Bacula Enterprise Troubleshooting

Bacula Systems Documentation

Contents

Contents

The following article aims at presenting information on how to troubleshoot Bacula Enterprise issues.

1 Backup Troubleshooting

I Didn't Realize that the Backups Were Not Working on My Client. What Should I Do?

You should be sending yourself an email message for each job. This will avoid the possibility of not knowing about a failed backup. To do so put something like:

```
Mail = yourname@yourdomain = all, !skipped
```

in your Director's message resource. You should then receive one email for each Job that ran. When you are comfortable with what is going on, you might change that to:

```
MailOnError = yourname@yourdomain = all, !skipped
```

then you only get email messages when a Job errors as is the case for your Windows machine.

2 Client Troubleshooting

I'm Getting Authorization Errors. What is Going On?

For security reasons, **Bacula** requires that both the File daemon and the Storage daemon know the name of the Director as well as its password. As a consequence, if you change the Director's name or password, you must make the corresponding change in the Storage daemon's and in the File daemon's configuration files.

During the authorization process, the Storage daemon and File daemon also require that the Director authenticates itself, so both ends require the other to have the correct name and password.

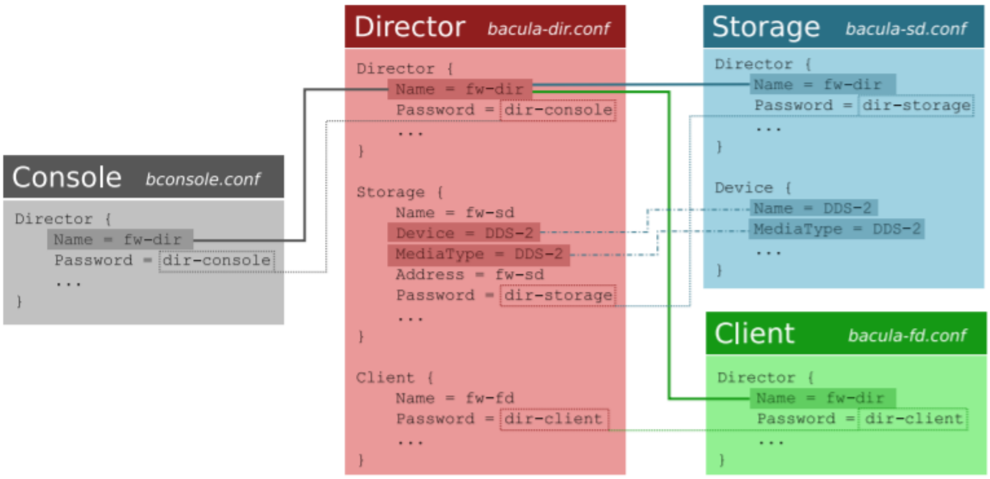
If you have edited the configuration files and modified any name or any password, and you are getting authentication errors, then your best bet is to go back to the original configuration files generated by the **Bacula** installation process. Make only the absolutely necessary modifications to these files – e.g. add the correct email address.

Another reason that you can get authentication errors is if you are running Multiple Concurrent Jobs in the Director, but you have not set them in the File daemon or the Storage daemon. Once you reach their limit, they will reject the connection producing authentication (or connection) errors.

If you are having problems connecting to a Windows machine that previously worked, you might try restarting the **Bacula** service.

Some users report that authentication fails if there is not a proper reverse DNS lookup entry for the machine. This seems to be a requirement of gethostbyname(), which is what **Bacula** uses to translate names into IP addresses. If you cannot add a reverse DNS entry, or you don't know how to do so, you can avoid the problem by specifying an IP address rather than a machine name in the appropriate **Bacula** conf file.

Here is a picture that indicates what names/passwords in which files/Resources must match up:



```
Director {
    Name = fw-dir
    Password = dir-console
    ...
}
```

```
Director {
    Name = fw-dir
    Password = dir-console
    ...
}

Storage {
    Name = fw-sd
    Device = DDS-2
    MediaType = DDS-2
    Address = fw-sd
    Password = dir-storage
    ...
}

Client {
    Name = fw-fd
    Password = dir-client
    ...
}
```

```
Director {
    Name = fw-dir
    Password = dir-storage
    ...
}

Device {
    Name = DDS-2
    MediaType = DDS-2
    ...
}
```

```
Director {
    Name = fw-dir
    Password = dir-client
    ...
}
```

Fig. 1: Configuration Diagram

In the center column, you will find the Director, Storage, and Client resources, with their names and passwords – these are all in **bacula-dir.conf**. The left column is where the corresponding values should be found in the Console and right column is where the corresponding values should be found in the Storage daemon (SD), and File daemon (FD) configuration files.

Another thing to check is to ensure that the **Bacula** component you are trying to access has **Maximum Concurrent Jobs** set large enough to handle each of the Jobs and the Console that want to connect simultaneously. Once the maximum connections has been reached, each **Bacula** component will reject all new connections.

Please also remember that File daemons with later versions than the Director and Storage daemons are not supported and can result in authorization errors.

Finally, make sure you have no **hosts.allow** or **hosts.deny** file that is not permitting access to the site trying to connect.

Bacula Runs Fine but Cannot Access a Client on a Different Machine. Why?

There are several reasons why **Bacula** could not contact a client on a different machine. They are:

- It is a Windows Client, and the client died because of an improper configuration file. Check that the **Bacula** icon is in the system tray and the the menu items work. If the client has died, the icon will disappear only when you move the mouse over the icon.
- The Client address or port is incorrect or not resolved by DNS. See if you can ping the client machine using the same address as in the Client record.
- You have a firewall, and it is blocking traffic on port 9102 between the Director's machine and the Client's machine (or on port 9103 between the Client and the Storage daemon machines).
- Your password or names are not correct in both the Director and the Client machine. Try configuring everything identical to how you run the client on the same machine as the Director, but just change the Address. If that works, make the other changes one step at a time until it works.
- You may also be having problems between your File daemon and your Storage daemon. The name you use in the Storage resource of your Director's conf file must be known (resolvable) by the File daemon, because it is passed symbolically to the File daemon, which then resolves it to get an IP address used to contact the Storage daemon.
- You may have a **hosts.allow** or **hosts.deny** file that is not permitting access.

I'm Having VSS Problems on Windows FD. What Should I do?

If you are experiencing problems such as `VSS` hanging on MSDE, first try running **vssadmin** to check for problems, then try running **ntbackup** which also uses `VSS` to see if it has similar problems. If so, you know that the problem is in your Windows machine and not with **Bacula**.

The `FD` hang problems were reported with **MSDEwriter** when:

- a local firewall locked local access to the `MSDE` TCP port (MSDEwriter seems to use `TCP`/IP and not Named Pipes).
- **msdtcs** was installed to run under "localsystem": try running **msdtcs** under networking account (instead of local system) (com+ seems to work better with this configuration).

I'm Having Restore Problems on Windows FD. What Should I do?

If during a Backup, you get the message: **ERR=Access is denied** and you are using the portable option, you should try both adding both the non-portable (backup API) and the Volume Shadow Copy options to your Director's conf file.

In the Options resource:

```
portable = no
```

In the Fileset resource:

```
EnableVSS = yes
```

In general, specifying these two options should allow you to backup any file on a Windows system. However, in some cases, if users have allowed to have full control of their folders, even system programs such a **Bacula** can be locked out. In this case, you must identify which folders or files are creating the problem and do the following:

1. Grant ownership of the file/folder to the Administrators group, with the option to replace the owner on all child objects.
2. Grant full control permissions to the Administrators group, and change the user's group to only have Modify permission to the file/folder and all child objects.

I don't know if the FD on Windows opened the port and is listening. What to do?

If you want to see if the File Daemon has properly opened the port and is listening, you can enter the following command in a shell window:

```
netstat -an | findstr 910[123]
```

is another program that has been recommend, but it is not a standard Windows program, so you must find and download it from the Internet.

How to Fix the Windows Boot Record?

An effective way to restore a Windows backup for those who do not purchase the bare metal restore capability is to install Windows on a different hard drive and restore the backup. Then run the recovery CD and run:

```
diskpart
    select disk 0
    select part 1
    active
    exit
bootrec  /rebuldbcd
bootrec  /fixboot
bootrec  /fixmbr
```

Why Do I Have Slow Backup on Windows Machines?

Sometimes Windows machines the File Daemon may have very slow backup transfer rates compared to other machines. To you might try setting the Maximum Network Buffer Size to 32,768 in both the File daemon and in the Storage daemon. The default size is larger, and apparently some Windows ethernet controllers do not deal with a larger network buffer size.

Many Windows ethernet drivers have a tendency to either run slowly due to old broken firmware, or because they are running in half-duplex mode. Please check with the ethernet card manufacturer for the latest firmware and use whatever techniques are necessary to ensure that the card is running in duplex.

I’m Experiencing Problems with Opening the Files (Windows). What to do?

If you are not using the portable option, and you have VSS (Volume Shadow Copy) enabled in the Director, and you experience problems with **Bacula** not being able to open files, it is most likely that you are running an antivirus program that blocks **Bacula** from doing certain operations. In this case, disable the antivirus program and try another backup. If it succeeds, either get a different (better) antivirus program or use something like RunClientJobBefore/After to turn off the antivirus program while the backup is running.

If turning off anti-virus software does not resolve your VSS problems, you might have to turn on VSS debugging. The following link describes how to do this: support.microsoft.com/kb/887013/en-us.

In Microsoft Windows Small Business Server 2003 the VSS Writer for Exchange is turned off by default. To turn it on, see the following link: support.microsoft.com/default.aspx?scid=kb;EN-US;Q838183.

The most likely source of problems is authentication when the Director attempts to connect to the File Daemon that you installed. This can occur if the names and the passwords defined in the File Daemon’s configuration file **bacula-fd.conf** file on the Windows machine do not match with the names and the passwords in the Director’s configuration file **bacula-dir.conf** located on your Unix/Linux server.

More specifically, the password found in the **Client** resource in the Director’s configuration file must be the same as the password in the **Director** resource of the File daemon’s configuration file. In addition, the name of the **Director** resource in the File daemon’s configuration file must be the same as the name in the **Director** resource of the Director’s configuration file.

It is a bit hard to explain in words, but if you understand that a Director normally has multiple Clients and a Client (or File Daemon) may permit access by multiple Directors, you can see that the names and the passwords on both sides must match for proper authentication.

One user had serious problems with the configuration file until he realized that the Unix end of line conventions were used and **Bacula** wanted them in Windows format. This has not been confirmed though, and **Bacula** version 2.0.0 and above should now accept all end of line conventions (Windows, Unix, Mac).

Running Unix like programs on Windows machines is a bit frustrating because the Windows command line shell (DOS Window) is rather primitive. As a consequence, it is not generally possible to see the debug information and certain error messages that **Bacula** prints. With a bit of work, however, it is possible. When everything else fails and you want to **see** what is going on, try the following:

```
Start a DOS shell Window.
c:Files-fd -t >out
type out
```

The precise path to bacula-fd depends on where it is installed. The **-t** option will cause **Bacula** to read the configuration file, print any error messages and then exit. the redirects the output to the file named **out**, which you can list with the **type** command.

If something is going wrong later, or you want to run **Bacula** with a debug option, you might try starting it as:

```
c:Files-fd -d 100 >out
```

In this case, **Bacula** will run until you explicitly stop it, which will give you a chance to connect to it from your Unix/Linux server. When you start the File daemon in debug mode it can write the output to a trace file **bacula.trace** in the current directory. To enable this, before running a job, use the console, and enter:

```
trace on
```

then run the job, and once you have terminated the File daemon, you will find the debug output in the **bacula.trace** file, which will probably be located in the same directory as **bacula-fd.exe**.

In addition, you should look in the System Applications log on the Control Panel to find any Windows errors that **Bacula** got during the startup process.

Finally, due to the above problems, when you turn on debugging, and specify **trace=1** on a **setdebug** command in the Console, **Bacula** will write the debug information to the file **bacula.trace** in the directory from which **Bacula** is executing.

My ClientRunBeforeJob Scripts are Randomly Dying (Windows). What Should I Do?

If you are having problems with ClientRunBeforeJob scripts randomly dying, it is possible that you have run into an Oracle bug. See bug number 622 in the **bugs.bacula.org** database. The following information has been provided by a user on this issue:

```
The information in this document applies to:
Oracle HTTP Server - Version: 9.0.4
Microsoft Windows Server 2003
Symptoms
When starting an OC4J instance, the System Clock runs faster, about 7
seconds per minute.
Cause
+ This is caused by the Sun JVM bug 4500388, which states that
↪"Calling
Thread.sleep() with a small argument affects the system clock".␣
↪Although
this is reported as fixed in JDK 1.4.0_02, several reports␣
↪contradict this
(see the bug in
http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=4500388).
+ Also reported by Microsoft as "The system clock may run fast when␣
↪you
use the ACPI power management timer as a high-resolution counter on␣
↪Windows
2000-based computers" (See http://support.microsoft.com/?id=821893)
```

You may wish to start the daemon with debug mode on rather than doing it using bconsole. To do so, edit the following registry key:

```
HKEY_LOCAL_MACHINE-dir
```

using regedit, then add **-dnn** after the /service option, where nn represents the debug level you want.

My Linux Client Immediately Dies When I Start It

The most common problem is either that the configuration file is not where it expects it to be, or that there is an error in the configuration file. The location of your configuration file should be `/opt/bacula/etc/` for Bacula Enterprise and `/etc/bacula/` by default for Bacula Community.

To **see** what is going on when the File daemon starts on Linux, do the following:

```
/opt/bacula/bin/bacula-fd -d100 -c /opt/bacula/etc/bacula-fd.conf
```

or

```
/etc/bacula/bin/bacula-fd -d100 -c /etc/bacula/bacula-fd.conf
```

This will cause the FD to write the debug output to the command line, which you can examine and thereby determine the problem.

3 Storage Troubleshooting

I Change Recycling, Retention Periods, or File Sizes in My Pool Resource and They Still Don't Work.

The different variables associated with a Pool are defined in the Pool Resource, but are actually read by **Bacula** from the Catalog database.

When **Bacula** creates a Media record (Volume), it uses many default values from the Pool record. If you subsequently change the Pool record, the new values will be used as a default for the next Volume that is created, but if you want the new values to apply to existing Volumes, you must manually update the Volume Catalog entry using the `update volume` command in the Console program.

I Have Configured Compression On, But None of My Files Are Compressed. Why?

There are two kinds of compression. One is tape compression. This is done by the tape drive hardware, and you either enable or disable it with system tools such as `mt`. This compression works independently of **Bacula**, and when it is enabled, you should not use the **Bacula** software compression.

Bacula also has software compression code in the File daemons, which you normally need to enable only when backing up to file Volumes.

Bacula is Asking for a New Tape After 2 GB of Data but My Tape Holds 33 GB. Why?

There are several reasons why **Bacula** will request a new tape.

- There is an I/O error on the tape. **Bacula** prints an error message and requests a new tape. **Bacula** does not attempt to continue writing after an I/O error.
- **Bacula** encounters and end of medium on the tape. This is not always distinguishable from an I/O error.
- You have specifically set some size limitation on the tape. For example the **Maximum Volume Bytes** or **Maximum Volume Files** in the Director's Pool resource, or **Maximum Volume Size** in the Storage daemon's Device resource.

Why Does Bacula Ignore the MaxVolumeSize Set in My Pool?

The MaxVolumeSize that **Bacula** uses comes from the Media record, so most likely you changed your Pool, which is used as the default for creating Media records, **after** you created your Volume. Check what is in the Media record by doing:

```
llist Volume=xxx
```

If it doesn't have the right value, you can use:

```
update Volume=xxx
```

to change it.

How Do I Test Wild-card Expressions against a File of Data?

bwild is a simple program that will allow you to test wild-card expressions against a file of data.

bwild is found in the src/tools directory and it is normally installed with your system binaries. To run it, use:

```
Usage: bwild [-d debug_level] -f <data-file>
    -f specify file of data to be matched
    -l suppress line numbers
    -n print lines that do not match
    -? print this message.
```

The <data-file> is a filename that contains lines of data to be matched (or not) against one or more patterns. When the program is run, it will prompt you for a wild-card pattern, then apply it one line at a time against the data in the file. Each line that matches will be printed preceded by its line number. You will then be prompted again for another pattern.

Enter an empty line for a pattern to terminate the program. You can print only lines that do not match by using the -n option, and you can suppress printing of line numbers with the -l option.

This program can be useful for testing wild expressions to be applied against a list of filenames.

How Do I Test Wild-card Regular against a File of Data?

bregex is a simple program that will allow you to test regular expressions against a file of data. This can be useful because the regex libraries on most systems differ, and in addition, regex expressions can be complicated.

bregex is found in the src/tools directory and it is normally installed with your system binaries. To run it, use:

```
Usage: bregex [-d debug_level] -f <data-file>
    -f specify file of data to be matched
    -l suppress line numbers
    -n print lines that do not match
    -? print this message.
```

The <data-file> is a filename that contains lines of data to be matched (or not) against one or more patterns. When the program is run, it will prompt you for a regular expression pattern, then apply it one line at a time against the data in the file. Each line that matches will be printed preceded by its line number. You will then be prompted again for another pattern.

Enter an empty line for a pattern to terminate the program. You can print only lines that do not match by using the -n option, and you can suppress printing of line numbers with the -l option.

This program can be useful for testing regex expressions to be applied against a list of filenames.

4 Debugging

How to Run Bacula Under the Debugger Manually?

If for some reason you cannot get the automatic traceback, or if you want to interactively examine the variable contents after a crash, you can run **Bacula** under the debugger. Assuming you want to run the Storage daemon under the debugger (the technique is the same for the other daemons, only the name changes), you would do the following:

1. Start the Director and the File daemon. If the Storage daemon also starts, you will need to find its PID (`ps fax | grep bacula-sd`) and kill it with a command like the following:

   ```
   kill -15 PID
   ```

   where you replace **PID** by the actual value.

2. At this point, the Director and the File daemon should be running but the Storage daemon should not.

3. `cd` to the directory containing the Storage daemon

4. Start the Storage daemon under the debugger:

   ```
   gdb ./bacula-sd
   ```

5. Run the Storage daemon:

   ```
   run -s -f -c ./bacula-sd.conf
   ```

   You may replace the **./bacula-sd.conf** with the full path to the Storage daemon's configuration file.

6. At this point, **Bacula** will be fully operational.

7. In another shell command window, start the Console program and do what is necessary to cause **Bacula** to die.

8. When **Bacula** crashes, the `gdb` shell window will become active and `gdb` will show you the error that occurred.

9. To get a general traceback of all threads, issue the following command:

   ```
   thread apply all bt
   ```

After that you can issue any debugging command.

How to Get Debug Output from Bacula?

Each of the daemons normally has debug compiled into the program, but disabled. There are two ways to enable the debug output. One is to add the **-d nnn** option on the command line when starting the debugger. The **nnn** is the debug level, and generally anything between 50 and 200 is reasonable. The higher the number, the more output is produced. The output is written to standard output.

The second way of getting debug output is to dynamically turn it on using the Console using the `setdebug` command. The full syntax of the command is:

```
setdebug level=nnn client=client-name storage=storage-name dir
```

If none of the options are given, the command will prompt you. You can selectively turn on/off debugging in any or all the daemons (i.e. it is not necessary to specify all the components of the above command).

5 Bacula Error and Event Code List

The following chapter aims at presenting the reader with Bacula error and event code list.

The Bacula error and event codes use the format XYZZZZ, where:

- X is the component code (D for Director, S for Storage Daemon, F for File Daemon, C for Console)
- Y is the code level (F for fatal, E for error, A for Abort, W for warning, I for information, D for Daemon message, U for catalog update, S for security, V for volume operations, C for Console and J is for Job)
- ZZZZ is a 4 digit code

5.1 Director - Bacula Error and Event Code List

Director Codes

DC0001
Purge client files event.

DD0001
Director startup event.

DJ0001
Job creation event.

DU0001
Update of the Bacula catalog schema event.

DS0001
Authentication failed event from a Console.

DV0001
Volume purged event.

DC0002
Purge job event.

DD0002
Director shutdown event.

DJ0002
Job termination event.

DC0003
Purge volume event.

DD0003
Director cleanup event during startup. Jobs marked as Running or Created will be marked as failed during the Director startup.

DJ0003
Resume job event.

DC0004
Truncate volume event.

DD0004
Director configuration reload error event.

DJ0004
Run job command event.

DC0005
Cancel job event.

DD0005
Director configuration reload event.

DC0006
Disable/Enable Batch mode event.

DC0007
Disable/Enable Job event.

DC0008
Disable/Enable Client event.

DC0009
Disable/Enable Schedule event.

DC0010
Delete volume command event.

DC0011
Delete pool command event.

DC0012
Delete client command event.

DC0013
Storage daemon command event.

DC0014
Storage daemon command event.

DC0015
Connection from a Console event.

DC0016
Disconnection from a Console event.

DC0017
Console event sent from a script.

DC0018
delete object command event.

DC0019
Reload command event.

DE0020
Password or resource name mismatch between the two components.

DE0021
No authorization for the given resource. Check the Console ACLs or Director configuration.

DE0028
The Director got a SQL Error. This error cannot be recovered.

DE0029
The Director cannot connect to a remote FileDaemon resource. Check that the remote FileDaemon is started and can be reached via the network.

DE0030
No socket found for this client. The client uses the ConnectToDirector feature and the Director did not found an open socket to access the client. Make sure the Client is started and can access to the Director.

DE0031
Protocol error between component. This error cannot be recovered. Check the version and the TLS settings of each components.

DE0037
Incorrect configuration, such as filedaemon not defined for a Job.

DE0038
Incorrect dynamic object/library/plugin found.

DE0039
The Director cannot connect to a remote Storage resource. Check that the remote Storage Daemon is started and can be reached via the network.

DI0050
No Malware found in given Job.

DE0051
Unable to update the Malware Database.

DI0052
No file to scan for Malware.

DE0053
Unable to open the Malware Database export.

DE0054
Unable to find a checksum for the JobIds. Use Signature = MD5/SHA256 Fileset option.

DE0055
Unable to find a valid checksum database for JobIds.

DE0056
Found Malware(s) on JobIds. Run the command "list files type=malware jobid=<jobid>" to list the files infected by malware(s).

DE0060
Network connection error. Unable to resolve hostname via gethostbyname.

DE0061
Network connection error. Socket open error. Check the TCP/IP configuration.

DE0062
Unable to find the source address to connect a remote host. Check that the interface set as source exists on the system.

DE0063
Network socket configuration error. Check the TCP/IP configuration.

DW0063
Unable to set socket option such as SO_KEEPALIVE or SO_KEEPIDLE on the socket.

DE0064
TLS Error detected. The connect/accept operation cannot be done, or the password is incorrect.

DE0065
Authentication error. Password or resource name mismatch between the two components.

DE0066
TLS initialization error.

DE0067
TLS Negotiation failed. Check the version and the TLS settings of each components.

DE0068
TLS certificate verification error.

DE0069
TLS configuration error.

DE0071
Unable to start TLS protocol. Check the version and the TLS settings of each components.

DE0072
TLS shutdown error.

DD0080
Callhome unsuccessful event.

DD0081
Callhome contract issue detected event.

DW0081
Callhome contract issue detected message.

DD0082
Callhome successful event.

5.2 Storage Daemon - Bacula Error and Event Code List

Storage Daemon Codes

SS0001
Authentication failed.

SD0001
Storage daemon startup event.

SD0002
Storage daemon device initialization event.

SJ0001
Storage Daemon Start of Job event.

SJ0002
Storage Daemon End of a Job event.

SD0003
Storage daemon device shutdown event.

SJ0003
Mark volume as immutable or read-only event.

SE0009
The Storage Daemon cannot connect to a remote resource. Check that the remote service is started and can be reached via the network.

SW0017
Incorrect configuration, such as incorrect Maximum Block Size on a Device.

SE0017
Incorrect configuration error, such as a miss-configuration of the device.

SF0017
Incorrect configuration fatal error, such as invalid device type.

SA0017
Incorrect configuration fatal error, such as Plugin Directory not defined.

SA0020
Unable to initialize kernel objects such as mutex, conditional variable.

SE0029
The Storage Daemon cannot connect to a remote FileDaemon resource. Check that the remote FileDaemon is started and can be reached via the network.

SF0031
Protocol error between component. This error cannot be recovered. Check the version and the TLS settings of each components.

SA0038
Incorrect dynamic object.

SE0038
Lookup of symbol "BaculaSDdriver" in driver for device failed.

SE0039
The Storage Daemon cannot connect to a remote Storage resource. Check that the remote Storage Daemon is started and can be reached via the network.

SE0060
Network connection error. Unable to resolve hostname via gethostbyname.

SE0061
Network connection error. Socket open error. Check the TCP/IP configuration.

SE0062
Unable to find the source address to connect a remote host. Check that the interface set as source exists on the system.

SE0063
Network socket configuration error. Check the TCP/IP configuration.

SW0063
Unable to set socket option such as SO_KEEPALIVE or SO_KEEPIDLE on the socket.

SE0064
TLS Error detected. The connect/accept operation cannot be done, or the password is incorrect.

SE0065
Authentication error. Password or resource name mismatch between the two components.

SF0065
Director Authentication error. Check the password on both Storage and Director side.

SE0066
TLS initialization error.

SE0067
TLS Negotiation failed. Check the version and the TLS settings of each components.

SE0068
TLS certificate verification error.

SE0069
TLS configuration error.

SE0071
Unable to start TLS protocol. Check the version and the TLS settings of each components.

SE0072
TLS shutdown error.

SW0106
Incorrect device configuration.

SW0108
Incorrect changer configuration.

SA0201
Write block error.

SE0201
Write error on device.

SF0201
Write error on device. Usually found after SE0201 errors.

SW0201
Free space caused end of the volume.

SE0202
Block buffer size looping problem.

SF0202
Error writing final JobMedia record.

SI0202
End of volume message.

SE0203
Volume label error.

SF0203
Write/read error on disabled device.

SE0204
Read error on device.

SF0204
Write error on device at EOM.

SE0205
Short block found on device, block discarded.

SF0205
Attempt to write on read-only volume.

SE0206
Invalid block length found.

SF0206
Attemp to write/read on closed device.

SE0207
Invalide block length found for tape device.

SF0207
Seek error on adata volume.

SE0208
Volume data error after a short block. block discarded.

5.3 File Daemon - Bacula Error and Event Code List

File Daemon/Client Error Codes

FC0001
Director disconnection event.

FC0002
Director connection event.

FC0003
Cancel command event.

FS0001
Authentication failure event.

FD0001
Start of the File Daemon event.

FD0002
Shutdown of the File Daemon event.

FJ0001
Start of Job event.

FJ0002
End of Job event.

FE0009
The Client cannot connect to a remote resource. Check that the remote service is started and can be reached via the network.

FE0019
The FileDaemon cannot connect to a remote Director resource. Check that the remote Director Daemon is started and can be reached via the network.

FE0027
Connection of a Director/Console not allowed on the Client.

FE0031
Protocol error between component. This error cannot be recovered. Check the version and the TLS settings of each components.

FF0037

Incorrect Plugin Option for Job requested. The Plugin specified is not loaded on the FileDaemon, check that the plugin is correctly installed and listed in “status client” output.

FE0039

The FileDaemon cannot connect to a remote Storage resource. Check that the remote Storage Daemon is started and can be reached via the network.

FE0060

Network connection error. Unable to resolve hostname via gethostbyname.

FE0061

Network connection error. Socket open error. Check the TCP/IP configuration.

FE0062

Unable to find the source address to connect a remote host. Check that the interface set as source exists on the system.

FE0063

Network socket configuration error. Check the TCP/IP configuration.

FW0063

Unable to set socket option such as SO_KEEPALIVE or SO_KEEPIDLE on the socket.

FE0064

TLS Error detected. The connect/accept operation cannot be done, or the password is incorrect.

FE0065

Authentication error. Password or resource name mismatch between the two components.

FF0065

Director Authentication error. Check the password on both Storage and Director side.

FE0066

TLS initialization error.

FE0067

TLS Negotiation failed. Check the version and the TLS settings of each components.

FE0068

TLS certificate verification error.

FE0069

TLS configuration error.

FE0071

Unable to start TLS protocol. Check the version and the TLS settings of each components.

FE0072

TLS shutdown error.

5.4 Console - Bacula Error and Event Code List

Console Error Codes

CE0029

The Console cannot connect to a remote resource. Check that the remote service is started and can be reached via the network.

6 What to Do When Bacula Crashes

If you are running on a Linux system, and you have a set of working configuration files, it is very unlikely that **Bacula** will crash. As with all software, however, it is inevitable that someday, it may crash, particularly if you are running on another operating system or using a new or unusual feature.

This chapter explains what you should do if one of the three **Bacula** daemons (Director, File, Storage) crashes. When we speak of crashing, we mean that the daemon terminates abnormally because of an error. There are many cases where **Bacula** detects errors (such as PIPE errors) and will fail a job. These are not considered crashes. In addition, under certain conditions, **Bacula** will detect a fatal in the configuration, such as lack of permission to read/write the working directory. In that case, **Bacula** will force itself to crash with a SEGFAULT. However, before crashing, **Bacula** will normally display a message indicating why. For more details, please read on.

6.1 Traceback

Each of the three **Bacula** daemons has a built-in exception handler which, in case of an error, will attempt to produce a traceback. If successful the traceback can even be emailed to you.

For this to work, you need to ensure that a few things are setup correctly on your system:

1. You must have an installed copy of `gdb` (the `GNU` debugger), and it must be on **Bacula's** path. On some systems such as Solaris, `gdb` may be replaced by `dbx`.
2. The **Bacula** installed script file `btraceback` must be in the same directory as the daemon which dies, and it must be marked as executable.
3. The script file **btraceback.gdb** must have the correct path to it specified in the `btraceback` file.
4. You must have a `mail` program which is on **Bacula's** path. By default, this `mail` program is set to `bsmtp`, and it must be correctly configured in case you would prefer to receive the traceback report via email.
5. If you run either the Director or Storage daemon under a non-root userid, you will most likely need to modify the `btraceback` file to do something like `sudo` (raise to root priority) for the call to `gdb` so that it has the proper permissions to debug **Bacula**.

If all the above conditions are met, the daemon that crashes will produce a traceback report and place it in the working directory of the failed daemon, and, if set correctly, also sent it via email. If the above conditions are not true, you can either run the debugger by hand as described below, or you may be able to correct the problems by editing the `btraceback` file.

Since each daemon has the same traceback code, a single btraceback file is sufficient if you are running more than one daemon on a machine.

6.2 Testing the Traceback

To “manually” test the traceback feature, you simply start **Bacula** then obtain the **PID** of the main daemon thread (there are multiple threads). The output produced here will look different depending on what OS and what version of the kernel you are running. Unfortunately, the output had to be split to fit on this page:

```
[kern@rufus kern]$ ps fax --columns 132 | grep bacula-dir
2103 ? S 0:00 /opt/bacula/bin/bacula-dir -c
                /opt/bacula/etc/bacula-dir.conf
2104 ? S 0:00 \_ /opt/bacula/bin/bacula-dir -c
                /opt/bacula/etc/bacula-dir.conf
2106 ? S 0:00 \_ /opt/bacula/bin/bacula-dir -c
                /opt/bacula/etc/bacula-dir.conf
2105 ? S 0:00 \_ /opt/bacula/bin/bacula-dir -c
                /opt/bacula/etc/bacula-dir.conf
```

The **PID** of the main daemon thread in this case is 2103. While **Bacula** is running, you call the program giving it the path to the **Bacula** executable, the **PID** and the working directory. In this case, it is:

```
./btraceback /opt/bacula/bin 2103 /opt/bacula/working
```

It should produce a file named (in this case) `bacula.2103.traceback` in the working directory and an email showing you the current state of the daemon (in this case the Director), and then exit leaving **Bacula** running as if nothing happened. If this is not the case, you will need to correct the problem by modifying the `btraceback` script.

Typical problems might be that `gdb` or `dbx` for Solaris is not on the default path. Fix this by specifying the full path to it in the `btraceback` file. Another common problem is that you haven’t modified the script so that the `bsmtp` program has an appropriate smtp server or the proper syntax for your `SMTP` server. If you use the `mail` program and it is not on the default path, it will also fail. On some systems, it is preferable to use `Mail` rather than `mail`.

6.3 Getting Traceback on Other Systems

It should be possible to produce a similar traceback on systems other than Linux, either using `gdb` or some other debugger. Solaris with `dbx` loaded works quite fine. On other systems, you will need to modify the `btraceback` program to invoke the correct debugger, and possibly correct the **btraceback.gdb** script to have appropriate commands for your debugger.