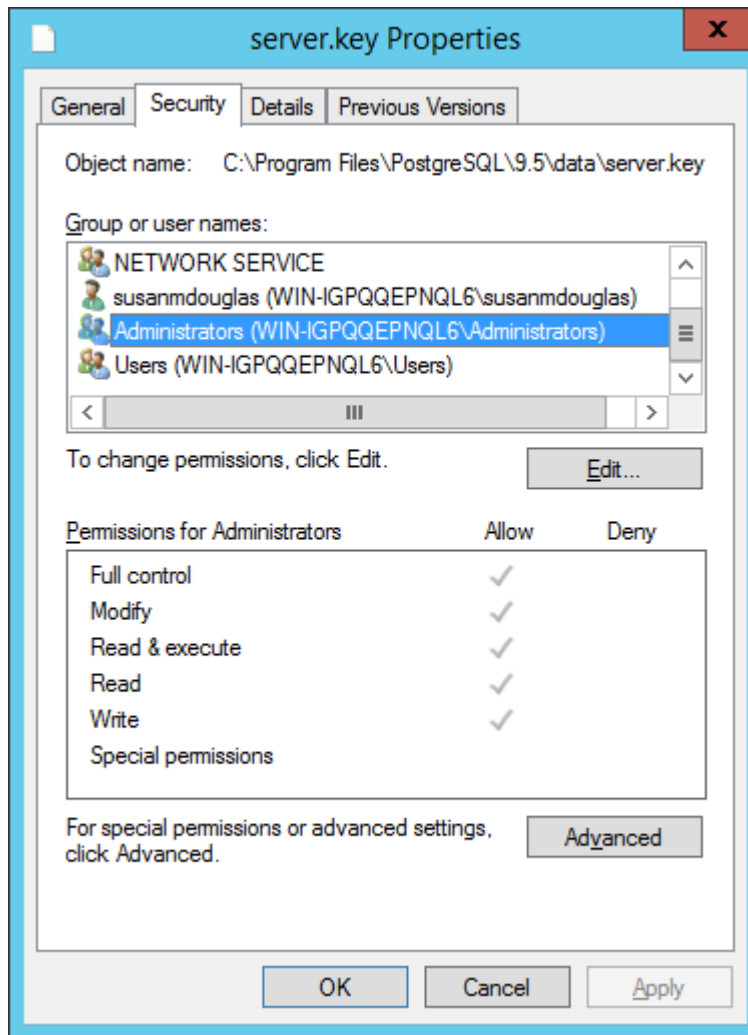


図 1 The Security tab.

Security タブに移動し、Group or user name を強調表示して、割り当てられた権限を表示します。Edit または Advanced を選択して、選択したユーザーに関連付けられた権限を変更できるダイアログにアクセスします。

ステップ 6-新しいサーバー構成ファイルの更新

postgresql.conf ファイルには、サーバーの動作を指定するパラメーター設定が含まれています。古いサーバーの postgresql.conf ファイルで指定された構成と一致するように、新しいサーバーの postgresql.conf ファイルを変更する必要があります。

デフォルトでは、postgresql.conf ファイルがあります:

- Linux で 10 未満の Postgres バージョンの場合、/opt/PostgreSQL/<version.x>/data
- Linux のグラフィカルインストーラーでインストールされた Postgres バージョン 10 以降の場合、/opt/PostgreSQL/<version>/data

- Linux に RPM をインストールした Postgres バージョン 10 以降の場合、`/usr/edb/PostgreSQL/<version>/data`
- Windows 上の Postgres バージョンの場合、`C:/Program Files/PostgreSQL/<version.x>/data`

ここで、`version` はシステム上の Postgres のメジャーバージョンです。

選択したエディターを使用して、新しいサーバーの `postgresql.conf` ファイルを更新します。以下のパラメーターを変更します。

- 元のバックアップデータベースによって監視されるポートでリッスンする `port` パラメーター（通常は 5432）。
- `ssl` パラメーターは `on` に設定する必要があります。

また、次のパラメーターが有効になっていることを確認する必要があります。パラメーターがコメント化されている場合、各 `postgresql.conf` ファイルエントリの前からポンド記号を削除します。

- `ssl_cert_file = 'server.crt' # (change requires restart)`
- `ssl_key_file = 'server.key' # (change requires restart)`
- `ssl_ca_file = 'root.crt' # (change requires restart)`
- `ssl_crl_file = 'root.crl'`

インストールには、新しいバックアップデータベースが古いバックアップデータベースと同等の方法で動作することを保証するために変更が必要な他のパラメータ設定がある場合があります。 `postgresql.conf` ファイルを注意深く確認して、新しいサーバーの構成が古いサーバーの構成と一致することを確認します。

ステップ 7-新しいサーバー認証ファイルの更新

`pg_hba.conf` ファイルには、サーバーがホストベースの認証を強制する方法を指定するパラメーター設定が含まれています。 PEM サーバーをインストールすると、インストーラーは `pg_hba.conf` ファイルを変更し、ファイルの先頭にエントリを追加します。

```
# Adding entries for PEM agents and admins to connect to PEM server

hostssl pem +pem_user 192.168.2.0/24 md5

hostssl pem +pem_agent 192.168.2.0/24 cert

# Adding entries (localhost) for PEM agents and admins to connect to PEM
server

hostssl pem +pem_user 127.0.0.1/32 md5

hostssl postgres +pem_user 127.0.0.1/32 md5

hostssl pem +pem_user 127.0.0.1/32 md5
```

```
hostssl pem +pem_agent 127.0.0.1/32 cert
```

デフォルトでは、pg_hba.conf ファイルは次の場所にあります。

- Linux で 10 未満の Postgres バージョンの場合、 /opt/PostgreSQL/<version>.x/data
- Linux のグラフィカルインストーラーでインストールされた Postgres バージョン 10 以降の場合、 /Opt/PostgreSQL/<version>/data
- Linux に RPM をインストールした Postgres バージョン 10 以降の場合、 /var/lib/PostgreSQL/<version>/data
- Linux 上に RPM をインストールした Advanced Server バージョン 10 以上の場合、 /var/lib/edb/AS<version>/data
- Windows 上の Postgres バージョンの場合、 C:/Program Files/PostgreSQL/version.x/data

ここで、version はシステム上の Postgres のメジャーバージョンであり、*x* はマイナーバージョンです。

任意のエディターを使用して、古いサーバーの pg_hba.conf ファイルから新しいサーバーの pg_hba.conf ファイルにエントリをコピーします。

ステップ 8-新しい Postgres サーバーを再起動します

新しいバックアップデータベースのサービスを開始します。

-RHEL または CentOS 6.x で、コマンドラインを開き、スーパーユーザーの ID を引き継ぎます。次のコマンドを入力します。

```
/etc/init.d/<service_name> start
```

-RHEL または CentOS 7.x で、コマンドラインを開き、スーパーユーザーの ID を引き継ぎます。次のコマンドを入力します。

```
systemctl stop <service_name>
```

ここで、service_name はバックアップデータベースサーバーの名前です。

Windows を使用している場合、Services ダイアログを使用してサービスを制御できます。Services ダイアログを開くには、Control Panel から System and Security メニューに移動します。Administrative Tools を選択し、Services アイコンをダブルクリックします。Services ダイアログが開いたら、リスト内のサービス名を強調表示し、ダイアログで提供されるオプションを使用してサービスを開始します。

第 3

Postgres Enterprise Manager™ サーバーの移動

このセクションの手順では、**PEM** サーバーをあるホストマシンから新しいホストマシンに移動する方法について説明します。新しいホスト（ターゲット）の **PEM** サーバーは、元のホスト（ソース）と同じバージョンの **PEM** サーバーインストーラーでインストールする必要があります。同じインストーラーバージョンを使用しない場合、スキーマ不一致エラーが発生する可能性があることに注意してください。

ターゲットサーバー（PostgreSQL または Advanced Server）のバックアップデータベースは、ソース **PEM** サーバーのバックアップデータベースと同じタイプとバージョン、または異なるタイプとバージョンである場合があります。PostgreSQL ホストにある **PEM** サーバーは、Advanced Server ホストに移行できます。

サーバーの移行を開始する前に、ソースホスト、ターゲットホスト、および **PEM** エージェントのホスト間のファイアウォールがサービス間の接続を許可することを確認する必要があります。

ステップ 1-ターゲットホストの準備

ターゲットホストで **PEM** サーバーのインストーラーを起動します。ソース **PEM** サーバーのインストール時に使用したものと同一バージョンの **PEM** サーバーインストーラーを使用する必要があることに注意してください。

ターゲットサーバーのバックアップデータベースは、ソースのバックアップデータベースとは異なるバージョンまたはタイプである場合があります。新しい **PEM** サーバーが元のサーバーと同じ種類のバックアップデータベースに*存在しない*場合、新しいサーバーホストに同じバージョンの `sslutils` 拡張機能がインストールされていることを確認する必要があります。**PEM** インストーラーと共に配布される `sslutils` のバージョンは、次の EnterpriseDB Web サイトから無料でダウンロードできます。

<http://www.enterprisedb.com/downloads/component-source-code>

PEM サーバーまたは `sslutils` 拡張機能のインストールについては、次の場所にある* PEM インストールガイド*を参照してください。

<https://www.enterprisedb.com/edb-docs/p/edb-postgres-enterprise-manager>

ステップ 2-新しい PEM サーバーから既存のスキーマを削除

移行プロセスにより、ターゲット PEM サーバー上のソース PEM サーバーから `pem`、`pemdata`、および `pemhistory` スキーマが再作成されます。移動の準備として、`psql` クライアントを使用して、ターゲットホスト上の `pem` データベースからこれらのスキーマを削除します。コマンドラインで `psql` クライアントを開くか、Postgres Enterprise Manager メニューから SQL Shell (`psql`) を選択します。

`psql` クライアントが開いたら、データベーススーパーユーザーとして `pem` バックアップデータベースに接続します。ターゲットホスト上の `pem` データベースに接続した後、次のコマンドを使用してスキーマを削除します。

- `DROP SCHEMA pem CASCADE;`
- `DROP SCHEMA pemdata CASCADE;`
- `DROP SCHEMA pemhistory CASCADE;`

スキーマを削除するとき、すべての依存オブジェクトを削除するようにサーバーに指示する `CASCADE` キーワードを含める必要があります。コマンドを実行すると、`psql` クライアントは依存オブジェクトのリストを表示します; クライアントは、`“DROP SCHEMA”` を表示することにより、各スキーマが削除されることを確認します。

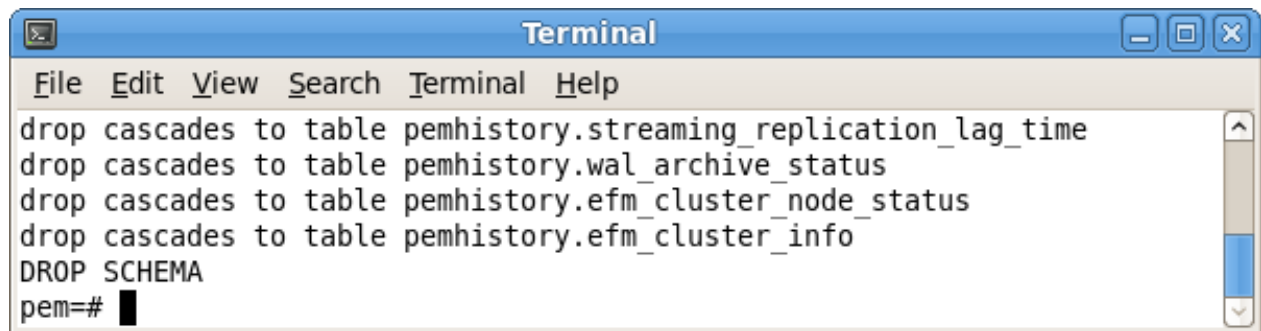


図 1 Dropping the pem schema

ステップ 3-新しい PEM サーバーで PEM エージェントを準備する

PEM サーバーを移動する前に、ソース PEM サーバーによって監視されているエージェントの数を特定し、ターゲットサーバーでその数のエージェント (1 つ未満) の ID を作成する必要があります。PEM サーバーによって監視されている PEM エージェントの総数を検出するには、`psql` クライアントを使用してソースホスト上の `pem` データベースに接続し、`pem.agent` テーブルを照会します。

```
SELECT id FROM pem.agent WHERE active = true;
```

元の PEM サーバーに存在するエージェントの数を 1 つ少なくして手動で作成する必要があります。PEM サーバーインストーラーは、ターゲットホスト上に既に 1 つのエージェントを作成しています。たとえば、ソースサー

バーに 3 つのエージェントが含まれている場合、2 つの追加エージェントを手動で作成する必要があります。ターゲットサーバー上の pem データベースで psql セッションを開き、必要なエージェントを作成します。次のコマンドを使用します。

```
CREATE USER agent <x>;
```

ここで、<x> はエージェント番号を指定します。PEM サーバーのインストーラーによってターゲットホスト上に agent1 が作成されることを忘れないでください。

次に、GRANT コマンドを使用して、ターゲット PEM サーバーに存在する各エージェントに pem_agent 権限を割り当てます。

```
GRANT pem_agent TO agent <x>;
```

ここで、<x> はエージェント番号を指定します。

ステップ 4-ソース PEM サーバーのバックアップスクリプトを生成

pg_dump ユーティリティを使用して、ターゲットホストで pem データベースを再作成するために必要なコマンドを含むスクリプトを生成できます。デフォルトでは、Postgres インストールの bin ディレクトリに pg_dump がインストールされます。pg_dump を呼び出すには、コマンドラインを開き、bin ディレクトリに移動して、次のように入力します。

```
pg_dump -U <user_name> <db_name> > <file_name>
```

どこで:

<user_name> は、PEM バッキングデータベースのデータベーススーパーユーザーの名前を指定します。

<db_name> は PEM バッキングデータベースの名前を指定します。

<file_name> は pg_dump によって生成されたスクリプトの名前を指定します。

プロンプトが表示されたら、指定したユーザーに関連付けられたパスワードを入力します。

表示されるコマンドは、pg_dump に指示して、(実行時に) pem データベースを再作成するスクリプトを生成します。スクリプトの名前は backup.sql で、tmp ディレクトリに作成されます。pg_dump は、ユーザーの資格情報 postgres を使用してサーバーに接続しています。

pg_dump ユーティリティを呼び出しても、現在のデータベースユーザーは中断されません。

ステップ 5-バックアップをターゲットホストに移動

pg_dump ユーティリティによって生成されたスクリプトを PEM サーバーのターゲットホストに移動します。

ステップ 6-ターゲットホストでバックアップを復元します

ターゲットホスト上でコマンドラインを開き、bin ディレクトリ (Postgres バッキングデータベースのインストールディレクトリの下) に移動します。psql を起動し、pg_dump ユーティリティによって生成されたスクリプトを実行します:

```
psql -U <user_name> -d pem -f <file_name>
```

どこで:

<user_name> はデータベースのスーパーユーザーの名前を指定します。指定されたユーザーは、バックアップデータベースの接続特権を持っている必要があります。

<file_name> は、pg_dump によって生成されたバックアップスクリプトへの完全なパスを指定します。

プロンプトが表示されたら、データベースのスーパーユーザーに関連付けられたパスワードを入力します。

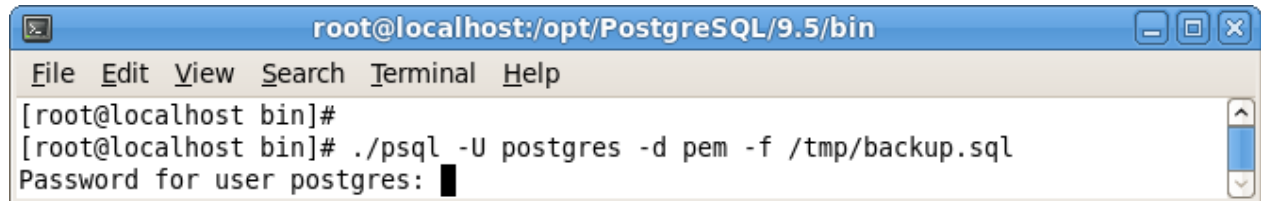


図2 バックアップスクリプトの復元

この例では、psql クライアントを使用して backup.sql というスクリプトを呼び出し、pem データベースを再作成します。スクリプトは、データベーススーパーユーザー postgres に関連付けられた特権を使用して呼び出されます。

ステップ 7-ターゲットホストでデータベースサーバーを停止します

Linux で PEM サーバーを停止するには、次のコマンドを使用します。

```
/etc/init.d/<service_name> stop
```

<service_name> は、バックアップデータベースサーバーの名前を指定します。PostgreSQL バックアップデータベースの場合、サービス名は postgresql-x.x であり、Advanced Server バックアップデータベースの場合、サービス名は ppas-x.x です。ここで、x.x はバージョン番号を指定します。

Windows を使用している場合、Services ダイアログを使用してサービスを制御できます。Services ダイアログを開くには、Control Panel から System and Security メニューに移動します。Administrative Tools を選択し、Services アイコンをダブルクリックします。Services ダイアログが開いたら、リスト内のサービス名を強調表示し、ダイアログに表示されるオプションを使用してサービスを停止します。

ステップ 8-証明書ファイルをターゲットホストにコピー

ターゲットホストのインストール時に作成される証明書ファイルを、ソースホストの証明書ファイルに置き換える必要があります。次のファイルをソース PEM サーバーからターゲット PEM サーバーにコピーします。

- ca_certificate.crt
- ca_key.key
- root.crt

- root.crl
- server.key
- server.crt

ターゲットクラスターのバックアップデータベースを提供する Postgres インストールの下の data ディレクトリにファイルをコピーします。Linux では、デフォルトで、ファイルは次の場所にあります。

```
/opt/PostgreSQL/<x.x>/data/
```

Windows では、ファイルは次の場所にあります。

```
C:/Program Files/PostgreSQL/<x.x>/data
```

どこで:

<x.x> はシステム上の PostgreSQL のバージョンを指定します。

ファイルはターゲットクラスターに既に存在します。コピーを実行する前に既存のファイルを削除するか、既存のファイルをソースサーバーのファイルで上書きします。ターゲットサーバーに配置したら、ファイルには、以降のセクションで説明する（プラットフォーム固有の）権限が必要です。

Linux での許可と所有権

ファイル名	所有者	許可
ca_certificate.crt	ポストGRES	-rw ——
ca_key.key	ポストGRES	-rw ——
root.crt	ポストGRES	-rw ——
root.crl	ポストGRES	-rw ——
server.key	ポストGRES	-rw ——
server.crt	ポストGRES	-rw-r—r—

Linux では、証明書ファイルは postgres が所有している必要があります。コマンドラインで次のコマンドを使用して、ファイルの所有権を変更できます。

```
chown postgres <file_name>
```

ここで、file_name は証明書ファイルの名前を指定します。

server.crt ファイルは、ファイルの所有者のみが変更できますが、すべてのユーザーが読み取ることができます。次のコマンドを使用して、server.crt ファイルのファイル許可を設定できます。

```
chmod 644 server.crt
```

他の証明書ファイルは、ファイルの所有者のみが変更または読み取ることができます。次のコマンドを使用して、ファイルのアクセス許可を設定できます。

```
chmod 600 <file_name>
```

ここで、file_name はファイルの名前を指定します。

Windows での権限と所有権

Windows では、ソースホストから移動された証明書ファイルは、ターゲットホストで PEM サーバーとバックギングデータベースのインストールを実行したサービスアカウントが所有している必要があります。Run as Administrator オプション（インストーラーのコンテキストメニューから選択）を使用して PEM サーバーと Postgres インストーラーを呼び出した場合、証明書ファイルの所有者は Administrators になります。

Windows でファイルのアクセス許可を確認および変更するには、ファイル名を右クリックし、Properties を選択します。

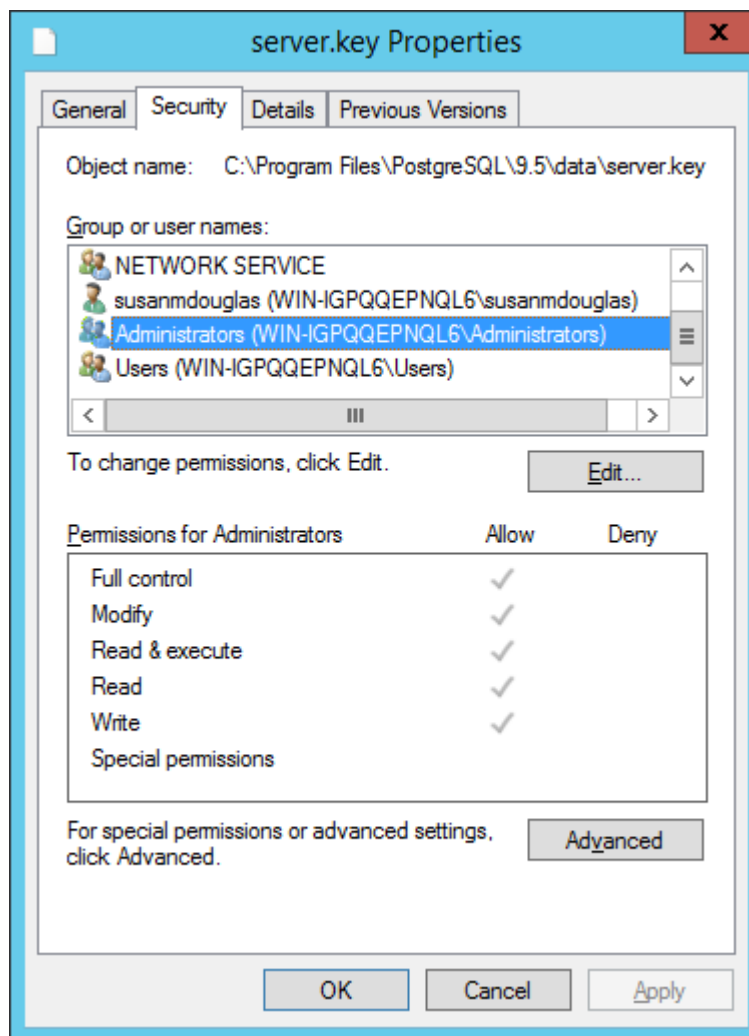


図 3 The Permissions tab

Security タブに移動し、Group or user name を強調表示して、割り当てられた権限を表示します。Edit または Advanced を選択して、選択したユーザーに関連付けられた権限を変更できるダイアログにアクセスし

ます。

ステップ 9-PEM エージェント証明書ファイルを PEM サーバーホストに移動

ソース PEM サーバーの PEM エージェントが使用する証明書ファイルをターゲットホストに移動する必要があります。このステップはプラットフォーム固有です。

Linux で

agent1.key および agent1.crt ファイルをソースホストからターゲットホストにコピーします。デフォルトでは、Linux では、ファイルは /root/.pem にインストールされます。ファイルをターゲットホスト上の同じディレクトリにコピーします。

ファイルのファイル所有権とアクセス許可は、次のように設定する必要があります。

ファイル名	所有者	許可
agent1.key	根	-rw ——
agent1.crt	根	-rw-r-r-

必要に応じて、/root/.pem に移動し、次のコマンドを使用して agent1.key ファイルのアクセス権と所有権を変更します。

```
chmod 600 agent1.key
chown root agent1.key
```

次のコマンドを使用して、agent1.crt ファイルのアクセス権と所有権を変更します。

```
chmod 644 agent1.crt
chown root agent1.crt
```

Windows で

agent1.key および agent1.crt ファイルをソースホストからターゲットホストにコピーします。Windows では、ファイルは次の場所にあります。

```
C:/Users/<user_name>/AppData/Roaming/pem
```

Where *user_name* is the name of the user that invoked the PEM installer.

ターゲットマシン上の証明書ファイルに関連付けられた所有権と許可は、ソースマシン上の証明書ファイルの所有権と許可と一致する必要があります。Run as Administrator オプション（インストーラーのコンテキストメニューから選択）を使用して PEM サーバーと Postgres インストーラーを呼び出した場合、エージェント証明書ファイルの所有者は Administrators になります。

Windows でファイルのアクセス許可を確認および変更するには、ファイル名を右クリックし、Properties を選択します。Security タブに移動し、Group or user name をハイライトして割り当てられた権限を表示

します。Edit または Advanced を選択して、選択したユーザーに関連付けられた権限を変更できるダイアログにアクセスします。

ステップ 10-ターゲットホスト上の “pg_hba.conf” ファイルを更新します

ターゲットホスト上の pg_hba.conf ファイルを変更して、各 PEM エージェントからの接続を許可します。デフォルトでは、pg_hba.conf ファイルは Postgres インストールの下のデータディレクトリにあります。

ステップ 11-ターゲットホストでサーバーを起動

pg_hba.conf ファイルを変更した後、変更を有効にするにはサーバーを再起動する必要があります。

Linux でデータベースサーバーを再起動するには、次のコマンドを使用します。

```
/etc/init.d/<service_name> start
```

ここで、service_name はバックアップデータベースサーバーの名前です。

Windows を使用している場合、Services ダイアログを使用してサービスを制御できます。Services ダイアログを開くには、Control Panel から System and Security メニューに移動します。Administrative Tools を選択し、Services アイコンをダブルクリックします。Services ダイアログが開いたら、リスト内のサービス名を強調表示し、ダイアログに表示されるオプションを使用してサービスを開始します。

ステップ 12-監視対象エージェントを新しい PEM サーバーホストに接続

既存の PEM エージェントに新しい PEM サーバーホストに接続するように指示するには、以下を行う必要があります。

- PEM エージェントホストが新しい PEM サーバーホストに接続できることを確認します。
- 新しい PEM サーバーの IP アドレスとポートを指定して、レジストリ（PEM エージェントを使用する各 Windows ホスト）またはエージェント構成ファイル（PEM エージェントを使用する各 Linux ホスト）を変更します。
- PEM エージェントのサービスを再起動します。

これらの手順はプラットフォーム固有です。

PEM エージェントが Linux に存在する場合

エディターの選択を使用して、agent.cfg ファイルを変更し、pem_host および pem_port パラメーターで PEM サーバーの新しい IP アドレスとポート番号を指定します。

デフォルトでは、agent.cfg ファイルは次の場所にあります。

```
/opt/PEM/agent/etc/agent.cfg
```

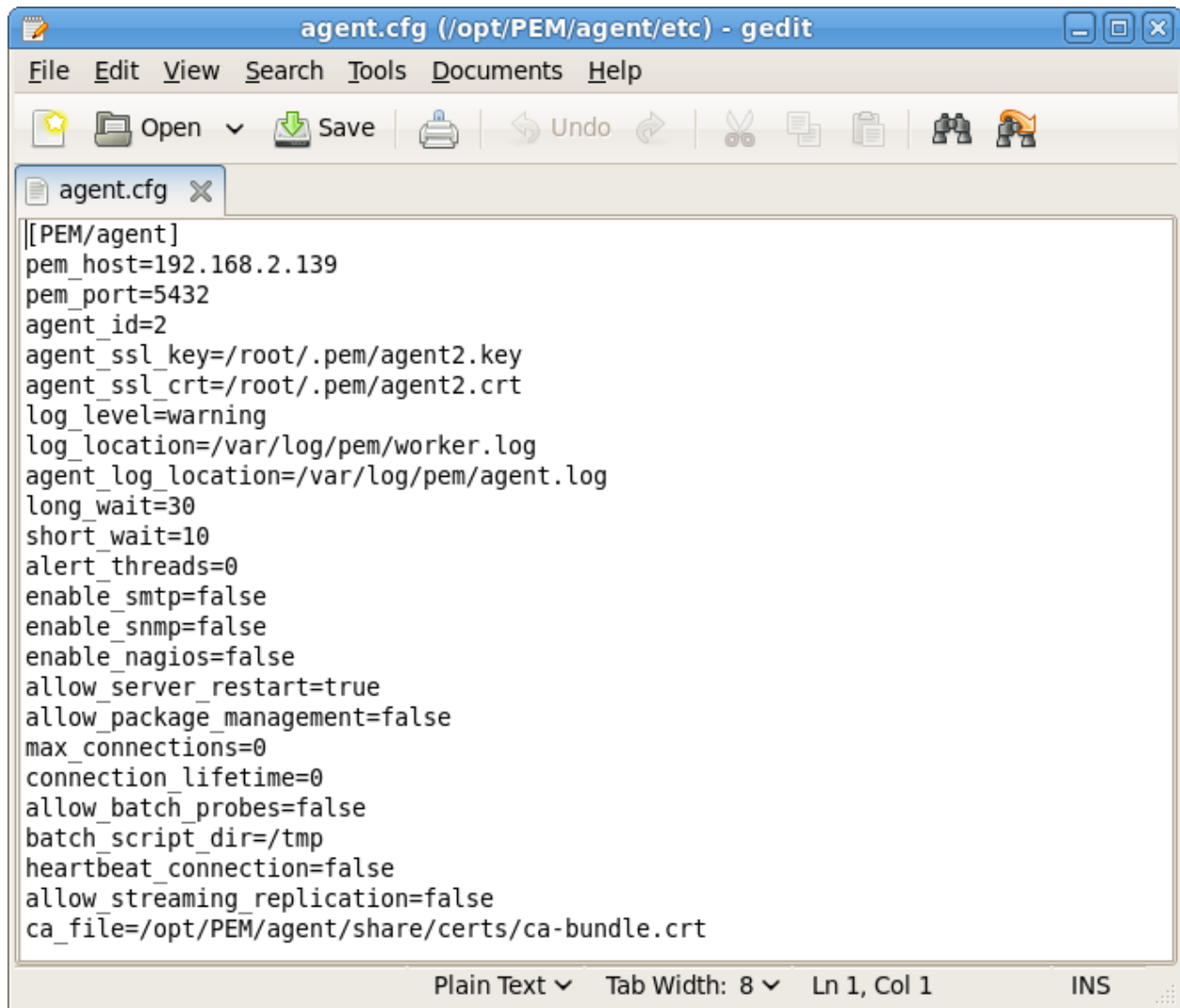


図 4 The agent.cfg file

agent.cfg ファイルを変更した後、PEM エージェントサービスを再起動する必要があります。Linux コマンドラインで pemagent サービススクリプトを使用して、サービスを再起動できます。

```
/etc/init.d/pemagent restart
```

PEM エージェントが Windows に存在する場合

Registry Editor。PEM エージェントホストが PEM サーバーホストに接続できることを確認したら、Windows の Registry Editor を使用して PEM_HOST および PEM_PORT のエントリを確認および編集し、ホストが正しく識別されることを確認できますおよび PEM サーバーが使用するポート。Registry Editor を開くには、Windows の Run ダイアログまたは Windows のスタートメニューの検索ボックスに regedit と入力します。

レジストリツリーコントロール内を移動して、レジストリエントリを表示または変更します。64 ビット Windows

では、PEM エージェントのレジストリエントリは次の場所にあります。

```
HKEY_LOCAL_MACHINE SOFTWARE wow6432Mode EnterpriseDB PEM agent
```

32 ビット Windows では、PEM エージェントのレジストリエントリは次の場所にあります。

```
HKEY_LOCAL_MACHINE SOFTWARE EnterpriseDB PEM agent
```

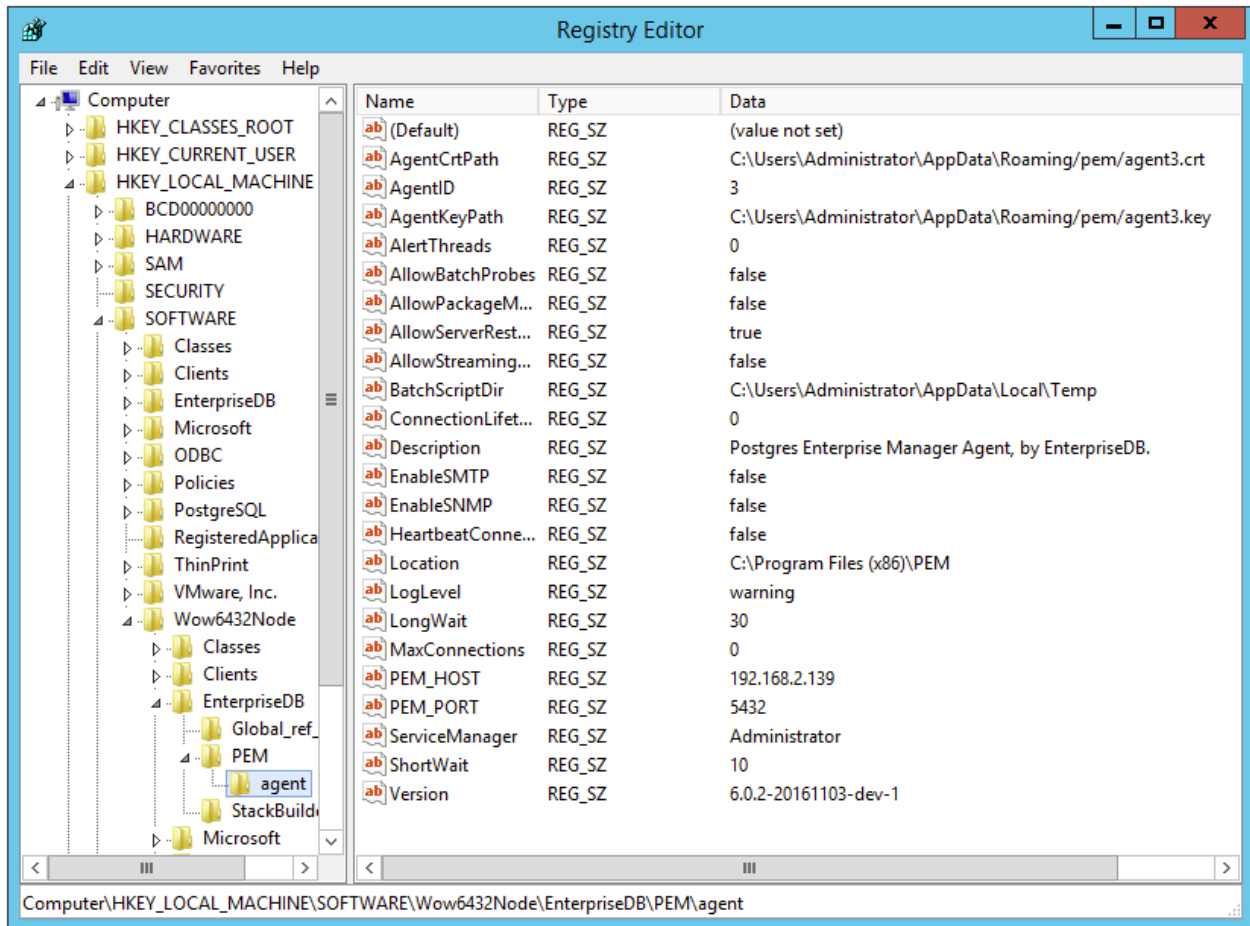


図 5 The Windows Registry Editor

PEM_HOST および PEM_PORT エントリは、ターゲットホスト上の新しい PEM サーバーのアドレスとポート番号を指定する必要があります。レジストリエントリを変更するには、エントリ Name を右クリックし、コンテキストメニューから Modify を選択して Edit String ダイアログを開きます。

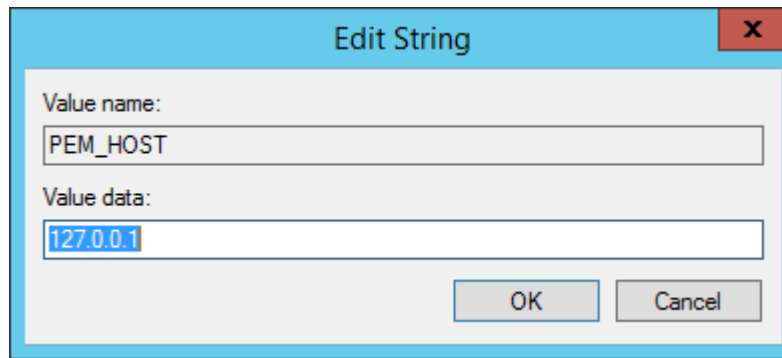


図 6 The Windows Registry Editor

エントリの値を変更するには、Edit String ダイアログを使用します。終了したら、OK をクリックして変更を保存するか、Cancel をクリックして保存せずに終了します。

レジストリを変更した後、PEM エージェントのサービスを再起動する必要があります。Services ダイアログ (Windows の Control Panel からアクセス) を使用して、Postgres Enterprise Manager - pemAgent サービスを再起動できます。

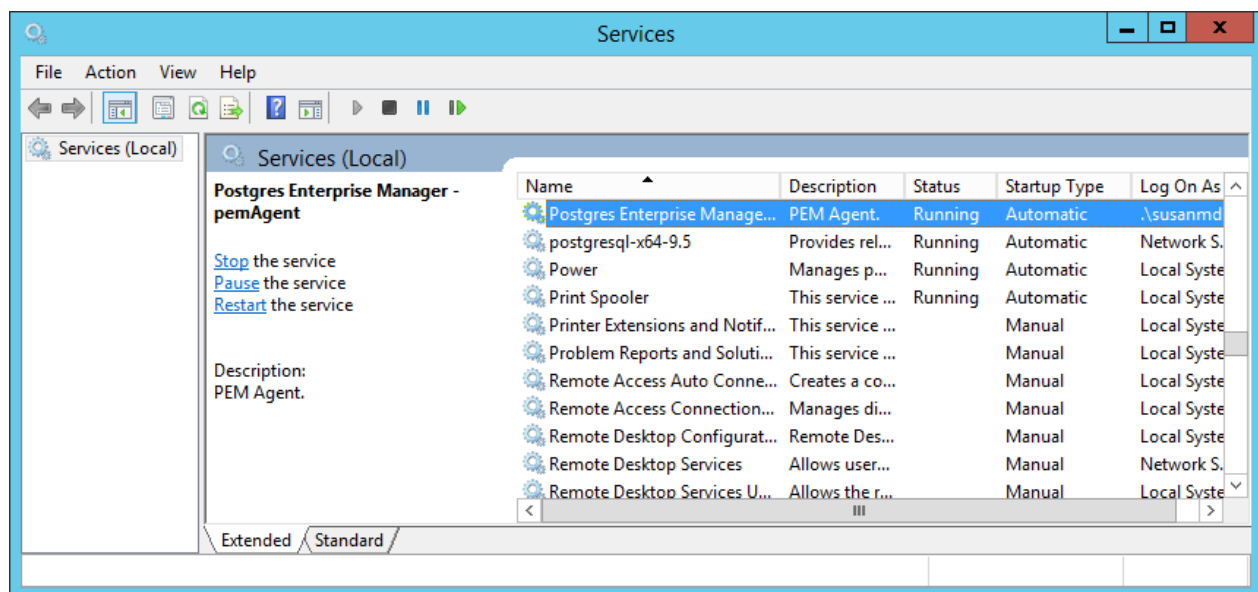


図 7 Restarting the PEM Agent's service

サーバーを移動した後、インストールされている PEM クライアントの接続プロパティを変更して、PEM サーバー、エージェント、監視対象サーバーの新しいホストに接続します。

トラブルシューティング

4.1 pem.alert テーブルが復元に失敗する

バックアップから pem バックアップデータベースを復元すると、pem.alert テーブルの復元中にエラーが発生する場合があります。これは、テーブルのテーブルの前提条件が欠落しているために発生します。pg_restore ユーティリティは、pem.alert を復元しようとした後、pem.alert の前提条件を復元できます。

これが発生した場合、pg_restore からの出力には、alert テーブルを参照するエラーメッセージが含まれます:

```
pg_restore: [archiver (db)] could not execute query: ERROR: insert or
update on table "alert_history" violates foreign key constraint
"alert_history_alert_id_fkey"
DETAIL: Key (alert_id)=(3) is not present in table "alert".
Command was: ALTER TABLE ONLY alert_history
ADD CONSTRAINT alert_history_alert_id_fkey FOREIGN KEY (alert_id)
REFERENCES alert(id) ON...
pg_restore: creating FK CONSTRAINT alert_status_alert_id_fkey
pg_restore: [archiver (db)] Error from TOC entry 3265; 2606 18355 FK
CONSTRAINT alert_status_alert_id_fkey postgres
pg_restore: [archiver (db)] could not execute query: ERROR: insert or
update on table "alert_status" violates foreign key constraint
"alert_status_alert_id_fkey"
DETAIL: Key (alert_id)=(1) is not present in table "alert".
Command was: ALTER TABLE ONLY alert_status
ADD CONSTRAINT alert_status_alert_id_fkey FOREIGN KEY (alert_id)
REFERENCES alert(id) ON U...
```

この問題が発生した場合は、`pem.alert` テーブルを復元する前に `pem` データベースを復元してください。 `pem` データベースを復元すると、`pem.alert` の前提条件がインストールされ、テーブルの復元が期待どおりに完了するはずです。

Postgres Enterprise Manager™ のアンインストール

PEM サーバーまたはエージェントをアンインストールするプロセスはプラットフォーム固有です。PEM サーバーのパッケージ名は `edb-pem-server` で、PEM エージェントのパッケージ名は `edb-pem-agent` です。

ホストから PEM サーバーパッケージをアンインストールする場合、同じホストにインストールされている PEM エージェントパッケージはアンインストールされません。ただし、PEM エージェントパッケージをアンインストールすると、同じホストにインストールされている PEM サーバーパッケージもアンインストールされます。

5.1 Windows ホストからの PEM のアンインストール

PEM インストールが Windows ホストにある場合、Windows の Add/Remove Programs アプリケーションを使用して PEM コンポーネントを削除できます。Windows の Control Panel から Add/Remove Programs オプションを選択します。control panel が開いたら、プログラムリストで PEM コンポーネントの名前を見つけてください。コンポーネントを削除するには、Remove ボタンをクリックします。

次の場所にあるアンインストーラーを呼び出すこともできます。

```
C:/Program Files/edb/pem/server
```

5.2 CentOS または RHEL ホストからの PEM のアンインストール

`rpm`、`yum remove`、または `yum erase` コマンドのバリエーションを使用して、インストールされたパッケージを削除できます。パッケージを削除しても、PEM データディレクトリが破損することはありません。

- `rpm` コマンドを呼び出してインストール済みパッケージを削除する場合は、`-e` オプションを含めます。コマンド構文は次のとおりです。

```
rpm -e <package_name>
```

- `yum remove` コマンドを使用して、`yum` によってインストールされた `pem` サーバーまたはエージェントパッケージを削除できます。パッケージを削除するには、ターミナルウィンドウを開き、スーパーユーザー権限を引き受けて、次のコマンドを入力します。

```
yum remove <package_name>
```

- `yum erase` コマンドを使用して、`edb-pem` および `edb-pem-docs` の依存関係とともに `pem` サーバーまたはエージェントパッケージを削除できます。パッケージを削除するには、ターミナルウィンドウを開き、スーパーユーザー権限を引き受けて、次のコマンドを入力します。

```
yum erase <package_name>
```

Where *package_name* is the name of the package that you would like to remove.

5.3 Debian または Ubuntu ホストからの PEM のアンインストール

`apt-get remove` または `apt-get purge` コマンドを使用して、Debian または Ubuntu ホストから PEM サーバーまたはエージェントパッケージをアンインストールできます。

- 構成ファイルとデータディレクトリに影響を与えずに Debian または Ubuntu ホストから PEM サーバーまたはエージェントをアンインストールするには、次のコマンドを呼び出します。

```
apt-get remove <package_name>
```

- 構成ファイルとデータディレクトリとともに PEM サーバーまたはエージェントをアンインストールするには、次のコマンドを呼び出します。

```
apt-get purge <package_name>
```

Where *package_name* is the name of the package that you would like to remove.

5.4 SLES ホストからの PEM のアンインストール

SLES ホストから PEM サーバーまたはエージェントをアンインストールするには、次のコマンドを呼び出します。

```
zypper remove <package_name>
```

Where *package_name* is the name of the package that you would like to remove.

結論

EDB Postgres Enterprise Manager アップグレードおよび移行ガイド Copyright©2007-2019 EnterpriseDB Corporation。無断複写・転載を禁じます。

EnterpriseDB®Corporation 34 Crosby Drive, Suite 201, Bedford, MA 01730, USA

T +1 781 357 3390 F +1 978 467 1307 E info@enterprisedb.com www.enterprisedb.com

- EDB は、EDB Postgres Enterprise Manager が存在する場合、ログオン UI の設計、コーディングのベストプラクティスの確立、レビュー、および入力検証の検証を行います。EDB は追加の入力コンポーネントに対して同じアプローチに従いますが、製品の性質上、限定された検証が可能な信頼できるユーザーによって入力および送信されるフリーフォーム SQL、WMI またはその他の文字列を受け入れる必要がある場合があります。このような場合、ユーザーが誤った入力やその他の危険な入力を防ぐことはできません。
- EDB は、将来、認証された信頼できるユーザーからのフリーフォーム SQL、WMI またはその他の潜在的に危険な入力を受け入れる製品に機能を追加する権利を留保しますが、そのような機能はすべて、可能な限り最小のリスクを提供するように設計およびテストされます可能であれば、スーパーユーザーまたは同等の特権が必要です。
- EDB は、すべての潜在的な脅威を予測できることを保証しません。したがって、すべての潜在的な脆弱性が対処または検討されたことをプロセスが完全に保証することはできません。

Backing Postgres データベースのアップグレード, 33

Linux ホスト上のグラフィカルインストーラーでインストールされた PEM のアップグレード, 25

PEM インストールのアップグレード, 2

Postgres Enterprise Manager™ サーバーの移動, 43

Postgres Enterprise Manager™ のアンインストール, 56

RPM パッケージと共に Linux ホストにインストールされた PEM のアップグレード, 20

Upgrading PEM agent installed with graphical installer, 29

Upgrading PEM server installed with graphical installer, 25

Windows ホストでの PEM インストールのアップグレード, 2

トラブルシューティング, 54

隔離されたネットワークでの PEM リポジトリの作成, 24

結論, 58