



DB2 Plugin

Bacula Systems Documentation

Contents

1	Executive summary	3
2	DB2	3
2.1	Architecture	3
2.2	Features	3
2.3	Installation of the Plugin	3
2.4	Configuration	4
2.5	Backup	8
2.6	Restore	14
2.7	Limitations	21
2.8	Troubleshooting	21

Contents

Note
You can download this article as a [PDF](#)

Enterprise

Bacula Enterprise Only
This solution is **only** available for Bacula Enterprise. For subscription inquiries, please reach out to sales@baculasystems.com.

- *Executive summary*
- *DB2*
 - *Architecture*
 - *Features*
 - *Installation of the Plugin*
 - *Configuration*
 - *Backup*
 - *Restore*
 - *Limitations*
 - *Troubleshooting*

1 Executive summary

IT organizations are constantly being challenged to deliver high quality solutions with reduced total cost of ownership.

This document is intended to provide insight into the considerations and processes required to successfully implement DB2 backup technique.

2 DB2

2.1 Architecture

The Bacula Enterprise DB2 Plugin relies on the extensive DB2 backup and restore API.

2.2 Features

The Bacula Enterprise DB2 Plugin can :

- use a named pipe to transfer data
- list and backup all databases from an instance
- select the databases to backup
- handle ONLINE/OFFLINE backups automatically
- detect if a database can support incremental and differential backup level
- stop/start automatically a database to do an OFFLINE backup
- backup a database executed via Docker
- automatically backup the database schema
- restore a database backup into a new database automatically
- ease the incremental restore procedure
- keep track of the backup timestamp
- control all steps and report errors to the job log
- restore a database dump file into a local directory for manual restore

2.3 Installation of the Plugin

On the Bacula File Daemon that you want to use, extend the repository file for your package manager to contain a section for the plugin. For example in RHEL, `/etc/yum.repos.d/bacula.repo`:

```
[Bacula]
name=Bacula Enterprise
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/bin/@version@/
↪rhel7-64/
enabled=1
protect=0
gpgcheck=0

[Bacula EnterpriseDB2Plugin]
name=Bacula Enterprise DB2 Plugin
baseurl=https://www.baculasystems.com/dl/@customer-string@/rpms/db2/@version@/
```

(continues on next page)

(continued from previous page)

```
↪rhel7-64/  
enabled=1  
protect=0  
gpgcheck=0
```

or in Debian Stretch, `/etc/apt/sources.list.d/bacula.list`:

```
#Bacula Enterprise  
deb https://www.baculasystems.com/dl/@customer-string@/debs/bin/@version@/  
↪jessie-64/ stretch main  
deb https://www.baculasystems.com/dl/@customer-string@/debs/db2/@version@/  
↪jessie-64/ stretch db2
```

Then perform a `yum update` or `apt-get update`, and after that the package `bacula-enterprise-db2-plugin` can be installed with `yum install` or `apt-get install`.

If you prefer to manually install the packages, you can also download them from your download area, and use one of the low level package manager tools (`rpm` or `dpkg`) to do the plugin installation.

2.4 Configuration

Plugin Command Line Parameters

The following options can be used in the Fileset Plugin definition:

- **can_stop** If the database configuration doesn't permit online backups, the plugin can stop the target database for the backup. If this parameter is not set, the database will not be backed up and an error will be raised in the Job.
- **use_sudo** Use the `sudo` command to prefix all `db2` commands. Used in conjunction with `unix_user`.
- **unix_user=<str>** Unix user to use to run `db2` commands. By default commands are executed under the Bacula FileDaemon service account (`root`) or set to `instance` parameter if used. Ex: `unix_user=db2inst1`
- **abort_on_error** Abort immediately the job if a serious error is found. By default, a Job error is raised, but other files and databases are backed up.
- **timeout=<int>** Number of seconds used to run various commands or wait to connect DB2 during a backup. The default value is 120 seconds. Ex: `timeout=5mins`
- **ctl_dir=<path>** Directory accessible to both the DB2 service and the Bacula File Daemon service. The Bacula DB2 Plugin will create command files inside this directory, and the DB2 instance user will execute them. The DB2 instance user must be able to read and create files inside this directory. The default value is `/tmp` and must be adapted to your configuration for security reasons. Ex: `ctl_dir=/data/backup`

```
# mkdir /data/backup  
# chmod 700 /data/backup  
# chown db2inst1 /data/backup
```

- **user_config_dir=<path>** Directory used to store local customization files for backup and restore procedures. (See ref:*custom-script* for more information)
- **bin_dir=<str>** String that is added to all `DB2` commands. It can be a directory outside the `PATH`, or some specific options such as a `docker` command. Ex: `bin_dir=/db2/config`

- **user=<str>** DB2 user account used in db2 commands.
- **password=<str>** DB2 user password used in db2 commands.
- **instance=<str>** DB2 instance name. This string is compared to the DB2 instance that will be used in the backup/restore procedure. If the instance doesn't match the instance found on the system, the plugin command will be skipped or the job will be aborted. Ex: `instance=db2inst1`
- **database=<glob>** DB2 database glob pattern to include in the backup. This glob expression is used to filter the database list found on the DB2 server. If the parameter is set and the selected database list is empty, an error will be raised. Ex: `database=prod*`

DB2 Docker Configuration

When using Docker DB2 package, the database directory must be shared with the main system, and Bacula can use this directory to communicate with DB2.

For example, if your Docker image was started with the following command:

```
mkdir -p /data/ctl
docker run -itd --name mydb2 -e DBNAME=testdb -v /data:/data \
  -e DB2INST1_PASSWORD=XXX -e LICENSE=accept -p 50000:50000 \
  --privileged=true ibmcom/db2
```

The Fileset will look like:

```
Plugin = "db2: bin_dir=\\\"docker exec mydb2 su - db2inst1 -c \\\" ctl_dir=/
↪data/ctl"
```

Where mydb2 is the DB2 docker container name, db2inst1 is the DB2 unix user inside the Docker container and /data is the volume shared between the host system and the docker container.

Plugin Restore Options

- **database** New database name. Take precedence over where parameter
- **user** Username
- **password** Password
- **unix_user** Unix user to use
- **instance** Instance where to restore
- **restore_options** Restore options passed to db2 restore command
- **replace_existing** Add the REPLACE EXISTING parameter to the restore command
- **rollforward_options** Rollforward options passed to db2 rollforward command
- **logtarget** LOGTARGET option to use when restoring the final image. Ex: `'/tmp/'`

Plugin Configuration File

During the Backup and the Restore operations, the Bacula FileDaemon DB2 Plugin will generate a set of dynamic scripts on disk to call the DB2 backup API.

example:

```

#!/bin/sh
export DB2INSTANCE=db2inst1

BDB2_BIN_DIR=""
BDB2_DATABASE="DB1"
BDB2_JOB_LEVEL=F
BDB2_FIFO="/data/tmp/fifo.5"
BDB2_INCLUDE_LOGS=""
BDB2_BACKUP_LEVEL=""
BDB2_STOP_DB=YES
BDB2_EXTRA_ARGS="WITHOUT PROMPTING"
BDB2_CMD='${BDB2_BIN_DIR}db2 -vst BACKUP DB "${BDB2_DATABASE}" ${BDB2_BACKUP_
↵LEVEL}
        to "${BDB2_FIFO}" ${BDB2_INCLUDE_LOGS} ${BDB2_EXTRA_ARGS}'
BDB2_VERSION=1

#####
if [ -f "/data/custom/db2_backup" ]
then
    . "/data/custom/db2_backup"
fi

if [ $BDB2_STOP_DB = YES ]; then
    echo Stopping the database...
    ${BDB2_BIN_DIR}db2 -v CONNECT TO "${BDB2_DATABASE}"
    ${BDB2_BIN_DIR}db2 -v LIST APPLICATION FOR DATABASE "${BDB2_DATABASE}"
    ${BDB2_BIN_DIR}db2 -v QUIESCE DATABASE IMMEDIATE FORCE CONNECTIONS
    ${BDB2_BIN_DIR}db2 TERMINATE
    ${BDB2_BIN_DIR}db2 -v DEACTIVATE DATABASE "${BDB2_DATABASE}"
fi
mkfifo "${BDB2_FIFO}"
eval ${BDB2_CMD}
ret=$?
rm -f "${BDB2_FIFO}"
if [ $BDB2_STOP_DB = YES ]; then
    echo Starting the database...
    ${BDB2_BIN_DIR}db2 -v ACTIVATE DATABASE "${BDB2_DATABASE}"
    ${BDB2_BIN_DIR}db2 -v CONNECT TO "${BDB2_DATABASE}"
    ${BDB2_BIN_DIR}db2 -v UNQUIESCE DATABASE
    ${BDB2_BIN_DIR}db2 TERMINATE
fi
exit $ret

```

These scripts are defining a set of variables to complete the operation. It is possible to overwrite the variables to customize the process execution. The `user_config_dir` can be used to configure where the custom configuration file will be stored.

In the following example, to use the DB2 COMPRESS option, it is possible to edit the file `/data/custom/db2_backup` file and overwrite the `BDB2_EXTRA_ARGS` variable to use the custom COMPRESS option. Note, that in this example, it is possible to use Bacula builtin compression instead.

```

db2inst1:~# grep db2 /opt/bacula/etc/bacula-dir.conf
Plugin = "db2: user_config_dir=/data/custom ctl_dir=/data/tmp"

```

(continues on next page)

(continued from previous page)

```
db2inst1:~# cat /data/custom/db2_backup
BDB2_EXTRA_ARGS="COMPRESS WITHOUT PROMPTING"
```

If the modification is more advanced, it is possible to control the Bacula FileDaemon DB2 shell protocol version.

```
db2inst1:~# cat /data/custom/db2_backup
if [ $BDB2_SHELL_VERSION = 1 ]; then
    BDB2_EXTRA_ARGS="COMPRESS WITHOUT PROMPTING"
fi
```

To optimize the format of the backup images for Aligned device or Global Endpoint Deduplication storage, it is possible to use:

```
db2inst1:~# cat /data/custom/db2_backup
if [ $BDB2_SHELL_VERSION = 1 ]; then
    BDB2_EXTRA_ARGS="DEDUP_DEVICE WITHOUT PROMPTING"
fi
```

DB2 Plugin Customization Interface Version 1

- db2_backup, db2_size, db2_dump_schema, db2_instance, db2_config, db2_list user files
 - BDB2_BIN_DIR=<string> Directory or string to prepend to all db2 commands.
 - BDB2_DATABASE=<string> Current database name.
 - BDB2_SHELL_VERSION=<int> Current Bacula Enterprise DB2 shell version.
- db2_backup user file
 - BDB2_BACKUP_LEVEL=<string> DB2 option to specify the backup level. Ex: BDB2_BACKUP_LEVEL=INCREMENTAL DELTA
 - BDB2_CMD=<string> Command to be executed via eval.
 - BDB2_EXTRA_ARGS=<string> DB2 database backup command arguments. Ex: BDB2_EXTRA_ARGS=WITHOUT PROMPTING
 - BDB2_FIFO=<path> Path to the communication Unix named pipe used by the script.
 - BDB2_INCLUDE_LOGS=<string> DB2 option to include LOGS in the backup if the database supports it. Ex: BDB2_INCLUDE_LOGS=INCLUDE LOGS
 - BDB2_JOB_LEVEL=<char> Current Bacula backup job level. Can be “F” for Full, “I” for Incremental and “D” for Differential.
 - BDB2_ONLINE=<string> DB2 ONLINE option if the database supports it. Ex: BDB2_ONLINE=ONLINE
 - BDB2_STOP_DB=<bool> If the DB2 database must be stopped during the backup. Ex: BDB2_STOP_DB=YES
 - BDB2_USER_STRING=<string> User and password configuration string passed to dblook schema analyzer.
- db2_restore

- BDB2_NEWDATABASE The target database name
- BDB2_REPLACE If the restore must replace a database. Ex: BDB2_REPLACE=REPLACE EXISTING
- BDB2_LOGTARGET is used when restoring the final image. (It is also possible to modify the option via bacula restore menu). Ex: BDB2_LOGTARGET=LOGTARGET '/tmp/'
- BDB2_TAKEN Ex. BDB2_TAKEN=TAKEN AT 20190815120000

These variables can be configured in the `user_config_dir`.

Examples

```
Fileset {
  Name = DB2_ALL_DB
  Description = "Backup all DB2 databases"
  Include {
    Plugin = "db2: ctl_dir=/data/tmp instance=db2inst1"
  }
}

Fileset {
  Name = DB2_OFFLINE
  Description = "Backup all DB2 databases OFFLINE"
  Include {
    Plugin = "db2: ctl_dir=/data/tmp instance=db2inst1 can_stop database=mydb"
  }
}
```

2.5 Backup

During a backup Job, the Bacula DB2 Plugin will analyze the configuration and adjust the parameters automatically.

For example, in the following example, Bacula will handle a database named DB2, as the database is configured with the TRACKMOD and the LOGARCHMETH1 options, Bacula will be able to run incremental and differential job level and backup database logs.

```
Fileset {
  Name = FS_DB2_DB
  Description = "Backup DB2 database"
  Include {
    Plugin = "db2: ctl_dir=/data/tmp instance=db2inst1 database=DB2"
  }
}

Job {
  Name = J_DB2_DB
  Fileset = FS_DB2_DB
  Client = db2-fd
  JobDefs = Defaults
}
```

The Full job output will look like the following:

```

dir JobId 1: No prior or suitable Full backup found in catalog. Doing FULL
->backup.
dir JobId 1: Start Backup JobId 1, Job=J_DB2_DB.2019-08-13_17.07.03_04
dir JobId 1: Using Device "FileStorage" to write.
sd JobId 1: Wrote label to prelabeled Volume "Vol001" on File device
->"FileStorage" (/storage)
fd JobId 1: DB2: -- No userid was specified, db2look tries to use Environment
->variable USER
fd JobId 1: DB2: -- USER is: DB2INST1
fd JobId 1: DB2: -- Creating DDL for table(s)
fd JobId 1: DB2: -- Binding package automatically ...
fd JobId 1: DB2: -- Bind is successful
fd JobId 1: DB2: -- Binding package automatically ...
fd JobId 1: DB2: -- Bind is successful
fd JobId 1: DB2: Doing backup of database "DB2"
fd JobId 1: DB2: BACKUP DATABASE DB2 ONLINE to /data/tmp/fifo.1 INCLUDE LOGS
->WITHOUT PROMPTING
fd JobId 1: DB2: Backup successful. The timestamp for this backup image is :
->20190813150713
sd JobId 1: Elapsed time=00:00:09, Transfer rate=1.586 M Bytes/second
sd JobId 1: Sending spooled attrs to the Director. Despooling 1,753 bytes ...
dir JobId 1: Bacula db2-dir 12.0.1 (05Aug19):
  Build OS:          x86_64-pc-linux-gnu archlinux
  JobId:             1
  Job:               J_DB2_DB.2019-08-13_17.07.03_04
  Backup Level:     Full (upgraded from Incremental)
  Client:           "db2-fd" 12.0.1 (05Aug19) x86_64-pc-linux-gnu,
->archlinux,
  Fileset:          "FS_DB2_DB" 2019-08-13 17:06:58
  Pool:             "Default" (From Job resource)
  Catalog:          "MyCatalog" (From Client resource)
  Storage:          "File" (From Job resource)
  Scheduled time:   13-Aug-2019 17:07:03
  Start time:       13-Aug-2019 17:07:05
  End time:         13-Aug-2019 17:07:14
  Elapsed time:     9 secs
  Priority:         10
  FD Files Written: 7
  SD Files Written: 7
  FD Bytes Written: 14,280,473 (14.28 MB)
  SD Bytes Written: 14,282,545 (14.28 MB)
  Rate:             1586.7 KB/s
  Software Compression: 92.9% 14.1:1
  Comm Line Compression: 4.3% 1.0:1
  Snapshot/VSS:    no
  Encryption:      no
  Accurate:        no
  Volume name(s):  Vol001
  Volume Session Id: 1
  Volume Session Time: 1565708807
  Last Volume Bytes: 14,328,370 (14.32 MB)
  Non-fatal FD errors: 0

```

(continues on next page)

(continued from previous page)

```
SD Errors:                0
FD termination status:    OK
SD termination status:    OK
Termination:              Backup OK

*list files jobid=1
Using Catalog "MyCatalog"
+-----+
| filename                |
+-----+
| /@DB2/db2inst1/DB2/schema.sql |
| /@DB2/db2inst1/DB2/data.sql.F |
| /@DB2/db2inst1/DB2/         |
| /@DB2/                    |
| /@DB2/db2inst1/           |
+-----+
```

The Incremental job output will look like the following:

```
dir JobId 2: Start Backup JobId 2, Job=J_DB2_DB.2019-08-13_17.07.15_05
dir JobId 2: Using Device "FileStorage" to write.
sd JobId 2: Volume "Vol001" previously written, moving to end of data.
fd JobId 2: DB2: -- No userid was specified, db2look tries to use Environment_
↪variable USER
fd JobId 2: DB2: -- USER is: DB2INST1
fd JobId 2: DB2: -- Creating DDL for table(s)
fd JobId 2: DB2: Doing backup of database "DB2"
fd JobId 2: DB2: BACKUP DATABASE DB2 ONLINE INCREMENTAL DELTA to /data/tmp/
↪fifo.2 INCLUDE LOGS
        WITHOUT PROMPTING
fd JobId 2: DB2: Backup successful. The timestamp for this backup image is :_
↪20190813150723
sd JobId 2: Elapsed time=00:00:07, Transfer rate=36.31 K Bytes/second
sd JobId 2: Sending spooled attrs to the Director. Despooling 1,131 bytes ...
dir JobId 2: Bacula db2-dir 12.0.1 (05Aug19):
  Build OS:                x86_64-pc-linux-gnu archlinux
  JobId:                    2
  Job:                      J_DB2_DB.2019-08-13_17.07.15_05
  Backup Level:             Incremental, since=2019-08-13 17:07:05
  Client:                   "db2-fd" 12.0.1 (05Aug19) x86_64-pc-linux-gnu,
↪archlinux,
  Fileset:                  "FS_DB2_DB" 2019-08-13 17:06:58
  Pool:                     "Default" (From Job resource)
  Catalog:                  "MyCatalog" (From Client resource)
  Storage:                  "File" (From Job resource)
  Scheduled time:           13-Aug-2019 17:07:15
  Start time:               13-Aug-2019 17:07:17
  End time:                 13-Aug-2019 17:07:24
  Elapsed time:             7 secs
  Priority:                 10
  FD Files Written:         6
  SD Files Written:         6
```

(continues on next page)

(continued from previous page)

```
FD Bytes Written:      252,832 (252.8 KB)
SD Bytes Written:      254,221 (254.2 KB)
Rate:                  36.1 KB/s
Software Compression:  99.5% 198.5:1
Comm Line Compression: 80.7% 5.2:1
Snapshot/VSS:          no
Encryption:            no
Accurate:              no
Volume name(s):        Vol001
Volume Session Id:     2
Volume Session Time:   1565708807
Last Volume Bytes:     14,592,649 (14.59 MB)
Non-fatal FD errors:   0
SD Errors:             0
FD termination status: OK
SD termination status: OK
Termination:           Backup OK
```

```
*list files jobid=2
```

```
Using Catalog "MyCatalog"
```

```
+-----+
| filename |
+-----+
| /@DB2/db2inst1/DB2/data.sql.I.20190813170822 |
| /@DB2/db2inst1/DB2/schema.sql |
| /@DB2/db2inst1/DB2/ |
| /@DB2/db2inst1/ |
| /@DB2/ |
+-----+
```

In the following example, the DB3 and DB1 databases are not configured to allow ONLINE backups. The plugin will stop the databases automatically during the backup. If a user or an application is connected to the database, the plugin will terminate all connections automatically.

```
Fileset {
  Name = FS_DB2_DB3
  Description = "Backup DB1/DB3 database"
  Include {
    Plugin = "db2: ctl_dir=/data/tmp instance=db2inst1 database=DB[13] can_stop
↪"
  }
}

Job {
  Name = J_DB3_DB
  Fileset = FS_DB3_DB
  Client = db2-fd
  JobDefs = Defaults
}
```

```
dir JobId 5: Start Backup JobId 5, Job=J_DB3_DB.2019-08-13_17.07.43_08
dir JobId 5: Using Device "FileStorage" to write.
```

(continues on next page)

(continued from previous page)

```
sd JobId 5: Volume "Vol001" previously written, moving to end of data.
fd JobId 5: DB2: -- No userid was specified, db2look tries to use Environment
↳variable USER
fd JobId 5: DB2: -- USER is: DB2INST1
fd JobId 5: DB2: -- Creating DDL for table(s)
fd JobId 5: DB2: -- Binding package automatically ...
fd JobId 5: DB2: -- Bind is successful
fd JobId 5: DB2: -- Binding package automatically ...
fd JobId 5: DB2: -- Bind is successful
fd JobId 5: DB2: Doing backup of database "DB3"
fd JobId 5: DB2: Stopping the database...
fd JobId 5: DB2: CONNECT TO DB3
fd JobId 5: DB2: Database Connection Information
fd JobId 5: DB2: Database server = DB2/LINUX8664 11.5.0.0
fd JobId 5: DB2: SQL authorization ID = DB2INST1
fd JobId 5: DB2: Local database alias = DB3
fd JobId 5: DB2: LIST APPLICATION FOR DATABASE DB3
fd JobId 5: DB2: Auth Id Application Appl. Application Id
↳ DB # of
fd JobId 5: DB2: Name Handle
↳ Name Agents
fd JobId 5: DB2: -----
↳-----
fd JobId 5: DB2: DB2INST1 db2bp 996 *LOCAL.db2inst1.
↳190813151420 DB3 1
fd JobId 5: DB2: QUIESCE DATABASE IMMEDIATE FORCE CONNECTIONS
fd JobId 5: DB2: DB20000I The QUIESCE DATABASE command completed
↳successfully.
fd JobId 5: DB2: DB20000I The TERMINATE command completed successfully.
fd JobId 5: DB2: DEACTIVATE DATABASE DB3
fd JobId 5: DB2: DB20000I The DEACTIVATE DATABASE command completed
↳successfully.
fd JobId 5: DB2: BACKUP DATABASE DB3 to /database//tmp/fifo.5 WITHOUT
↳PROMPTING
fd JobId 5: DB2: Backup successful. The timestamp for this backup image is :
↳20190813150759
fd JobId 5: DB2: Starting the database...
fd JobId 5: DB2: ACTIVATE DATABASE DB3
fd JobId 5: DB2: DB20000I The ACTIVATE DATABASE command completed
↳successfully.
fd JobId 5: DB2: CONNECT TO DB3
fd JobId 5: DB2: Database Connection Information
fd JobId 5: DB2: Database server = DB2/LINUX8664 11.5.0.0
fd JobId 5: DB2: SQL authorization ID = DB2INST1
fd JobId 5: DB2: Local database alias = DB3
fd JobId 5: DB2: UNQUIESCE DATABASE
fd JobId 5: DB2: DB20000I The UNQUIESCE DATABASE command completed
↳successfully.
fd JobId 5: DB2: DB20000I The TERMINATE command completed successfully.
fd JobId 5: DB2: -- No userid was specified, db2look tries to use Environment
↳variable USER
fd JobId 5: DB2: -- USER is: DB2INST1
```

(continues on next page)

(continued from previous page)

```
fd JobId 5: DB2: -- Creating DDL for table(s)
fd JobId 5: DB2: -- Binding package automatically ...
fd JobId 5: DB2: -- Bind is successful
fd JobId 5: DB2: -- Binding package automatically ...
fd JobId 5: DB2: -- Bind is successful
fd JobId 5: DB2: Doing backup of database "DB1"
fd JobId 5: DB2: Stopping the database...
fd JobId 5: DB2: CONNECT TO DB1
fd JobId 5: DB2: Database Connection Information
fd JobId 5: DB2: Database server = DB2/LINUX8664 11.5.0.0
fd JobId 5: DB2: SQL authorization ID = DB2INST1
fd JobId 5: DB2: Local database alias = DB1
fd JobId 5: DB2: LIST APPLICATION FOR DATABASE DB1
fd JobId 5: DB2: Auth Id Application Appl. Application Id
↪ DB # of
fd JobId 5: DB2: Name Handle
↪ Name Agents
fd JobId 5: DB2: -----
↪ -----
fd JobId 5: DB2: DB2INST1 db2bp 1052 *LOCAL.db2inst1.
↪190813151516 DB1 1
fd JobId 5: DB2: QUIESCE DATABASE IMMEDIATE FORCE CONNECTIONS
fd JobId 5: DB2: DB20000I The QUIESCE DATABASE command completed.
↪successfully.
fd JobId 5: DB2: DB20000I The TERMINATE command completed successfully.
fd JobId 5: DB2: DEACTIVATE DATABASE DB1
fd JobId 5: DB2: DB20000I The DEACTIVATE DATABASE command completed.
↪successfully.
fd JobId 5: DB2: BACKUP DATABASE DB1 to /database//tmp/fifo.5 WITHOUT
↪PROMPTING
fd JobId 5: DB2: Backup successful. The timestamp for this backup image is :.
↪20190813150813
fd JobId 5: DB2: Starting the database...
fd JobId 5: DB2: ACTIVATE DATABASE DB1
fd JobId 5: DB2: DB20000I The ACTIVATE DATABASE command completed.
↪successfully.
fd JobId 5: DB2: CONNECT TO DB1
fd JobId 5: DB2: Database Connection Information
fd JobId 5: DB2: Database server = DB2/LINUX8664 11.5.0.0
fd JobId 5: DB2: SQL authorization ID = DB2INST1
fd JobId 5: DB2: Local database alias = DB1
fd JobId 5: DB2: UNQUIESCE DATABASE
fd JobId 5: DB2: DB20000I The UNQUIESCE DATABASE command completed.
↪successfully.
fd JobId 5: DB2: DB20000I The TERMINATE command completed successfully.
sd JobId 5: Elapsed time=00:00:31, Transfer rate=6.386 M Bytes/second
sd JobId 5: Sending spooled attrs to the Director. Despooling 2,494 bytes ...
dir JobId 5: Bacula db2-dir 12.0.1 (05Aug19):
Build OS: x86_64-pc-linux-gnu archlinux
JobId: 5
Job: J_DB3_DB.2019-08-13_17.07.43_08
Backup Level: Full (upgraded from Incremental)
```

(continues on next page)

(continued from previous page)

```
Client: "db2-fd" 12.0.1 (05Aug19) x86_64-pc-linux-gnu,
↪archlinux,
Fileset: "FS_DB3_DB" 2019-08-13 17:07:43
Pool: "Default" (From Job resource)
Catalog: "MyCatalog" (From Client resource)
Storage: "File" (From Job resource)
Scheduled time: 13-Aug-2019 17:07:43
Start time: 13-Aug-2019 17:07:45
End time: 13-Aug-2019 17:08:16
Elapsed time: 31 secs
Priority: 10
FD Files Written: 11
SD Files Written: 11
FD Bytes Written: 197,964,193 (197.9 MB)
SD Bytes Written: 197,967,386 (197.9 MB)
Rate: 6385.9 KB/s
Software Compression: 43.9% 1.8:1
Comm Line Compression: 84.2% 6.3:1
Snapshot/VSS: no
Encryption: no
Accurate: no
Volume name(s): Vol001
Volume Session Id: 5
Volume Session Time: 1565708807
Last Volume Bytes: 213,265,390 (213.2 MB)
Non-fatal FD errors: 0
SD Errors: 0
FD termination status: OK
SD termination status: OK
Termination: Backup OK

*list files jobid=5
+-----+
| filename |
+-----+
| /@DB2/ |
| /@DB2/db2inst1/ |
| /@DB2/db2inst1/DB3/schema.sql |
| /@DB2/db2inst1/DB3/data.sql.F |
| /@DB2/db2inst1/DB3/ |
| /@DB2/db2inst1/DB1/schema.sql |
| /@DB2/db2inst1/DB1/data.sql.F |
| /@DB2/db2inst1/DB1/ |
+-----+
```

2.6 Restore

The Bacula Enterprise DB2 plugin is designed to help DB2 administrator manage the restore procedure.

Restoring a Single Full Database

To restore a single database with the Bacula Enterprise DB2 plugin, the appropriate files from the database directory are selected during the restore process.

To restore the database with its original name, the selection should only contain the data file (data.sql.*).

To restore a single database to a new name, the file data.sql.* must be selected. The **where** parameter is used to specify the new database name. If **where** is set to a single word consisting of only a . . z, 0-9 and _, Bacula will create the specified database and restore the data into it. On DB2, the database name has some restrictions, please see DB2 documentation on that subject.

```
* restore where=bacula2
...
cwd is: /
$ cd /@DB2/db2inst1/BACULA
cwd is: /@DB2/db2inst1/BACULA
$ m data.sql.F
$ lsmark
*data.sql.F
```

If the **replace** parameter is set to **Never**, Bacula will check the database list, and will abort the Job if the database being restored already exists.

Once the database is restored, the **ROLLFORWARD** command is needed to finish the restore. Example: db2 ROLLFORWARD DATABASE test1 COMPLETE.

Restoring an Incremental/Differential Backup

The following procedure describes how to restore a database to a point in time.

1. Select the final image to be restored. This is the usually just the most recent data.sql.* file from your Bacula backups of the database. Access this using the normal Bacula restore procedure, selecting the jobs for the client up to the current date, and then mark and restore just the most recent backup. Please see the following code snippets for an example. This will restore the database configurations and necessary initial information.
2. Next, the restore command is run again, and all the database backups back to the most recent full are selected. Make sure that the job selected in step 1 is included. The 'mark *' command should select the correct data as long as the job listing is the same as in step 1. Bacula will restore each of the required full, differential or incremental backupimages, in the order in which they were produced, on top of the baseline image restored in Step 1. The last image restored will extract database logs automatically.
3. Once all the data is restored, the rollforward command must be executed to complete the backup and replay all necessary log files.

During a Bacula incremental restore operation, only the logs included in the target image of the restore operation will be retrieved from the backup image. Any logs that are included in intermediate images referenced during the incremental restore process will not be extracted from those backup images.

Once the database is restored, the **ROLLFORWARD** command is needed to finish the restore. Example: db2 ROLLFORWARD DATABASE test1 TO END OF LOGS AND COMPLETE

```
*restore client=sap-fd where=test1 select
+-----+-----+-----+-----+-----+-----+-----+-----+
↔+
```

(continues on next page)

(continued from previous page)

```
| jobid | level | jobfiles | jobbytes | starttime | volumename |
+-----+-----+-----+-----+-----+-----+
| 1 | F | 7 | 14,296,244 | 2019-08-15 11:13:43 | TestVolume001 |
| 4 | D | 6 | 253,359 | 2019-08-15 11:14:14 | TestVolume001 |
| 6 | I | 6 | 253,715 | 2019-08-15 11:15:05 | TestVolume001 |
| 7 | I | 6 | 253,770 | 2019-08-15 11:15:16 | TestVolume001 |
+-----+-----+-----+-----+-----+-----+

You have selected the following JobIds: 1,4,6,7

Building directory tree for JobId(s) 1,4,6,7 ...
5 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ cd @DB2
cwd is: /@DB2/
$ cd db2inst1/
cwd is: /@DB2/db2inst1/
$ cd DB2
cwd is: /@DB2/db2inst1/DB2/
$ ls
data.sql.D.20190815110819
data.sql.F
data.sql.I.20190815110810
data.sql.I.20190815110820
schema.sql
$ m data.sql.I.20190815110820
1 file marked.
$ done
Bootstrap records written to /tmp/regress/working/127.0.0.1-dir.restore.2.bsr

The Job will require the following (*=>InChanger):
Volume(s) Storage(s) SD Device(s)
=====
TestVolume001 File FileStorage

Volumes marked with "*" are in the Autochanger.

1 file selected to be restored.
```

(continues on next page)

(continued from previous page)

```
Run Restore job
JobName:          RestoreFiles
Bootstrap:        /opt/bacula/working/working/127.0.0.1-dir.restore.2.bsr
Where:            test1
Replace:          Always
Fileset:          Full Set
Backup Client:    db2-fd
Restore Client:   db2-fd
Storage:          File
When:             2019-08-15 11:25:10
Catalog:          MyCatalog
Priority:          10
Plugin Options:   *None*
OK to run? (yes/mod/no): yes
Job queued. JobId=18
*messages
dir JobId 18: Start Restore Job RestoreFiles.2019-08-15_11.25.22_03
dir JobId 18: Restoring files from JobId(s) 1,4,6,7
dir JobId 18: Using Device "FileStorage" to read.
sd JobId 18: Ready to read from volume "TestVolume001" on File device
↳"FileStorage" (/tmp).
sd JobId 18: Forward spacing Volume "TestVolume001" to addr=196768511
sd JobId 18: Elapsed time=00:00:01, Transfer rate=243.8 K Bytes/second
fd JobId 18: DB2: RESTORE DATABASE DB2 INCREMENTAL from /database//tmp/fifo.18
                TAKEN AT 20190815091522 INTO test1 LOGTARGET DEFAULT WITHOUT_
↳PROMPTING
fd JobId 18: DB2: DB20000I The RESTORE DATABASE command completed_
↳successfully.
dir JobId 18: Bacula 127.0.0.1-dir 12.0.1 (05Aug19):
  Build OS:          x86_64-pc-linux-gnu archlinux
  JobId:             18
  Job:               RestoreFiles.2019-08-15_11.25.22_03
  Restore Client:    127.0.0.1-fd
  Where:             test1
  Replace:           Always
  Start time:        15-Aug-2019 11:25:24
  End time:          15-Aug-2019 11:25:28
  Elapsed time:      4 secs
  Files Expected:    1
  Files Restored:    1
  Bytes Restored:    50,376,704 (50.37 MB)
  Rate:              12594.2 KB/s
  FD Errors:         0
  FD termination status: OK
  SD termination status: OK
  Termination:      Restore OK
```

At this step, the DB2 database was created on the server, and you need now to select all files to perform the restore.

```
*restore client=sap-fd where=test1 select
```

(continues on next page)

(continued from previous page)

```
+-----+-----+-----+-----+-----+-----+-----+-----+
↪+
| jobid | level | jobfiles | jobbytes  | starttime          | volumename  ↪
↪|
+-----+-----+-----+-----+-----+-----+-----+-----+
↪+
|      1 | F     |         7 | 14,296,244 | 2019-08-15 11:13:43 | TestVolume001↪
↪|
|      4 | D     |         6 |   253,359 | 2019-08-15 11:14:14 | TestVolume001↪
↪|
|      6 | I     |         6 |   253,715 | 2019-08-15 11:15:05 | TestVolume001↪
↪|
|      7 | I     |         6 |   253,770 | 2019-08-15 11:15:16 | TestVolume001↪
↪|
+-----+-----+-----+-----+-----+-----+-----+-----+
↪+
You have selected the following JobIds: 1,4,6,7

Building directory tree for JobId(s) 1,4,6,7 ...
5 files inserted into the tree.

You are now entering file selection mode where you add (mark) and
remove (unmark) files to be restored. No files are initially added, unless
you used the "all" keyword on the command line.
Enter "done" to leave this mode.

cwd is: /
$ cd @DB2
cwd is: /@DB2/
$ cd db2inst1/
cwd is: /@DB2/db2inst1/
$ cd DB2
cwd is: /@DB2/db2inst1/DB2/
$ m *
5 files marked.
$ lsmark
*data.sql.D.20190815110819
*data.sql.F
*data.sql.I.20190815110810
*data.sql.I.20190815110820
*schema.sql
$ done
Bootstrap records written to /tmp/regress/working/127.0.0.1-dir.restore.1.bsr

The Job will require the following (*=>InChanger):
  Volume(s)                Storage(s)                SD Device(s)
=====
  TestVolume001            File                       FileStorage

Volumes marked with "*" are in the Autochanger.
```

(continues on next page)

(continued from previous page)

```
5 files selected to be restored.

Using Catalog "MyCatalog"
Run Restore job
JobName:          RestoreFiles
Bootstrap:        /tmp/regress/working/127.0.0.1-dir.restore.1.bsr
Where:            test1
Replace:          Always
Fileset:          Full Set
Backup Client:    db2-fd
Restore Client:   db2-fd
Storage:          File
When:             2019-08-15 11:40:54
Catalog:          MyCatalog
Priority:          10
Plugin Options:   *None*
OK to run? (yes/mod/no): Job queued. JobId=19
*messages
dir JobId 19: Start Restore Job RestoreFiles.2019-08-15_11.40.56_03
dir JobId 19: Restoring files from JobId(s) 1,4,6,7
dir JobId 19: Using Device "FileStorage" to read.
sd JobId 19: Ready to read from volume "TestVolume001" on File device
↳ "FileStorage" (/tmp).
sd JobId 19: Forward spacing Volume "TestVolume001" to addr=223
fd JobId 19: DB2: RESTORE DATABASE DB2 INCREMENTAL from /database//tmp/fifo.19
      TAKEN AT 20190815091351 INTO test1 WITHOUT PROMPTING
sd JobId 19: Elapsed time=00:00:03, Transfer rate=5.009 M Bytes/second
fd JobId 19: DB2: DB20000I The RESTORE DATABASE command completed.
↳ successfully.
fd JobId 19: DB2: RESTORE DATABASE DB2 INCREMENTAL from /database//tmp/fifo.19
      TAKEN AT 20190815091420 INTO test1 WITHOUT PROMPTING
fd JobId 19: DB2: DB20000I The RESTORE DATABASE command completed.
↳ successfully.
fd JobId 19: DB2: RESTORE DATABASE DB2 INCREMENTAL from /database//tmp/fifo.19
      TAKEN AT 20190815091511 INTO test1 WITHOUT PROMPTING
fd JobId 19: DB2: DB20000I The RESTORE DATABASE command completed.
↳ successfully.
fd JobId 19: DB2: RESTORE DATABASE DB2 INCREMENTAL from /database//tmp/fifo.19
      TAKEN AT 20190815091522 INTO test1 LOGTARGET DEFAULT WITHOUT
↳ PROMPTING
fd JobId 19: DB2: DB20000I The RESTORE DATABASE command completed.
↳ successfully.
fd JobId 19: DB2: No ROLLFORWARD option specified. Database is in pending mode
db2 ROLLFORWARD DATABASE test1 TO END OF LOGS
db2 ROLLFORWARD DATABASE test1 COMPLETE
15-Aug 11:41 127.0.0.1-dir JobId 19: Bacula 127.0.0.1-dir 12.0.1 (05Aug19):
  Build OS:          x86_64-pc-linux-gnu archlinux
  JobId:              19
  Job:                RestoreFiles.2019-08-15_11.40.56_03
  Restore Client:     127.0.0.1-fd
  Where:              test1
```

(continues on next page)

(continued from previous page)

```
Replace:                Always
Start time:             15-Aug-2019 11:40:58
End time:               15-Aug-2019 11:41:26
Elapsed time:           28 secs
Files Expected:         5
Files Restored:         5
Bytes Restored:         352,538,624 (352.5 MB)
Rate:                   12590.7 KB/s
FD Errors:              0
FD termination status: OK
SD termination status: OK
Termination:           Restore OK
```

To access the database, you must now execute the ROLLFORWARD function up to the point in time that is needed.

```
[db2inst1@db2 tmp]$ db2 ROLLFORWARD DATABASE test1 TO END OF LOGS AND COMPLETE
```

Rollforward Status

```
Input database alias           = test1
Number of members have returned status = 1

Member ID                      = 0
Rollforward status             = not pending
Next log file to be read       =
Log files processed             = -
Last committed transaction     = 2019-08-15-09.15.22.000000 UTC
```

```
DB20000I The ROLLFORWARD command completed successfully.
```

```
[db2inst1@db2 tmp]$ db2 connect to test1
```

Database Connection Information

```
Database server      = DB2/LINUX8664 11.5.0.0
SQL authorization ID = DB2INST1
Local database alias = TEST1
```

```
[db2inst1@db2 tmp]$ db2 'select * from a order by when desc limit 2'
```

```
VAL          WHEN
-----
1 2019-08-15-09.15.17.792096
1 2019-08-15-09.15.13.121704
```

```
2 record(s) selected.
```

Restoring Dump Files to a Directory

To restore DB2 images to a directory, the **where** parameter needs to be set to indicate an existing directory.

```
* restore where=/tmp
```

If the **where** parameter is a directory (containing /), Bacula will restore all files into this directory. Doing so, it is possible to use `db2 restore` directly and restore only particular contents.

```
[db2inst1@db2 tmp]$ cd /tmp/@DB2/db2inst1/DB2
[db2inst1@db2 DB2]$ mkfifo myfifo
[db2inst1@db2 DB2]$ cat data.sql.F > myfifo &
[db2inst1@db2 DB2]$ db2 restore dbtest from myfifo
```

2.7 Limitations

The Bacula Enterprise DB2 plugin has been written to help backup administrator manage and recover DB2 databases. The Plugin doesn't implement all DB2 backup features. For example, it is not possible to backup a particular tablespace with the plugin natively. For very advanced and granular use, it is possible to use the bpipe plugin. Please contact Bacula Systems for assistance with the bpipe plugin.

The XBA API is not part of the Bacula Enterprise DB2 Plugin.

The `restart` command has limitations with plugins, as it initiates the Job from scratch rather than continuing it. Bacula determines whether a Job is restarted or continued, but using the `restart` command will result in a new Job.

2.8 Troubleshooting

Troubleshooting: Restore

DB2 error SQL1224N

If, during a restore to the DB2 environment, the following error is reported:

```
DB2: SQL1224N The database manager is not able to accept new requests, has
DB2: terminated all requests in progress, or has terminated the specified
↳request
DB2: because of an error or a forced interrupt.
```

please, check the maximum number of allowed database connections configured in the DB2 environment and, if possible, increase it. Alternatively, removing as much as possible (preferably all) active database connections before initiating the restore will help mitigate the issue.