



EDB

Troubleshooting

リリース 15.2.0

EnterpriseDB Corporation

2023 年 03 月 03 日

目次

第 1 章 Linux installation troubleshooting	3
1.1 Installation troubleshooting	3
1.1.1 Java ベースのアプリケーションの表示が難しい	3
1.1.2 既存のデータディレクトリの内容が原因でインストールが完了しない	3
1.1.3 EPEL リリースパッケージのインストールが難しい	3
1.2 Enabling core dumps	4
1.2.1 RHEL、CentOS、Rocky Linux または AlmaLinux ホストでコアダンプを有効にする	4
1.2.2 Debian または Ubuntu ホストでコアダンプを有効にする	6
第 2 章 Windows installation troubleshooting	9
2.1 Java ベースのアプリケーションの表示が難しい	9
2.2 EDB Postgres Advanced Server のインストール中のエラー	9
2.3 既存のデータディレクトリの内容が原因でインストールが完了しない	9

このセクションでは、次に関する詳細情報を提供します。

- Linux プラットフォームでのインストールのトラブルシューティングとコアダンプの有効化。
- Windows プラットフォームでのインストールのトラブルシューティング。

第1章 Linux installation troubleshooting

このセクションでは、次に関する詳細情報を提供します。

- Linux プラットフォームでのインストールのトラブルシューティング。
- Linux プラットフォームでコアダンプを有効にします。

1.1 Installation troubleshooting

1.1.1 Java ベースのアプリケーションの表示が難しい

Java ベースのサーバー機能（コントロールまたはテキストが正しく表示されない、または空白のウィンドウ）の表示に問題が発生した場合、最新の `libxcb-xlib` ライブラリにアップグレードすると、ほとんどのディストリビューションで問題が修正されるはずです。その他の回避策については、

[Java bug database](#) を参照してください。

1.1.2 既存のデータディレクトリの内容が原因でインストールが完了しない

データディレクトリに既存のコンテンツが原因でインストールが完了しない場合、サーバーはサーバーログにエラーメッセージを書き込みます。

A data directory is neither empty, or a recognisable data directory.

同様のメッセージが表示された場合は、データディレクトリが空であることを確認する必要があります。ファイル（システム生成の `lost+found` フォルダーを含む）が存在するため、インストールが完了しません。インストーラーを再呼び出ししてインストールを完了する前に、データディレクトリからファイルを削除するか、データディレクトリに別の場所を指定します。

1.1.3 EPEL リリースパッケージのインストールが難しい

EPEL リリースパッケージのインストールに問題が発生した場合、次のコマンドを使用して、RHEL/CentOS 7.x および RHEL/Rocky Linux/AlmaLinux 8.x に `epel-release` パッケージをインストールできます。

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
```

```
dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

CentOS-Base.repo ファイル (/etc/yum.repos.d にあります) で [extras] リポジトリ定義を有効にする必要がある場合があります。yum が epel-release を含むリポジトリにアクセスできない場合、エラーメッセージが表示されます。

```
No package epel available.  
Error: Nothing to do
```

このエラーを受け取った場合は、EPEL rpm パッケージをダウンロードして手動でインストールできます。EPEL を手動でインストールするには、rpm パッケージをダウンロードし、スーパーユーザー権限を引き受けて、パッケージが含まれるディレクトリに移動し、次のコマンドで EPEL をインストールします。

```
yum -y install epel-release  
  
dnf -y install epel-release
```

1.2 Enabling core dumps

コアダンプを使用して、エラーを診断またはデバッグできます。コアダンプは、プロセスが予期せず終了したときのプロセスのアドレス空間（メモリ）を含むファイルです。コアダンプは、オンデマンド（デバッガーなど）で生成されるか、終了時に自動的に生成されます。

1.2.1 RHEL、CentOS、Rocky Linux または AlmaLinux ホストでコアダンプを有効にする

RHEL/CentOS 7.x または RHEL/Rocky Linux/AlmaLinux 8.x では、コアファイルの作成はデフォルトで無効になっています。コアファイルの生成を有効にするには、次のコマンドに従います。

- `ulimit -c` または `ulimit -a` コマンドを使用して、システムの電流制限を特定します。0 は、コアファイルの生成が無効になっていることを示します。

```
# ulimit -c  
0  
  
# ulimit -a  
core file size          (blocks, -c) 0  
data seg size           (kbytes, -d) unlimited  
scheduling priority     (-e) 0  
file size                (blocks, -f) unlimited  
pending signals          (-i) 3756  
max locked memory        (kbytes, -l) 64  
max memory size          (kbytes, -m) unlimited  
open files               (-n) 1024  
pipe size                (512 bytes, -p) 8  
POSIX message queues     (bytes, -q) 819200
```

(次のページに続く)

(前のページからの続き)

```
real-time priority      (-r) 0
stack size              (kbytes, -s) 8192
cpu time                (seconds, -t) unlimited
max user processes      (-u) 3756
virtual memory          (kbytes, -v) unlimited
file locks              (-x) unlimited
```

- コアダンプを保存する新しいディレクトリを作成し、`kernel.core_pattern` を変更して指定したディレクトリにダンプを保存します。

```
mkdir -p /var/coredumps
chmod a+w /var/coredumps

sysctl kernel.core_pattern=/var/coredumps/core-%e-%p
kernel.core_pattern = /var/coredumps/core-%e-%p
```

- 次のコマンドを使用して、再起動しても `kernel.core_pattern` 設定を維持します。

```
echo kernel.core_pattern=/var/coredumps/core-%e-%p >> /etc/sysctl.conf
```

- `/etc/security/limits.conf` のコアダンプを有効にして、ユーザーがコアファイルを作成できるようにします。各行は、次の形式でユーザーの制限を説明します。

```
<domain> <type> <item> <value>
*          soft   core   unlimited
```

* を使用して、コアダンプサイズを無制限に有効にします。

- 次のコマンドを使用して、コアファイルのサイズの制限を `UNLIMITED` に設定します。

```
ulimit -c unlimited

ulimit -c
unlimited
```

- サービスのコア制限を設定するには、次の設定を `/usr/lib/systemd/system/edb-as-15.service` に追加します。

```
[Service]
LimitCore=Infinity
```

- サービス構成をリロードします。

```
systemctl daemon-reload
```

- `systemd` を使用してグローバルなデフォルト制限を変更し、次の設定を `/etc/systemd/system.conf` に追加します。

```
DefaultLimitCORE=Infinity
```

- systemd を再起動します。

```
systemctl daemon-reexec
```

- EDB Postgres Advanced Server を停止してから起動します。

```
systemctl stop edb-as-15
systemctl start edb-as-15
```

- コアダンプが有効になりました。次のコマンドを使用して gdb ツールとデバッグパッケージをインストールします。

```
yum install gdb
debuginfo-install edb-as15 edb-as15-server-contrib edb-as15-server edb-as15-libs
```

- コアダンプファイルへのパスを置き換えてから、bt コマンドを使用してバックトレースを分析します。

```
gdb /usr/edb/as15/bin /var/coredumps/core-edb-postgres-65499
(gdb) bt full
```

1.2.2 Debian または Ubuntu ホストでコアダンプを有効にする

Debian 10 または Ubuntu 18、20 では、コアファイルの作成はデフォルトで無効になっています。コアファイルの生成を有効にするには、次のコマンドに従います。

- ulimit -c または ulimit -a コマンドを使用して、システムの電流制限を特定します。0 は、コアファイルの生成が無効になっていることを示します。

```
# ulimit -c
0

# ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority     (-e) 0
file size               (blocks, -f) unlimited
pending signals         (-i) 7617
max locked memory       (kbytes, -l) 65536
max memory size         (kbytes, -m) unlimited
open files              (-n) 1024
pipe size               (512 bytes, -p) 8
POSIX message queues    (bytes, -q) 819200
```

(次のページに続く)

(前のページからの続き)

```
real-time priority      (-r) 0
stack size              (kbytes, -s) 8192
cpu time                (seconds, -t) unlimited
max user processes      (-u) 7617
virtual memory          (kbytes, -v) unlimited
file locks              (-x) unlimited
```

- コアダンプを保存する新しいディレクトリを作成し、`kernel.core_pattern` を変更して指定したディレクトリにダンプを保存します。

```
mkdir -p /var/coredumps
chmod a+w /var/coredumps

sysctl kernel.core_pattern=/var/coredumps/core-%e-%p
kernel.core_pattern = /var/coredumps/core-%e-%p
```

- 次のコマンドを使用して、再起動しても `kernel.core_pattern` 設定を維持します。

```
echo kernel.core_pattern=/var/coredumps/core-%e-%p >> /etc/sysctl.conf
```

- `/etc/security/limits.conf` のコアダンプを有効にして、ユーザーがコアファイルを作成できるようにします。各行は、次の形式でユーザーの制限を説明します。

```
<domain> <type> <item> <value>
*          soft   core   unlimited
```

* を使用して、コアダンプサイズを無制限に有効にします。

- 次のコマンドを使用して、コアファイルのサイズの制限を `UNLIMITED` に設定します。

```
ulimit -c unlimited

ulimit -c
unlimited
```

- サービスにコア制限を設定する場合、`/lib/systemd/system/edb-as@.service` に以下の設定を追加します。

```
[Service]
LimitCore=Infinity
```

- サービス構成をリロードします。

```
systemctl daemon-reload
```

- `systemd` を使用してグローバルなデフォルト制限を変更し、次の設定を `/etc/systemd/system.conf` に追加します。

```
DefaultLimitCORE=Infinity
```

- systemd を再起動します。

```
systemctl daemon-reexec
```

- EDB Postgres Advanced Server を停止してから起動します。

```
systemctl stop edb-as@15.service  
systemctl start edb-as@15.service
```

- これで、コアダンプが有効になりました。次のコマンドを使用して gdb ツールとデバッグシンボルをインストールします。

```
apt-get install gdb  
apt-get install edb-as15 edb-as-contrib edb-as15-server edb-debugger-dbgsym
```

- コアダンプファイルへのパスを置き換えてから、bt コマンドを使用してバックトレースを分析します。

```
gdb /usr/lib/edb-as/15/bin /var/coredumps/core-edb-postgres-21638  
(gdb) bt full
```

注釈:

- Debian または Ubuntu ホスト上のデバッグ情報パッケージ名はさまざまで、`-dbgsym` または `-dbg`` サフィックスが含まれています。 `sources.list` の設定とデバッグ情報パッケージのインストールの詳細については、Debian または Ubuntu wiki (<https://wiki.debian.org/HowToGetABacktrace>) または <https://wiki.ubuntu.com/Debug%20Symbol%20Packages>) にアクセスしてください。それぞれ、- コアファイルはメモリ使用量によっては巨大になる場合があるため、システムでコアダンプを有効にすると、時間の経過とともに大容量ストレージがいっぱいになる場合があります。
-

第2章 Windows installation troubleshooting

2.1 Java ベースのアプリケーションの表示が難しい

Java ベースのサーバー機能（コントロールまたはテキストが正しく表示されない、または空白のウィンドウ）の表示に問題が発生した場合、最新の `libxcb-xlib` ライブラリにアップグレードすると、ほとんどのディストリビューションで問題が修正されるはずです。その他の回避策については、

[Java bug database](#) を参照してください。

- `-mode` 無人認証エラー
-

無人インストール中のコンポーネントモジュールからの認証エラーは、指定された `--servicepassword` または `--superpassword` の値が正しくないことを示している場合があります。

2.2 EDB Postgres Advanced Server のインストール中のエラー

インストールプロセス中にエラーが発生した場合は、インストールを終了し、Windows のバージョンが最新であることを確認してください。未処理のオペレーティングシステムの更新を適用した後、EDB Postgres Advanced Server インストーラーを再度呼び出します。

2.3 既存のデータディレクトリの内容が原因でインストールが完了しない

データディレクトリに既存のコンテンツが原因でインストールが完了しない場合、サーバーはサーバーログにエラーメッセージを書き込みます。

`A data directory is neither empty, or a recognisable data directory.`

同様のメッセージが表示された場合は、データディレクトリが空であることを確認します。ファイル（システム生成の `lost+found` フォルダーを含む）が存在するため、インストールが完了しません。インストーラーを再呼び出ししてインストールを完了する前に、データディレクトリからファイルを削除するか、データディレクトリに別の場所を指定します。