



EDB

Trouble Shooting

Release 15.2.0

EnterpriseDB Corporation

Mar 03, 2023

CONTENTS

1	Linux installation troubleshootig	3
1.1	Installation troubleshooting	3
1.1.1	Difficulty displaying Java-based applications	3
1.1.2	The installation fails to complete eue to existing data directory contents	3
1.1.3	Difficulty installing the EPEL release package	3
1.2	Enabling core dumps	4
1.2.1	Enabling core dumps on a RHEL or CentOS or Rocky Linux or AlmaLinux host	4
1.2.2	Enabling core dumps on a Debian or Ubuntu host	6
2	Windows installation troubleshooting	9
2.1	Difficulty displaying Java-based applications	9
2.2	Errors during an EDB Postgres Advanced Server installation	9
2.3	The installation fails to complete due to existing data directory contents	9

This section provides detailed information about:

- Troubleshooting installations and enabling core dumps on the Linux platforms.
- Troubleshooting installations on Windows platform.

LINUX INSTALLATION TROUBLESHOOTING

This section provides detailed information about:

- Troubleshooting installations on the Linux platforms.
- Enabling core dumps on the Linux platforms.

1.1 Installation troubleshooting

1.1.1 Difficulty displaying Java-based applications

If you encounter difficulty displaying Java-based server features (controls or text not being displayed correctly, or blank windows), upgrading to the latest `libxcb-xlib` libraries should correct the problem on most distributions. See the [Java bug database](#) for other possible workarounds.

1.1.2 The installation fails to complete due to existing data directory contents

If an installation fails to complete due to an existing content in the data directory, the server writes an error message to the server logs:

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

If you encounter a similar message, you should confirm that the data directory is empty; the presence of files (including the system-generated `lost+found` folder) prevents the installation from completing. Either remove the files from the data directory, or specify a different location for the data directory before re-invoking the installer to complete the installation.

1.1.3 Difficulty installing the EPEL release package

If you encounter difficulty when installing the EPEL release package, you can use the following command to install the `epel-release` package on RHEL/CentOS 7.x and RHEL/Rocky Linux/AlmaLinux 8.x:

```
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
```

You might need to enable the `[extras]` repository definition in the `CentOS-Base.repo` file (located in `/etc/yum.repos.d`). If `yum` cannot access a repository that contains `epel-release`, an error message appears:

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

If you receive this error, you can download the EPEL rpm package, and install it manually. To manually install EPEL, download the rpm package, assume superuser privileges, navigate into the directory that contains the package, and install EPEL with the command:

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

1.2 Enabling core dumps

You can use core dumps to diagnose or debug errors. A core dump is a file containing a process's address space (memory) when the process terminates unexpectedly. Core dumps may be produced on-demand (such as by a debugger) or automatically upon termination.

1.2.1 Enabling core dumps on a RHEL or CentOS or Rocky Linux or AlmaLinux host

On RHEL/CentOS 7.x or RHEL/Rocky Linux/AlmaLinux 8.x, core file creation is disabled by default. To enable the core file generation, follow the following commands:

- Identify the system's current limit using the `ulimit -c` or `ulimit -a` command. `0` indicates that core file generation is disabled.

```
# 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  
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
```

- Create a new directory to store the core dumps and modify `kernel.core_pattern` to store the dumps in a specified directory:

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

(continues on next page)

(continued from previous page)

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

- Use the following command to persist the `kernel.core_pattern` setting across reboots:

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

- Enable core dumps in `/etc/security/limits.conf` to allow a user to create core files. Each line describes a limit for a user in the following form:

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

Use `*` to enable the core dump size to unlimited.

- Set the limit of core file size to `UNLIMITED` by using the following command:

```
ulimit -c unlimited

ulimit -c
unlimited
```

- To set a core limit for the services, add the following setting in `/usr/lib/systemd/system/edb-as-15.service`:

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

- Reload the service configuration:

```
systemctl daemon-reload
```

- Modify the global default limit using `systemd`, add the following setting in `/etc/systemd/system.conf`:

```
DefaultLimitCORE=Infinity
```

- Restart the `systemd`:

```
systemctl daemon-reexec
```

- Stop and then start EDB Postgres Advanced Server:

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

- Now, the core dumps are enabled, install the `gdb` tool and debug packages using the following command:

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

- Replace the path to a core dump file before proceeding to get a backtrace using the `bt` command to analyze output:

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

1.2.2 Enabling core dumps on a Debian or Ubuntu host

On Debian 10 or Ubuntu 18, 20, core file creation is disabled by default. To enable the core file generation, follow the following commands:

- Identify the system's current limit using the `ulimit -c` or `ulimit -a` command. `0` indicates that core file generation is disabled.

```
# 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
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
```

- Create a new directory to store the core dumps and modify `kernel.core_pattern` to store the dumps in a specified directory:

```
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
```

- Use the following command to persist the `kernel.core_pattern` setting across reboots:

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

- Enable core dumps in `/etc/security/limits.conf` to allow a user to create core files. Each line describes a limit for a user in the following form:

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

Use `*` to enable the core dump size to unlimited.

- Set the limit of core file size to `UNLIMITED` by using the following command:

```
ulimit -c unlimited
```

```
ulimit -c
unlimited
```

- To set a core limit for the services, add the following setting in `/lib/systemd/system/edb-as@.service`.

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

- Reload the service configuration:

```
systemctl daemon-reload
```

- Modify the global default limit using `systemd`, add the following setting in `/etc/systemd/system.conf`.

```
DefaultLimitCORE=Infinity
```

- Restart the `systemd`:

```
systemctl daemon-reexec
```

- Stop and then start EDB Postgres Advanced Server:

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

- Now, the core dumps are enabled, install the `gdb` tool and debug symbols using the following command:

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

- Replace the path to a core dump file before proceeding to get a backtrace using the `bt` command to analyze output:

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

Note:

- The debug info packages name on a Debian or Ubuntu host may vary and include `-dbgsym` or `-dbg` suffix. For more information about setting `sources.list` and installing the debug info packages, visit Debian or Ubuntu wiki at <https://wiki.debian.org/HowToGetABacktrace> or <https://wiki.ubuntu.com/Debug%20Symbol%20Packages> respectively. - The core files can be huge depending on the memory usage, enabling the core dumps on a system may fill up its mass storage over time.

WINDOWS INSTALLATION TROUBLESHOOTING

2.1 Difficulty displaying Java-based applications

If you encounter difficulty displaying Java-based server features (controls or text not being displayed correctly, or blank windows), upgrading to the latest `libxcb-xlib` libraries should correct the problem on most distributions. See the [Java bug database](#) for other possible workarounds.

- `-mode unattended` authentication errors

Authentication errors from component modules during unattended installations may indicate that the specified values of `--servicepassword` or `--superpassword` may be incorrect.

2.2 Errors during an EDB Postgres Advanced Server installation

If you encounter an error during the installation process, exit the installation, and ensure that your version of Windows is up-to-date. After applying any outstanding operating system updates, re-invoke the EDB Postgres Advanced Server installer.

2.3 The installation fails to complete due to existing data directory contents

If an installation fails to complete due to existing content in the data directory, the server writes an error message to the server logs:

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

If you encounter a similar message, confirm that the data directory is empty; the presence of files (including the system-generated `lost+found` folder) prevents the installation from completing. Either remove the files from the data directory, or specify a different location for the data directory before re-invoking the installer to complete the installation.