



EDB Postgres Advanced Server

リリース **13**

Advanced Server Upgrade Guide

2021 年 01 月 03 日

目次

第 1 章	はじめに	1
第 2 章	Supported Platforms	3
第 3 章	Limitations	5
第 4 章	Upgrading an Installation With pg_upgrade	7
4.1	Performing an Upgrade	9
4.1.1	Linking versus Copying	10
4.2	Invoking pg_upgrade	11
4.2.1	Command Line Options - Reference	13
4.3	Upgrading to Advanced Server 13	17
4.4	Upgrading a pgAgent Installation	26
4.5	pg_upgrade Troubleshooting	27
4.5.1	Upgrade Error - There seems to be a postmaster servicing the cluster	27
4.5.2	Upgrade Error - fe_sendauth:no password supplied	27
4.5.3	Upgrade Error - New cluster is not empty; exiting	27
4.5.4	Upgrade Error - Failed to load library	27
4.6	Reverting to the Old Cluster	28
第 5 章	Performing a Minor Version Update of an RPM Installation	29
第 6 章	Using StackBuilder Plus to Perform a Minor Version Update	31
第 7 章	Migration to Version 13	37
第 8 章	Conclusion	41
	索引	43

第 1 章

はじめに

EDBPostgresAdvancedServer アップグレードガイドは、EDBPostgresAdvancedServer (AdvancedServer) のアップグレード処理に関する包括的なガイドです。このガイドでは、次の使用に関する詳細情報を確認できます。

- `pg_upgrade` を使用して、以前のバージョンの AdvancedServer から AdvancedServer13 にアップグレードします。
- Linux ホストでマイナーバージョンアップグレードを実行するには、`yum`。
- StackBuilder Plus を使用して、Windows ホストでマイナーバージョンアップグレードを実行します。

第 2 章

Supported Platforms

AdvancedServer でサポートされているプラットフォームとバージョンについては、次の EDB ウェブサイトしてください。

<https://www.enterprisedb.com/services-support/edb-supported-products-and-platforms#epas>

第 3 章

Limitations

EDBPostgresAdvancedServer には次の制限が適用されます。

- 稼動データベースの data ディレクトリは、NFS ファイルシステムに保存しないでください。
- 外部キーがパーティションテーブルを参照している場合、pg_upgrade ユーティリティはパーティションテーブルをアップグレードできません。
- あなたは、バージョン 9.4 サーバーまたは AdvancedServer の下位バージョンからアップグレード処理している、とあなたは SUBPARTITION BY 句が含まれるテーブルパーティションを使用する場合は、AdvancedServer の新しいバージョンに既存の AdvancedServer>をアップグレードする pg_dump と pg_restore を使用する必要があります。をアップグレードには、以下を行う必要があります。
 1. pg_dump を使用して、サブパーティションテーブルのコンテンツを保存します。
 2. AdvancedServer9.4 データベースまたは AdvancedServer データベースの下位バージョンからテーブルを削除します。
 3. pg_upgrade を使用して、AdvancedServer データベースの残りをより新しいバージョンにアップグレードします。
 4. pg_restore を使用して、サブパーティションテーブルを最新のアップグレードされた Advanced-Server データベースに復元します。
- AdvancedServer インストールのアップグレードを実行する場合、アップグレードされたサーバーでハッシュパーティションテーブルを再構築する必要があります。

第 4 章

Upgrading an Installation With pg_upgrade

バージョン間のマイナーアップグレードは非常にシンプルで、新しい実行可能ファイルのインストールのみが必要ですが、過去のメジャーバージョンのアップグレードは高価で時間がかかりました。pg_upgrade は、AdvancedServer の任意のバージョン（バージョン 9.0 以降）と、同じプラットフォームでサポートされている AdvancedServer の以降のリリースとの間の移行を容易にします。

pg_upgrade がない場合、AdvancedServer の以前のバージョンから AdvancedServer13 に移行するには、pg_dump を使用してすべてのデータをエクスポートし、新しいリリースをインストールし、initdb を実行して新しいクラスターを作成し、古いデータをインポートする必要があります。

pg_upgrade は、多くのメジャーバージョンアップに必要な時間とディスクスペースの両方を削減できます。

pg_upgrade ユーティリティは、AdvancedServer と後続のバージョン間で既存のデータのインプレース転送を実行します。

インプレースアップグレードが実用的かどうかは、いくつかの要因によって決まります。

- ユーザー定義テーブルのディスク上の表現形式は、オリジナルバージョンとアップグレードされたバージョンの間で変更してはなりません。
- データ型のディスク上の表現形式は、オリジナルバージョンとアップグレードされたバージョンの間で変更してはなりません。
- pg_upgrade を使用して AdvancedServer のメジャーバージョン間でアップグレードには、両方のバージョンで各データタイプの共通バイナリ表現形式を共有する必要があります。したがって、pg_upgrade を使用して 32 ビットから 64 ビット Linux プラットフォームに移行することはできません。

バージョンアップグレードを実行する前に、pg_upgrade は 2 つのクラスター（古いクラスターと新しいクラスター）の互換性を確認します。

アップグレードがデータベースオブジェクトまたはデータのディスク上の表現形式の変更を含む場合、またはデータ型のバイナリ表現形式の変更を含む場合、pg_upgrade はアップグレードを実行できません。アップグレードするには、古いデータを pg_dump してから、そのデータを新しいクラスターにインポートする必要があります。

pg_upgrade 実行可能ファイルは AdvancedServer13 とともに配布され、Database Server コンポーネントのパートとしてインストールされます。追加のインストールまたは構成手順は必要ありません。

4.1 Performing an Upgrade

AdvancedServer の以前のバージョンを現在のバージョンにアップグレードするには、以下を行う必要があります。

- AdvancedServer の現在のバージョンをインストールします。新しいインストールには、古いインストールと同じサポートサーバーコンポーネントが含まれている必要があります。
- ターゲットデータベースを空にするか、initdb を使用して新しいターゲットクラスターを作成します。
- 両方のデータベースの pg_hba.conf ファイルを trust 認証モード（認証の競合を避けるため）。
- 古い AdvancedServer サービスと新しい AdvancedServer サービスをシャットダウンします。
- pg_upgrade ユーティリティを呼び出します。

pg_upgrade が起動すると、互換性チェックを実行して、必要なすべての実行可能ファイルが存在し、予想されるバージョン番号が含まれている保証を確認します。検証プロセスでは、古い \$PGDATA ディレクトリと新しい \$PGDATA ディレクトリもチェックして、予想されるファイルとサブディレクトリが適切に配置されていることを保証します。検証プロセスが成功すると、pg_upgrade は古い postmaster を起動し、pg_dumpall --schema-only を実行して古いクラスターに含まれるメタデータをキャプチャします。pg_dumpall によって生成されたスクリプトは、新しいクラスター内のすべてのユーザー定義オブジェクトを再作成するために、後の手順で使用されます。

pg_dumpall によって生成されるスクリプトは、ユーザー定義オブジェクトのみを再作成し、システム定義オブジェクトは再作成しないことに注意してください。新しいクラスターには、AdvancedServer の最新バージョンによって作成されたシステム定義オブジェクトが*既に*含まれています。

古いクラスターからメタデータを抽出した後、pg_upgrade は、新しいクラスターを既存のデータと同期するために必要なブックキーピングタスクを実行します。

pg_upgrade は、新しいクラスターに対して pg_dumpall スクリプトを実行して、古いクラスターで見つかったものと同じ形状とタイプの（空の）データベースオブジェクトを作成します。次に、pg_upgrade は、古いクラスターから新しいクラスターに各テーブルとインデックスをリンクまたはコピーします。

9.5 より前の AdvancedServer のバージョンから AdvancedServer13 にアップグレード処理し、edb_dblink_oci または edb_dblink_libpq 拡張機能をインストールした場合、アップグレードを実行する前に拡張機能を削除する必要があります。拡張機能を削除するには、psql または PEM クライアントでサーバーに接続し、コマンドを呼び出します。

```
DROP EXTENSION edb_dblink_oci;
DROP EXTENSION edb_dblink_libpq;
```

アップグレード処理が完了したら、CREATE EXTENSION コマンドを使用して、現在のバージョンの拡張機能をインストールに追加できます。

4.1.1 Linking versus Copying

`pg_upgrade` を呼び出すときに、コマンドラインオプションを使用して、`pg_upgrade` が古いクラスターの各テーブルとインデックスを新しいクラスターに*コピー*または*リンク*するかどうかを指定できます。

`pg_upgrade` はクラスター内の各ファイルに 2 番目の名前（ハードリンク）を作成するだけなので、リンクははるかに高速です。`pg_upgrade` はオリジナルデータのコピーを作成しない `make`、リンクには追加のワークスペースも必要ありません。古いクラスターと新しいクラスターをリンクするとき、古いクラスターと新しいクラスターはデータを共有します。新しいクラスターを開始した後、以前のバージョンの `AdvancedServer` でデータを使用できなくなることにノートしてください。

古いクラスターから新しいクラスターにコピーデータことを選択した場合、`pg_upgrade` は、従来の `dump/restore` プロシージャと比較して、アップグレードの実行に必要な時間を短縮します。`pg_upgrade` は、ファイルごとのメカニズムを使用して、古いクラスターから新しいクラスターにデータファイルをコピーデータます（`dump/restore` で使用される行ごとのメカニズムとは異なります）。`pg_upgrade` を使用すると、新しいクラスターでインデックスを作成することを避けます。各インデックスは、古いクラスターから新しいクラスターに単純にコピーされます。最後に、`dump/restore` プロシージャを使用してアップグレードするには、すべてのデータの間接テキストベースのダンプを保持するために大量のワークスペースが必要ですが、`pg_upgrade` には余分なワークスペースがほとんど必要ありません。

ユーザー定義のテーブルスペースに保存されているデータは、新しいクラスターにコピーされません。ファイルシステム内の同じ場所に残りますが、名前が新しいクラスターのバージョン番号を反映するサブディレクトリにコピーされます。アップグレード処理後にテーブルスペースに保存されているファイルを手動で再配置するには、ファイルを新しい場所に移動し、ファイルを指すようにシンボリックリンク（クラスターの `data` ディレクトリの下に `pg_tblspc` ディレクトリにあります）を更新します。

4.2 Invoking pg_upgrade

pg_upgrade を呼び出す場合、新旧のクラスターの PGDATA および実行可能（ /bin ）ディレクトリーの場所、AdvancedServer スーパーユーザの名前、およびインストールがリッスンしているポートを指定する必要があります。pg_upgrade を呼び出して AdvancedServer12 から AdvancedServer13 に移行する一般的な呼び出しは、次のフォームを取ります。

```
pg_upgrade
--old-datadir <path_to_12_data_directory>
--new-datadir <path_to_13_data_directory>
--user <superuser_name>
--old-bindir <path_to_12_bin_directory>
--new-bindir <path_to_13_bin_directory>
--old-port <12_port> --new-port <13_port>
```

どこで:

```
--old-datadir path_to_12_data_directory
```

--old-datadir オプションを使用して、AdvancedServer12 インストール内の data ディレクトリへの完全なパスを指定します。

```
--new-datadir path_to_13_data_directory
```

--new-datadir オプションを使用して、AdvancedServer13 インストール内の data ディレクトリへの完全なパスを指定します。

```
--username superuser_name
```

--username オプションを含めて、AdvancedServer スーパーユーザの名前を指定します。スーパーユーザ名前は、AdvancedServer の両方のバージョンで同じである必要があります。デフォルトでは、AdvancedServer がオラクルモードでインストールされている場合、スーパーユーザの名前付けは enterprisedb です。PostgreSQL モードでインストールされている場合、スーパーユーザの名前付けは postgres です。

AdvancedServer のスーパーユーザ名前が両方のクラスターで同じでない場合、クラスターは pg_upgrade 整合性チェックに合格しません。

```
--old-bindir path_to_12_bin_directory
```

--old-bindir オプションを使用して、AdvancedServer12 インストールの bin ディレクトリへの完全なパスを指定します。

```
--new-bindir path_to_13_bin_directory
```

--new-bindir オプションを使用して、AdvancedServer13 インストールの bin ディレクトリへの完全なパスを指定します。

```
--old-port 12_port
```

`--old-port` オプションを含めて、AdvancedServer12 が接続をリッスンするポートを指定します。

`--new-port 13_port`

`--new-port` オプションを含めて、AdvancedServer13 が接続をリッスンするポートを指定します。

4.2.1 Command Line Options - Reference

pg_upgrade は、次のコマンドラインオプションを受け入れます。各オプションは、長いフォームまたは短いフォームで使用できます。

```
-b path_to_old_bin_directory  
--old-bindir path_to_old_bin_directory
```

-b または --old-bindir キーワードを使用して、古いクラスターの実行可能ディレクトリの場所を指定します。

```
-B path_to_new_bin_directory  
--new-bindir path_to_new_bin_directory
```

-B または --new-bindir キーワードを使用して、新しいクラスターの実行可能ディレクトリの場所を指定します。

```
-c  
--check
```

-c または --check キーワードを含めて、バージョンアップグレードを実行せずに pg_upgrade が新旧クラスターの整合性チェックを実行するように指定します。

```
-d path_to_old_data_directory  
--old-datadir path_to_old_data_directory
```

-d または --old-datadir キーワードを使用して、古いクラスターの data ディレクトリの場所を指定します。

```
-D path_to_new_data_directory  
--new-datadir path_to_new_data_directory
```

`-D` または `--new-datadir` キーワードを使用して、新しいクラスターの `data` ディレクトリの場所を指定します。

ユーザー定義のテーブルスペースに保存されているデータは、新しいクラスターにコピーされません。ファイルシステム内の同じ場所に残りますが、名前が新しいクラスターのバージョン番号を反映するサブディレクトリにコピーされます。アップグレード処理後にテーブルスペースに保存されているファイルを手動で再配置するには、ファイルを新しい場所に移動し、ファイルを指すようにシンボリックリンク（クラスターの `data` ディレクトリの下に `pg_tblspc` ディレクトリにあります）を更新する必要があります。

`-j`
`--jobs`

`-j` または `--jobs` キーワードを含めて、アップグレード中に使用する同時プロセスまたはスレッドの数を指定します。

`-k`
`--link`

`-k` または `--link` キーワードを含めて、新しいクラスターから古いクラスターへのハードリンクを作成します。シンボリックリンクの使用の詳細については、[リンクとコピー](#) を参照してください。

`-o options`
`--old-options options`

`-o` または `--old-options` キーワードを使用して、古い `postgres` コマンドに渡されるオプションを指定します。彼らはグループとして渡されることを保証するために、単一引用符またはダブルクオートで囲みオプション。

`-O options`
`--new-options options`

`-O` または `--new-options` キーワードを使用して、新しい `postgres` コマンドに渡すオプションを指定します。彼らはグループとして渡されることを保証するために、単一引用符またはダブルクオートで囲みオプション。

```
-p old_port_number  
--old-port old_port_number
```

-p または --old-port キーワードを含めて、アップグレード処理する AdvancedServer インストールのポート番号を指定します。

```
-P new_port_number  
--new-port new_port_number
```

-P または --new-port キーワードを含めて、新しい AdvancedServer インストールのポート番号を指定します。

注釈: AdvancedServer13 インストーラを呼び出すときにオリジナルの AdvancedServer インストールがポート番号 5444 を使用している場合、インストーラは AdvancedServer の新規インストールにリスナポート 5445 の使用を推奨します。

```
-r  
--retain
```

アップグレードプロセス中に、pg_upgrade は 4 つの追加専用ログファイルを作成します。アップグレードが完了すると、pg_upgrade はこれらのファイルを削除します。-r または --retain オプションを含めて、サーバーが pg_upgrade ログファイルを保持するように指定します。

```
-U user_name  
--username user_name
```

-U または --username キーワードを含めて、AdvancedServer データベーススーパーユーザの名前を指定します。同じスーパーユーザが両方のクラスターに存在する必要があります。

```
-v  
--verbose
```

`-v` または `--verbose` キーワードを含めて、アップグレードプロセス中に詳細な出力を有効にします。

`-V`

`--version`

`-V` または `--version` キーワードを使用して、`pg_upgrade` のバージョン情報を表示します。

`-?`

`-h`

`--help`

`-?`, `-h`, または `--help` オプションして、`pg_upgrade` ヘルプ情報を表示します。

4.3 Upgrading to Advanced Server 13

pg_upgrade を使用して、AdvancedServer の既存のインストールから AdvancedServer13 インストーラによって構築されたクラスター、または initdb コマンドを使用して作成された代替クラスターにアップグレードできます。このセクションでは、インストーラによって提供されるクラスターへのアップグレード処理の詳細を提供します。

initdb コマンドで作成された空のクラスターへのアップグレードを実行する基本的な手順は、AdvancedServer13 インストーラで作成されたクラスターへのアップグレード手順と同じですが、手順 2 (Empty the edb database) を省略して、代替の場所を置き換えることができますアップグレードのターゲットクラスターを指定する際のクラスター。

アップグレードプロセス中に問題が発生した場合は、以前のバージョンに戻すことができます。 [古いクラスターへの復帰](#) セクションを参照してください。このプロセスに関する詳細情報。

AdvancedServer のアップグレードを実行するには、オペレーティングシステムのスーパーユーザまたは Windows 管理者である必要があります。

Step 1 - Install the New Server

AdvancedServer13 をインストール。以前の AdvancedServer のインストール中にインストールされたものと同じ非サーバーコンポーネントを指定します。新しいクラスターと古いクラスターは異なるディレクトリに存在する必要があります。

Step 2 - Empty the target database

ターゲットクラスターにデータを含めることはできません。あなたは initdb コマンドを使用して、空のクラスターを作成することができ、またはあなたが PostgreSQL のモードで AdvancedServer をインストールした場合の AdvancedServer13 のインストール時に作成されたデータベースを空にすることができ、インストーラが postgres 名前付けの単一のデータベースを作成します。あなたは、オラクル>で AdvancedServer をインストールした場合、それは postgres とデータベースの名前付け edb 名前付けのデータベースを作成します。

ターゲットデータベースを空にする最も簡単な方法は、データベースを削除してから新しいデータベースを作成することです。DROP DATABASE コマンドを呼び出す前に、すべてのユーザーを切断し、現在データベースを使用しているサービスを停止する必要があります。

Windows、Control Panel から Services マネージャに移動します。Services リストで各サービスを強調表示し、Stop を選択します。

Linux では、ターミナルウィンドウを開き、スーパーユーザ権限を引き受け、各サービスを手動で停止します。例、Linux6.x を使用している場合、次のコマンドを呼び出して pgAgent サービスを停止します。

```
service edb-pgagent-13 stop
```

現在 AdvancedServer に接続されているサービスを停止した後、EDB-PSQL コマンドラインクライアントを使用して、データベースを削除および作成できます。クライアントが開いたら、データベーススーパーユーザとして

template1 データベースに接続します。プロンプトが表示されたら、認証情報を提供します。次に、次のコマンドを使用してデータベースを削除します。

```
DROP DATABASE <database_name>;
```

ここで、database_name はデータベースの名前です。

次に、template1 データベースの内容に基づいて空のデータベースを作成します。

```
CREATE DATABASE <database_name>;
```

Step 3 - Set both servers in trust mode

アップグレードプロセス中に、pg_upgrade は古いサーバーと新しいサーバーに数回接続します。接続プロセス make ために、あなたは trust に認証モードを設定し、pg_hba.conf ファイルを編集することができます。pg_hba.conf ファイルを変更するには、Start メニューから EDB Postgres メニューに移動します。Advanced Server メニューに移動し、Expert Configuration メニューを開きます。Edit pg_hba.conf メニューオプションを選択して、pg_hba.conf ファイルを開きます。

以前の AdvancedServer インストール、および AdvancedServer13 サーバーの信頼認証を許可する必要があります。次の図に示すように、AdvancedServer の両方のインストールの pg_hba.conf ファイルを編集します。

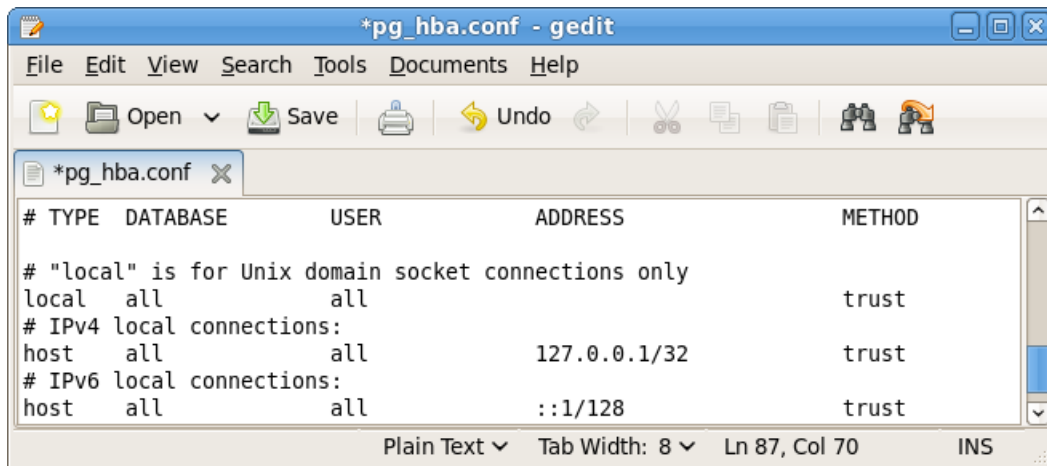


図 1 信頼認証を使用するための AdvancedServer の構成

各ファイルを編集した後、ファイルを保存してエディタを終了しファイル。

アップグレードプロセス中にシステムが md5 認証モードを維持する必要がある場合、パスワードファイル (Windows では pgpass.conf、Linux では .pgpass) でデータベーススーパーユーザのユーザパスワードを指定できます。パスワードファイルの設定の詳細については、次の URL で入手可能な PostgreSQL Core Documentation を参照してください。

<https://www.postgresql.org/docs/current/静的/libpq-pgpass.html>

Step 4 - Stop All Component Services and Servers

pg_upgrade を呼び出す前に、オリジナルの AdvancedServer インストール、AdvancedServer13、またはサポートコンポーネントに属するサービスを停止する必要があります。これにより、アップグレードプロセス中にサービスがいずれのクラスターにもアクセスしないようにします。

インストールで実行される可能性が最も高いサービスは次のとおりです。

Service:	On Linux:	On Windows:
PostgresPlusAdvancedServer9.0	ppas-9.0	ppas-9.0
PostgresPlusAdvancedServer9.1	ppas-9.1	ppas-9.1
PostgresPlusAdvancedServer9.2	ppas-9.2	ppas-9.2
PostgresPlusAdvancedServer9.3	ppas-9.3	ppas-9.3
PostgresPlusAdvancedServer9.4	ppas-9.4	ppas-9.4
PostgresPlusAdvancedServer9.5	ppas-9.5	ppas-9.5
EnterpriseDBPostgresAdvancedServer9.6	edb-as-9.6	edb-as-9.6
EnterpriseDBPostgresAdvancedServer10	edb-as-10	edb-as-10
EnterpriseDBPostgresAdvancedServer11	edb-as-11	edb-as-11
EnterpriseDBPostgresAdvancedServer12	edb-as-12	edb-as-12
EnterpriseDBPostgresAdvancedServer13	edb-as-13	edb-as-13
AdvancedServer9.0SchedulingAgent	ppasAgent-90	PostgresPlusAdvancedServer90 スケジューリングエージェント
AdvancedServer9.1 スケジューリングエージェント	ppasAgent-91	PostgresPlusAdvancedServer91 スケジューリングエージェント
AdvancedServer9.2 スケジューリングエージェント	ppas-agent-9.2	PostgresPlusAdvancedServer9.2 スケジューリングエージェント
AdvancedServer9.3 スケジューリングエージェント	ppas-agent-9.3	PostgresPlusAdvancedServer9.3 スケジューリングエージェント
AdvancedServer9.4 スケジューリングエージェント	ppas-agent-9.4	PostgresPlusAdvancedServer9.4 スケジューリングエージェント
AdvancedServer9.5 スケジューリングエージェント	ppas-agent-9.5	PostgresPlusAdvancedServer9.5 スケジューリングエージェント
AdvancedServer9.6 スケジューリングエージェント (pgAgent)	edb-pgagent-9.6	EnterpriseDBPostgresAdvancedServer9.6 スケジューリングエージェント
InfiniteCache9.2	ppas-infinitecache-9.2	なし
無限キャッシュ 9.3	ppas-infinitecache-9.3	なし
無限キャッシュ 9.4	ppas-infinitecache	なし
無限キャッシュ 9.5	ppas-infinitecache	なし
無限キャッシュ 9.6	edb-icache	なし
無限キャッシュ 10	edb-icache	なし
PgBouncer9.0	pgbouncer-90	pgbouncer-90

次のページに続く

表 1 – 前のページからの続き

Service:	On Linux:	On Windows:
PgBouncer9.1	pgbouncer-91	pgbouncer-91
PgBouncer9.2	pgbouncer-9.2	pgbouncer-9.2
PgBouncer9.3	pgbouncer-9.3	pgbouncer-9.3
PgBouncer	Pgbouncer	Pgbouncer
PgBouncer1.6	ppas-pgbouncer-1.6 または ppas-pgbouncer16	ppas-pgbouncer-1.6
PgBouncer1.7	edb-pgbouncer-1.7	edb-pgbouncer-1.7
PgPool9.2	ppas-pgpool-9.2	なし
PgPool9.3	ppas-pgpool-9.3	なし
PgPool	ppas-pgpool	なし
PgPool3.4	ppas-pgpool-3.4 または ppas- pgpool34	なし
pgPool-II	edb-pgpool-3.5	なし
スロニー 9.2	ppas-replication-9.2	ppas-replication-9.2
スロニー 9.3	ppas-replication-9.3	ppas-replication-9.3
スロニー 9.4	ppas-replication-9.4	ppas-replication-9.4
スロニー 9.5	ppas-replication-9.5	ppas-replication-9.5
スロニー 9.6	edb-slony-replication-9.6	edb-slony-replication-9.6
xDBPublicationServer9.0	edb-xdbpubserver-90	出版サービス 90
xDBPublicationServer9.1	edb-xdbpubserver-91	出版サービス 91
xDB サブスクリプションサー バー	edb-xdbsubserver-90	サブスクリプションサービス 90
xDB サブスクリプションサー バー	edb-xdbsubserver-91	サブスクリプションサービス 91
EDBReplicationServerv6.x	edb-xdbpubserver	xDBReplicationServer のパブリケーション サービス
EDBSubscriptionServerv6.x	edb-xdbsubserver	xDBReplicationServer のサブスクリプシ ョンサービス

To stop a service on Windows:

Services アプレットを開きます。リストに表示されている各 AdvancedServer またはサポートコンポーネントサービスを強調表示し、Stop を選択します。

To stop a service on Linux:

ターミナルウィンドウを開き、コマンドラインで各サービスを手動で停止します。

Step 5 for Linux only - Assume the identity of the cluster owner

Linux を使用している場合は、AdvancedServer クラスター所有者の ID を想定してください。（以下の例では、オラクル>・データベースモードとの互換性は、このようにクラスターの所有者として enterprisedb を割り当てる PostgreSQL>データベースモードとの互換性にインストールした場合、postgres がクラスターの所有者である AdvancedServer は、デフォルトでインストールされた前提としています。）

```
su - enterprisedb
```

プロンプトが表示されたら、AdvancedServer クラスターの所有者パスワード。次に、pg_upgrade 実行可能ファイルの場所を含むようにパスを設定しパス。

```
export PATH=$PATH:/usr/edb/as13/bin
```

アップグレードプロセス中に、pg_upgrade は enterprisedb ユーザのカレントワーキングディレクトリにファイルを書き込みます。enterprisedb ユーザが write 特権を持っているディレクトリから pg_upgrade を呼び出す必要があります。上記のコマンドを実行した後、enterprisedb ユーザがファイルを書き込むための十分な権限を持つディレクトリに移動します。

```
cd /tmp
```

ステップ 6 に進みます。

Step 5 for Windows only - Assume the identity of the cluster owner

Windows を使用している場合は、ターミナルウィンドウを開き、AdvancedServer クラスター所有者の ID を想定して、実行可能ファイル pg_upgrade へのパスを設定しパス。

AdvancedServer の初期インストール中に --serviceaccount service_account_user パラメータが指定された場合、service_account_user は AdvancedServer クラスター所有者であり、RUNAS コマンドで指定されるユーザです。

```
RUNAS /USER:service_account_user "CMD.EXE"  
SET PATH=%PATH%;C:\Program Files\edb\as13\bin
```

アップグレードプロセス中に、pg_upgrade はサービスアカウントユーザのカレントワーキングディレクトリにファイルを書き込みます。サービスアカウントユーザが write 権限を持っているディレクトリから pg_upgrade を呼び出す必要があります。上記のコマンドを実行した後、サービスアカウントユーザがファイルを書き込むための十分な権限を持つディレクトリに移動します。

```
cd %TEMP%
```

ステップ 6 に進みます。

AdvancedServer の初期インストール中に --serviceaccount パラメータが省略された場合、AdvancedServer サービスとデータベースクラスターのデフォルトの所有者は NT AUTHORITY/NetworkService です。

NT AUTHORITY/NetworkService がサービスアカウントユーザの場合、RUNAS コマンドはパスワードを求められ、NT AUTHORITY/NetworkService アカウントにはパスワードが割り当てられないため、使用できない場合があります。したがって、通常、「ユーザパスワードを取得できません」などのエラーメッセージ。

このような状況では、PsExec 名前付けの Windows ユーティリティプログラムを使用して、CMD.EXE をサービスアカウント NT AUTHORITY/NetworkService として実行する必要があります。

PsExec プログラムは、次のサイトで入手可能な PsTools をダウンロードして取得する必要があります。

<https://technet.microsoft.com/en-us/sysinternals/bb897553.aspx>。

その後、次のコマンドを使用して CMD.EXE を NT AUTHORITY/NetworkService として実行してから、pg_upgrade 実行可能ファイルへのパスを設定できます。

```
psexec.exe -u "NT AUTHORITY\NetworkService" CMD.EXE  
SET PATH=%PATH%;C:\Program Files\edb\as13\bin
```

アップグレードプロセス中に、pg_upgrade はサービスアカウントユーザのカレントワーキングディレクトリにファイルを書き込みます。サービスアカウントユーザが write 権限を持っているディレクトリから pg_upgrade を呼び出す必要があります。上記のコマンドを実行した後、サービスアカウントユーザがファイルを書き込むための十分な権限を持つディレクトリに移動します。

```
cd %TEMP%
```

ステップ 6 に進みます。

Step 6 - Perform a consistency check

アップグレードを試みる前に、一貫性チェックを実行して、古いクラスターと新しいクラスターが互換性があり、適切に構成されていることを確認します。pg_upgrade オプションを含めて、pg_upgrade に一貫性チェックを実行するよう指示します。

次の例は、Linux 上で pg_upgrade を呼び出して整合性チェックを実行する方法を示しています。

```
pg_upgrade -d /var/lib/edb/as12/data  
-D /var/lib/edb/as13/data -U enterprisedb  
-b /usr/edb/as12/bin -B /usr/edb/as13/bin -p 5444 -P 5445 --check
```

コマンドが成功すると、*Clusters are compatible* を結果ます。

Windows を使用している場合は、スペースを含むディレクトリ名をクオートする必要があります。

```
pg_upgrade.exe  
-d "C:\Program Files\ PostgresPlus\12AS\data"  
-D "C:\Program Files\edb\as13\data" -U enterprisedb  
-b "C:\Program Files\PostgresPlus\12AS\bin"  
-B "C:\Program Files\edb\as13\bin" -p 5444 -P 5445 --check
```

整合性チェックの処理中に、pg_upgrade はそれが pg_upgrade が呼び出されたディレクトリにあるファイルに見つけた矛盾をログします。整合性チェックが完了したら、ファイルを確認して、欠落しているコンポーネントまたはアップグレードの競合を特定します。pg_upgrade を起動してバージョンのアップグレードを実行する前に、競合を解決する必要があります。

pg_upgrade がコンポーネントの不足を警告する場合、StackBuilderPlus を使用して、コンポーネントを含むコンポーネントを追加できコンポーネント。StackBuilderPlus を使用する前に、AdvancedServer13 サービスをリスタートする必要があります。サービスを再起動した後、Start メニューから Advanced Server 13 メニューに移動し、StackBuilder Plus を選択して StackBuilderPlus を開きます。StackBuilderPlus ウィザードの画面上のアドバイスに従って、不足しているコンポーネントをダウンロードしてインストールします。

pg_upgrade がクラスターに互換性があることを確認したら、バージョンアップグレードを実行できます。

Step 7 - Run pg_upgrade

クラスターに互換性があることを確認したら、pg_upgrade を呼び出して、古いクラスターを AdvancedServer の新しいバージョンにアップグレードできます。

Linux の場合:

```
pg_upgrade -d /var/lib/edb/as12/data
-D /var/lib/edb/as13/data -U enterprisedb
-b /usr/edb/as12/bin -B /usr/edb/as13/bin -p 5444 -P 5445
```

Windows:

```
pg_upgrade.exe -d "C:\Program Files\PostgresPlus\12AS\data"
-D "C:\Program Files\edb\as13\data" -U enterprisedb
-b "C:\Program Files\PostgresPlus\12AS\bin"
-B "C:\Program Files\edb\as13\bin" -p 5444 -P 5445
```

pg_upgrade は、アップグレードの進行状況を画面に表示します。

```
$ pg_upgrade -d /var/lib/edb/as12/data -D /var/lib/edb/as13/data -U
enterprisedb -b /usr/edb/as12/bin -B /usr/edb/as13/bin -p 5444 -P 5445
Performing Consistency Checks
-----
Checking current, bin, and data directories          ok
Checking cluster versions                            ok
Checking database user is a superuser                ok
Checking for prepared transactions                   ok
Checking for reg* system OID user data types         ok
Checking for contrib/isn with bigint-passing mismatch ok
Creating catalog dump                                ok
Checking for presence of required libraries          ok
Checking database user is a superuser                ok
Checking for prepared transactions                   ok
```

(次のページに続く)

(前のページからの続き)

```
If pg_upgrade fails after this point, you must re-initdb the
new cluster before continuing.
```

```
Performing Upgrade
```

```
-----
Analyzing all rows in the new cluster          ok
Freezing all rows on the new cluster           ok
Deleting files from new pg\_clog               ok
Copying old pg\_clog to new server             ok
Setting next transaction ID for new cluster    ok
Resetting WAL archives                        ok
Setting frozenxid counters in new cluster      ok
Creating databases in the new cluster          ok
Adding support functions to new cluster        ok
Restoring database schema to new cluster       ok
Removing support functions from new cluster    ok
Copying user relation files                   ok

Setting next OID for new cluster               ok
Creating script to analyze new cluster         ok
Creating script to delete old cluster          ok
```

```
Upgrade Complete
```

```
-----
Optimizer statistics are not transferred by pg\_upgrade so,
once you start the new server, consider running:
```

```
    analyze_new_cluster.sh
```

```
Running this script will delete the old cluster's data files:
```

```
    delete_old_cluster.sh
```

pg_upgrade の実行中に、アップグレード中に発生した特別な状況をハンドルする SQL スクリプトが生成される場合があります。古いクラスタは、ラージオブジェクトが含まれている場合例、あなたは新しいクラスタ内のオブジェクトのデフォルトのアクセス許可を定義するスクリプトを起動する必要があるかもしれません。アップグレード前の整合性チェックを実行すると、pg_upgrade は、手動で実行する必要がある可能性のあるスクリプトを警告します。

pg_upgrade の完了後にスクリプトを呼び出す必要があります。スクリプトを呼び出すには、EDB-PSQL コマンドラインクライアントを使用してデータベーススーパーユーザとして新しいクラスターに接続し、/i オプションを使用して各スクリプトを呼び出します。

```
\i complete_path_to_script/script.sql
```

通常、再構築スクリプトが完了するまで、再構築スクリプトで被参照れるテーブルにアクセスすることは安全ではありません。テーブルにアクセスと、誤った結果やパフォーマンスの低下が生じる可能性があります。再構築ス

リプトで被参照されていないテーブルには、すぐにアクセスできます。

pg_upgrade がアップグレードプロセスの完了に失敗した場合、古いクラスターは変更されませんが、\$PGDATA/global/pg_control の名前は pg_control.old に変更され、各テーブルスペースの名前は tablespace.old に変更されます。呼び出し前の状態に戻すには:

1. 新しいクラスターによって作成されたテーブルスペースディレクトリを削除します。
2. .old のサフィックスを削除して、\$PGDATA/global/pg_control の名前を変更します。
3. .old サフィックスを削除して、古いクラスターテーブルスペースのディレクトリ名を変更します。
4. アップグレードが失敗する前に移動された可能性のあるデータベースオブジェクトを（新しいクラスターから）削除します。

これらの手順を実行した後、アップグレードを再試行する前に、発生したアップグレードの競合を解決してください。

アップグレードが完了すると、pg_upgrade は新しいクラスターのバキューム処理を推奨し、古いクラスターを削除できるスクリプトを提供します。

注釈: 古いクラスターを削除する前に、クラスターが期待どおりにアップグレードされていること、および以前のバージョンに戻す必要がある場合に備えてクラスターのバックアップを保持している保証してください。

Step 8 - Restore the authentication settings in the pg_hba.conf file

あなたが許可 trust 認証に pg_hba.conf ファイルを変更した場合は、お好みの認証設定を反映するために pg_hba.conf ファイルの内容を更新。

Step 9 - Move and identify user-defined tablespaces (Optional)

ユーザー定義のテーブルスペースにデータが保存されている場合、アップグレード処理後にテーブルスペースファイルを手動で再配置する必要があります。新しい場所にファイルを移動し、ファイルを指すように（クラスタの data ディレクトリの下に pg_tblspc ディレクトリにある）シンボリックリンクを更新。

4.4 Upgrading a pgAgent Installation

既存の AdvancedServer インストールで pgAgent を使用する場合、AdvancedServer13 インストーラで提供されるスクリプトを使用して pgAgent を更新できます。スクリプトの名前付けは `dbms_job.upgrade.script.sql` であり、AdvancedServer インストールの `/share/contrib/` ディレクトリにあります。

`pg_upgrade` を使用してインストールをアップグレードする場合、次のことを行う必要があります。

1. アップグレードを実行します。
2. `dbms_job.upgrade.script.sql` スクリプトを呼び出して、カタログファイルを更新します。既存の pgAgent インストールがスクリプトを使用して実行された場合、更新によりインストールが拡張機能に変換されます。

4.5 pg_upgrade Troubleshooting

このセクションのトラブルシューティングのヒントは、pg_upgrade の使用時に発生する可能性のある問題に対処します。

4.5.1 Upgrade Error - There seems to be a postmaster servicing the cluster

pg_upgrade からポストマスターがクラスターにサービスを提供していることが報告された場合は、すべての AdvancedServer サービスを停止し、アップグレードを再試行してください。

4.5.2 Upgrade Error - fe_sendauth:no password supplied

pg_upgrade が不足しているパスワードを参照する認証エラーを報告する場合、新旧クラスターの pg_hba.conf ファイルを変更して trust 認証を有効にするか、pgpass.conf ファイルを使用するようにシステムを構成してください。

4.5.3 Upgrade Error - New cluster is not empty; exiting

pg_upgrade から新しいクラスターが空でないことが報告された場合は、新しいクラスターを空にしてください。ターゲットクラスターには、ユーザー定義のデータベースを含めることはできません。

4.5.4 Upgrade Error - Failed to load library

オリジナルの AdvancedServer クラスターに AdvancedServer13 クラスターに含まれていないライブラリが含まれていた場合、pg_upgrade は、pg_upgrade を呼び出したディレクトリ内の loadable_libraries.txt ファイルにエントリーを書き込むことにより、整合性チェック中に欠落コンポーネントを警告します。通常、メジャーコンポーネントアップグレードの一部ではないライブラリが見つからない部分は、次の手順を実行します。

1. AdvancedServer サービスを再起動します。

StackBuilderPlus を使用して、不足しているモジュールをダウンロードしてインストールします。次に:

2. AdvancedServer サービスを停止します。
3. アップグレードプロセスを再開します。pg_upgrade を呼び出して、整合性チェックを実行します。
4. 整合性チェックで指摘された残りの問題を解決したら、pg_upgrade を呼び出して、古いクラスターから新しいクラスターへのデータ移行を実行します。

4.6 Reverting to the Old Cluster

前のクラスターに戻すために使用されるメソッドは、`pg_upgrade` を呼び出すときに指定されたオプションによって異なります。

- `pg_upgrade` の呼び出し時に `--check` オプションを指定した場合、アップグレードは実行されておらず、古いクラスターは変更されていません。古いクラスターはいつでも再利用できます。
- `pg_upgrade` を呼び出すときに `--link` オプションを含めた場合、データファイルはアップグレードの完了後に古いクラスターと新しいクラスターの間で共有されます。新しいクラスターにサービスを提供しているサーバーを起動した場合、新しいサーバーはそれらの共有ファイルに書き込みを行っているため、古いクラスターを使用するのは安全ではありません。
- `--link` を指定せずに `pg_upgrade` を実行した場合、または新しいサーバーを起動していない場合、`.old` サフィックスが `$PGDATA/global/pg_control` およびテーブルスペースディレクトリに追加されていることを除いて、古いクラスターは変更されません。
- 古いクラスターを再利用するには、新しいクラスターによって作成されたテーブルスペースディレクトリを削除し、`$PGDATA/global/pg_control` から `.old` サフィックスを削除し、古いクラスターテーブルスペースディレクトリ名を削除し、古いクラスターを処理するサーバーをリスタートします。

第 5 章

Performing a Minor Version Update of an RPM Installation

RPM パッケージを使用して AdvancedServer またはそのサポートコンポーネントをインストールした場合、yum を使用して、マイナーバージョンをより新しいバージョンにアップグレード。システムで利用可能なパッケージ更新のリストを確認するには、コマンドラインを開き、ルート権限を引き受けて、コマンドを入力します。

```
yum check-update <package_name>
```

package_name は、更新を検索する検索語です。検索語にワイルドカード値を含めることができることにノートしてください。yum update を使用して更新されたパッケージをインストールするには、次のコマンドを使用します。

```
yum update <package_name>
```

package_name は、更新するパッケージの名前です。更新コマンドにワイルドカード値を含めて、単一のコマンドでマルチブルの関連パッケージソフトを更新します。例、名前が式 edb を含め、すべてのパッケージソフトを更新するには、次のコマンドを使用します。

```
yum update edb*
```

注釈: yum update コマンドは、マイナーリリース間の更新のみを実行します。メジャーリリース間で更新するには、pg_upgrade を使用する必要があります。

yum コマンドとオプションの使用の詳細については、コマンドラインに yum --help を入力するか、次のサイトにアクセスしてください。

https://access.redhat.com/文書/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/ch-yum.html

第 6 章

Using StackBuilder Plus to Perform a Minor Version Update

StackBuilderPlus は、Windows システムでのみサポートされています。

StackBuilderPlus ユーティリティは、AdvancedServer インストールを補完するモジュールの更新、ダウンロード、インストールのプロセスを簡素化するグラフィカルインタフェースを提供します。StackBuilderPlus でモジュールをインストールすると、StackBuilderPlus はソフトウェアの依存関係を自動的に解決します。

インストールが完了したら、Apps メニューから StackBuilder Plus メニューオプションを選択して、StackBuilderPlus をいつでも呼び出すことができます。システムパスワード（プロンプトが表示された場合）、StackBuilderPlus のウェルカムウィンドウが開きます。

ようこそウィンドウのドロップダウンリストボックスを使用して、AdvancedServer インストールを選択します。

StackBuilderPlus にはインターネットアクセスが必要です。AdvancedServer のインストールがファイアウォールの背後にある場合（インターネットアクセスが制限されている場合）、StackBuilderPlus はプロキシサーバー経由でプログラムインストーラーをダウンロードできます。モジュールプロバイダーは、HTTP プロキシまたは FTP プロキシを介してモジュールにアクセスできるかどうかを判断します。現在、すべての更新は HTTP プロキシ経由で転送され、FTP プロキシ情報は使用されません。

選択した AdvancedServer インストールでインターネットアクセスが制限されている場合は、Welcome ウィンドウの Proxy Servers を使用して Proxy servers ダイアログを開きます（次の図を参照）。

Proxy servers ダイアログの HTTP proxy にプロキシサーバーの IP アドレスとポート番号を入力します。現在、すべての StackBuilderPlus モジュールは HTTP プロキシ経由で配布されています（FTP プロキシ情報は無視されます）。OK をクリックして続行します。

StackBuilderPlus モジュール選択ウィンドウ（図を参照）のツリーコントロールには、各モジュールカテゴリのノードが表示されます。

選択した AdvancedServer インストールに新しいコンポーネントを追加するか、コンポーネントをアップグレードするには、モジュール名前の左側のボックスをオンにして、Next をクリックします。プロンプトが表示された



図 1 StackBuilderPlus ウェルカムウィンドウ

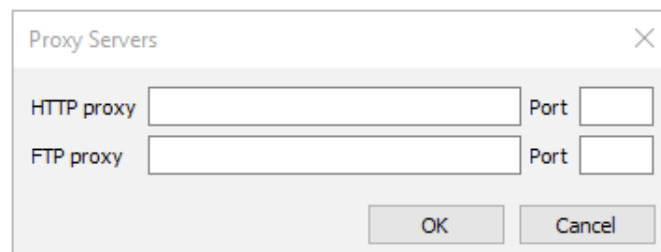


図 2 プロキシサーバーダイアログ

ら、StackBuilderPlus 登録ウィンドウでメールアドレスとパスワードします。

StackBuilderPlus は選択されたパッケージソフトを確認します。Selected packages ダイアログには、インストーラの名前とバージョンが表示されます。Next をクリックして続行します。

ダウンロードが完了すると、インストールファイルがダウンロードされ、インストールの準備ができていることを確認するウィンドウが開きインストール。

Skip Installation の横にあるチェックボックスをオンにし、Next を選択してダウンロードしたファイルをインストールせずに StackBuilderPlus を終了するか、チェックを外して Next をクリックしてインストールプロセスをスタートします。

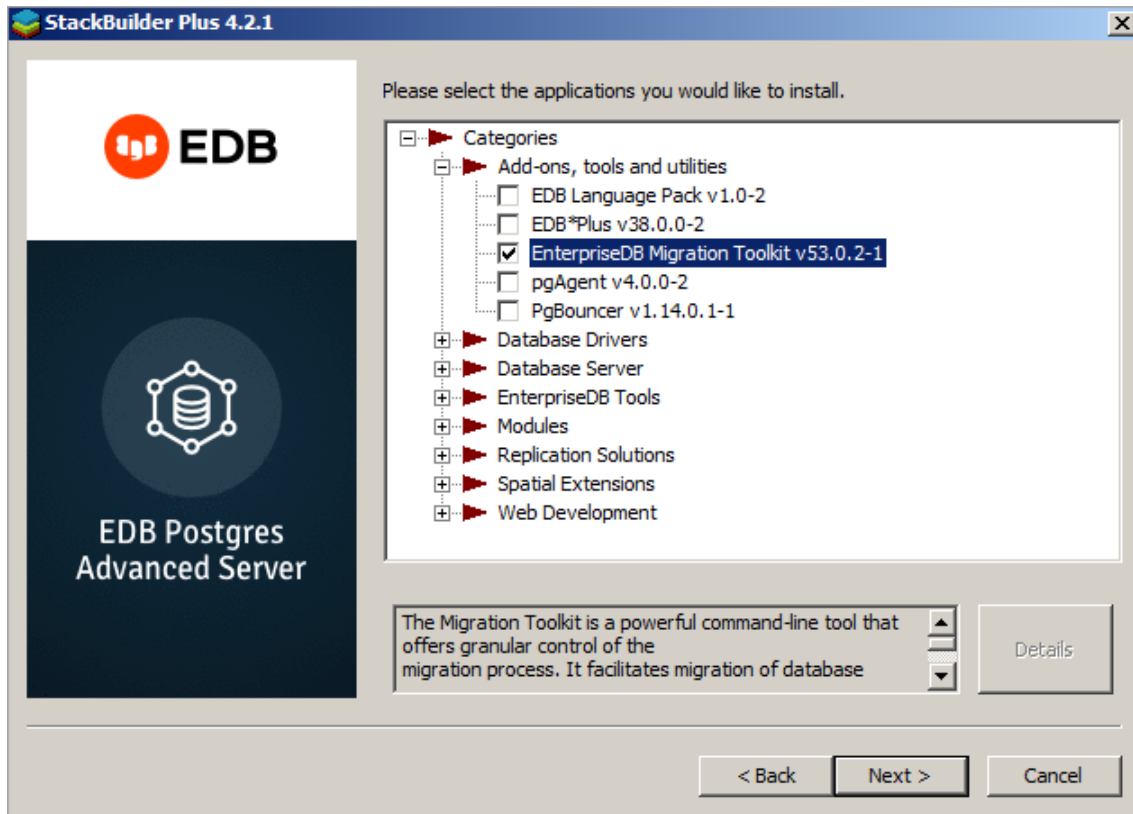


図3 StackBuilderPlus モジュール選択ウィンドウ

アップグレードが完了すると、StackBuilderPlus は要求されたパッケージのインストールの成功または失敗を警告します。インストーラからコンピュータのリスタートを求められた場合は、ここで再起動します。

注釈: 更新プログラムのインストールに失敗した場合、StackBuilderPlus はポップアップダイアログでインストールエラーを警告し、%TEMP% のログファイルにメッセージを書き込みます。

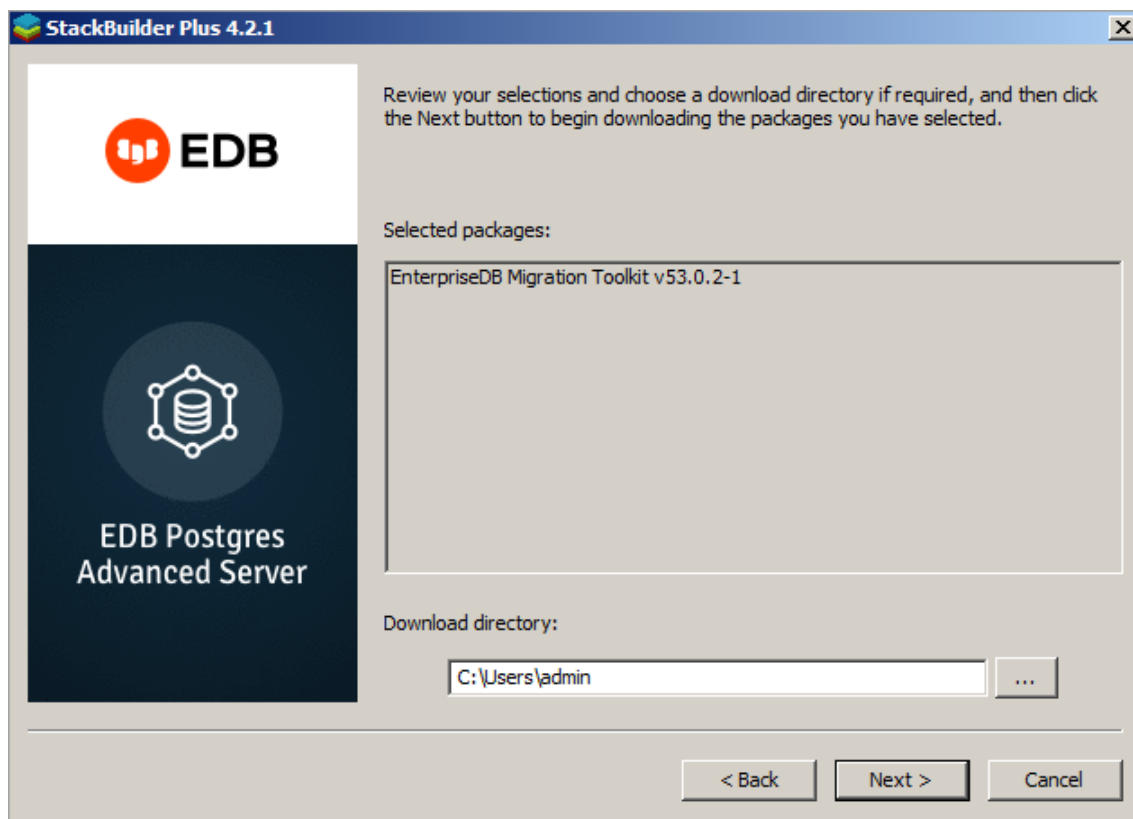


図 4 サマリウィンドウに選択したパッケージソフトのリストが表示されます

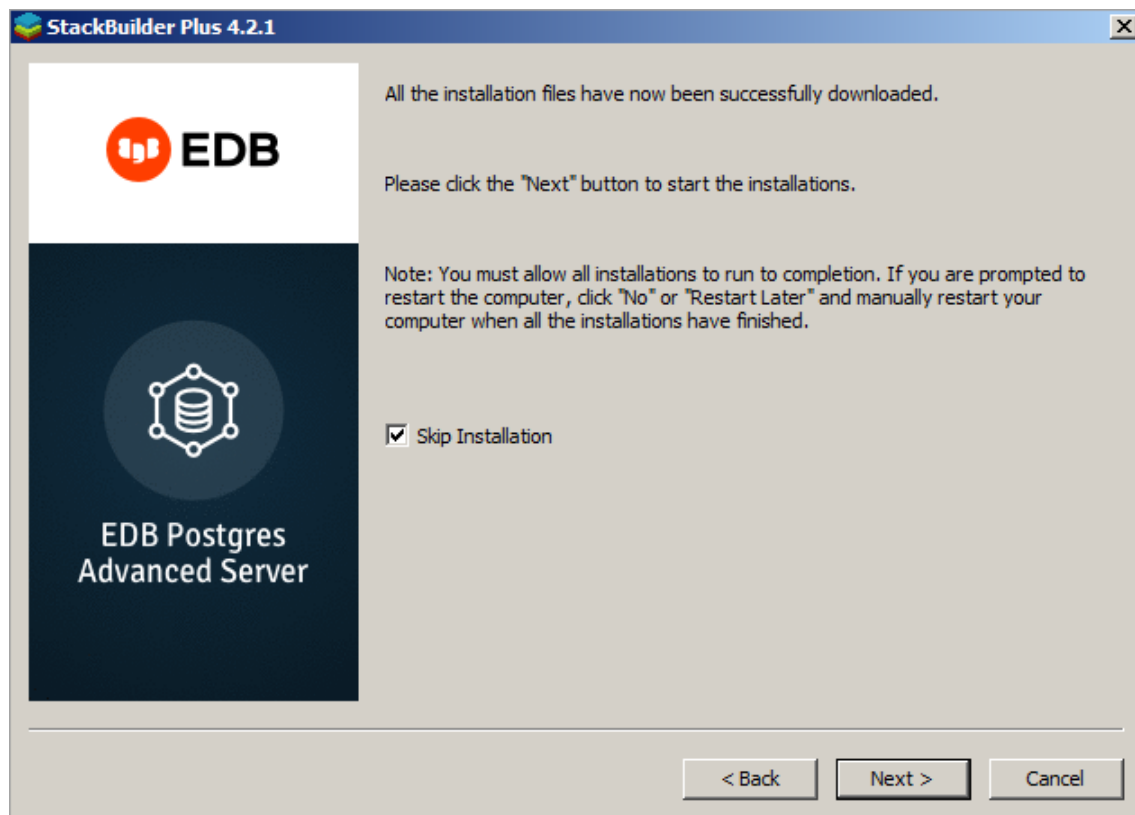


図 5 ダウンロードプロセスが完了したことの確認

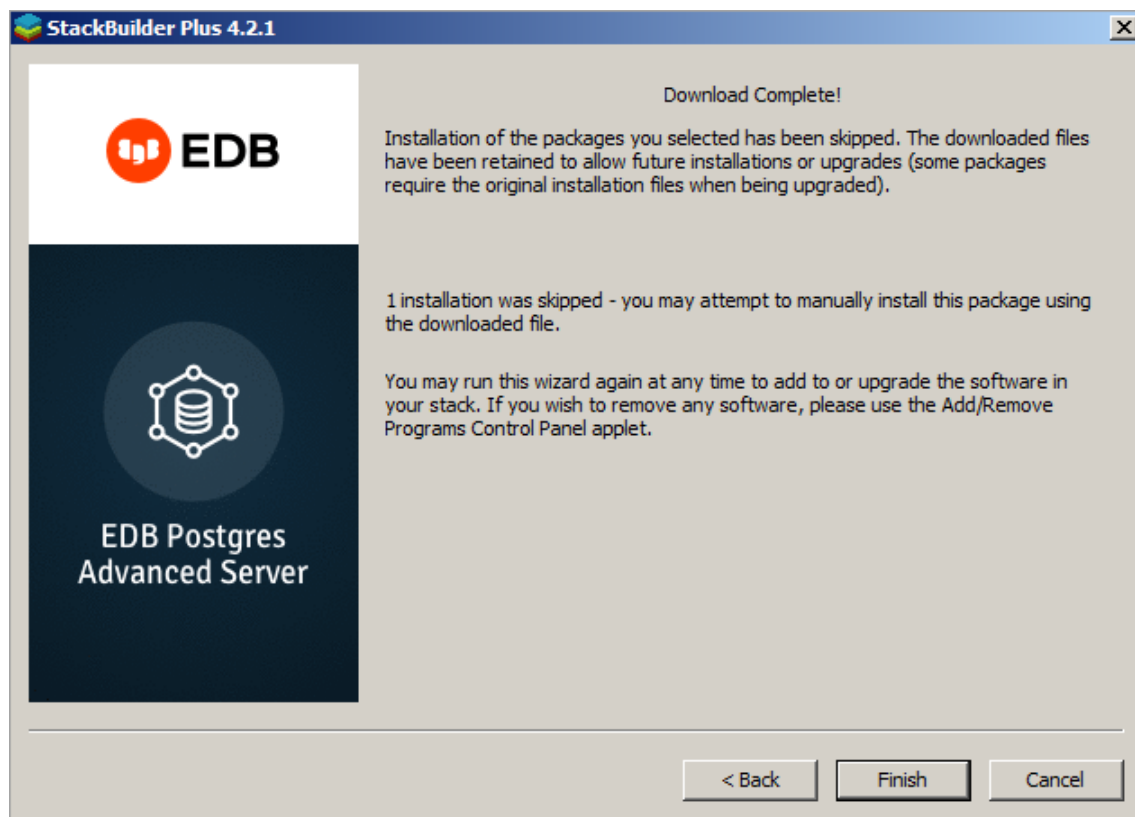


図 6 StackBuilderPlus は完了したインストールを確認します

第 7 章

Migration to Version 13

以前のリリースからデータを移行するには、`pg_dumpall` または `pg_upgrade` を使用したダンプ/復元またはロジカルレプリケーションが必要です。[pg_upgrade のインストールのアップグレード](#) を参照して、新しいメジャーリリースに移行してください。

バージョン 13 には、以前のリリースとの互換性に影響する可能性のある多くの変更が含まれています。次の非互換性がリストされています。

- `SIMILAR TO ... ESCAPE NULL` を変更して `NULL` を結果ます。

この新しい動作は、SQL 仕様に一致します。以前は、デフォルトのエスケープ文字列（バックスラッシュ文字）を使用することを意味するために、`ESCAPE` 値が `null` でした。これは `substring(text FROM pattern ESCAPE text)` にも適用されます。以前の動作は、オリジナルのファンクションを変更せずに維持することにより、古いビューで保持されています。

- `json[b]_to_tsvector()` が `string` オプションのスペルを完全にチェックするようにします。
- デフォルト以外の `effective_io_concurrency` 値が並行性に与える影響を変更します。

以前は、この値は、同時リクエストの数を設定する前に調整されていました。これで、値が直接使用されます。古い値から新しい値への変換は、次を使用して実行できます。

```
SELECT round(sum(OLDVALUE / n::float)) AS newvalue FROM generate_
series(1, OLDVALUE) s(n);
```

- `pg_stat_ssl` および `pg_stat_gssapi` システムビューで補助プロセスが表示されないようにします。これらのビューを `pg_stat_activity` に結合し、補助プロセスを表示結合クエリでは、左結合を使用する必要があります。
- 一貫性を改善するために、さまざまな `wait events` の名前を変更します。
- `ALTER FOREIGN TABLE ... RENAME COLUMN` を修正して、より適切なコマンドタグを結果ます。以前は `ALTER TABLE` を返しました。現在、`ALTER FOREIGN TABLE` を返します。

- `ALTER MATERIALIZED VIEW ... RENAME COLUMN` を修正して、より適切なコマンドタグを結果返します。以前は `ALTER TABLE` を返しました。現在、`ALTER MATERIALIZED VIEW` を返します。
- 設定パラメータ `wal_keep_segments` の名前を `wal_keep_size` に変更します。

これにより、スタンバイサーバ用に保持する WAL の量が決まります。古いパラメータのようにファイル数ではなく、メガバイト単位で指定されます。以前に `wal_keep_segments` を使用した場合、次の式はほぼ同等の設定を提供します。

```
wal_keep_size = wal_keep_segments * wal_segment_size (typically 16MB)
```

- PostgreSQL8.0 より前の構文を使用した演算子クラスの定義のサポートを削除し構文。
- PostgreSQL7.3 より前の構文を使用した外部キーコンストレインの定義のサポートを削除し構文。
- PostgreSQL7.3 より前のサーバで使用されていた `"opaque"` 疑似データ型のサポートを削除します。
- パッケージ化されていない (9.1 より前の) 拡張機能のアップグレード処理のサポートを削除します。

`CREATE EXTENSION` の `FROM` オプションはサポートされなくなりました。まだパッケージ化されていない拡張機能を使用しているインストールは、PostgreSQL13 に更新する前にパッケージ化されたバージョンにアップグレードする必要があります。

- タイムゾーンデータベースの `posixrules` ファイルのサポートを削除します。

IANA のタイムゾーングループはこの機能を非推奨にしました。つまり、今後数年間でシステムのタイムゾーンデータベースから次第に消えていきます。タイムゾーンデータの更新で予期しない動作の変化が現れるのではなく、バージョン 13 からこの機能に対する PostgreSQL のサポートを削除しました。これは、明示的な夏時間移行ルールがない POSIX スタイルのタイムゾーン仕様の動作にのみ影響します。以前の移行ルールは、カスタム `posixrules` ファイルをインストールすることで決定できましたが、現在はハードワイヤードです。影響を受けるインストールの推奨される修正プログラムは、地理的なタイムゾーン名前の使用をスタートすることです。

- `ltree` で、`lquery` パターンにブレース付きの隣接するアスタリスクが含まれている場合、たとえば `*{2}.*{3}` は、それを `* {5}` として適切に解釈します。
- `pageinspect` の `bt_metap()` を修正して、オーバーフローの可能性が低い、より適切なデータ型を結果返します。
- `SELECT DISTINCT` クエリーの動作の `SELECT DISTINCT...ORDER BY` 句は、アップグレード後に異なります。

`SELECT DISTINCT` が指定されている場合、または `SELECT` ステートメントに `SELECT DISTINCT ... ORDER BY` 句が含まれている場合、`ORDER BY` のすべての式が `SELECT DISTINCT` クエリーの選択リストに存在する必要があります (バージョン 9.5 または 9.6 から AdvancedServer の上位バージョンにアップグレード処理する場合に適用可能)。

- ICU を使用して以前のバージョン（12、11、10、9.6、9.5）から最新の AdvancedServer バージョンに移行するには、照合順序に依存するすべてのオブジェクトを `pg_upgrade` を使用して再構築する必要があります。

データベースシステムは特定のソートオーダーを持つ格納オブジェクトに依存しているため、照合順序定義の変更はインデックスの破損やその他の問題を引き起こす可能性があります。通常、これは避けるべきですが、`pg_upgrade` を使用して ICU の新しいバージョンにリンクされたサーバーバイナリにアップグレードする場合など、正当な状況で発生する可能性があります。これが発生した場合、照合順序に依存するすべてのオブジェクトを、例 `REINDEX` を使用して再構築する必要があります。完了したら、コマンド `ALTER COLLATION ... REFRESH VERSION` を使用して照合順序バージョンを更新できます。これは、現在のコレータバージョンを記録するためのシステムカタログを更新するとワーニングが離れて行く `make` ます。これは、影響を受けるすべてのオブジェクトが正しく再構築されたかどうかを実際にはチェックしないことに注意してください。

第 8 章

Conclusion

EDBPostgres™AdvancedServer アップグレードガイド

Copyright©2007-2020EnterpriseDBCorporation。

予約を禁じます。

EnterpriseDB

34CrosbyDrive、Suite201、Bedford、MA01730、アメリカ

T+17813573390F+19784671307E

info@enterprisedb.com

www.enterprisedb.com

- EnterpriseDB および PostgresEnterpriseManager は、EnterpriseDBCorporation の登録商標です。EDB および EDBPostgres は、EnterpriseDBCorporation の商標です。オラクルは、オラクル Inc の登録商標です。その他の商標は、それぞれの所有者の商標です。
- EDB は、存在する場合、EDB 製品のログオン 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、および入力バリデーションを行います。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、限定されたバリデーション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危険な入力を禁止することはできません。
- EDB は、フリーフォーム SQL または将来的に認証され、信頼されたユーザーからの他の潜在的に危険な入力を受け入れるの製品に機能を追加する権利を留保しますが、彼らは最低限のリスクの可能性を提供確保する保証に設計され、テストされているようなすべての機能を保証し、どこ可能であれば、スーパーユーザ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ないため、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完全に保証することはできません。

索引

Command Line Options - Reference, 13
Conclusion, 41

Invoking pg_upgrade, 11

Limitations, 5
Linking versus Copying, 10

Migration to Version 13, 37

Performing a Minor Version Update of an RPM
Installation, 29
Performing an Upgrade, 9
pg_upgrade Troubleshooting, 27

Reverting to the Old Cluster, 28

Supported Platforms, 3

Upgrade Error - Failed to load library, 27
Upgrade Error - fe_sendauth: no password
supplied, 27
Upgrade Error - New cluster is not empty
exiting, 27
Upgrade Error - There seems to be a postmaster
servicing the cluster, 27
Upgrading a pgAgent Installation, 26
Upgrading an Installation With pg_upgrade, 7
Upgrading to Advanced Server 13, 17
Using StackBuilder Plus to Perform a Minor
Version Update, 31