



## Managing the RDU Database

---

The regional distribution unit (RDU) database is the central database that Broadband Provisioning Registrar (BPR) uses. As with any database, it is essential that you, as the administrator, understand and be familiar with database recovery from failure conditions. This chapter explains how to perform database backup and recovery.

### Enabling Recovery Using Write-Ahead Logging

The RDU database uses a technique known as write-ahead logging to enable recovery. This involves writing a description of any database changes to a log file before the change is written into the database.

Write-ahead logging prevents the database from becoming corrupted from an application or system crash. Such a crash can create inconsistencies within the database files. However, during subsequent start-up of the RDU server, automatic recovery is performed. During this recovery process, the log and database files are synchronized.

Database changes that were written into the log but not, prior to a crash, written into the database are, during automatic recovery, written into the database.

In this way, write-ahead logging virtually guarantees that the database will never become corrupted when the RDU server crashes because the database is automatically repaired when the RDU server is restarted.

### Benefits

Since the database and database logs are stored in separate files, they can be also be stored on separate physical storage media. Doing so has two distinct advantages:

- Safety against one of the disks failing—This is achieved when, if a database disk fails, the entire database can be recovered from its backup and restored it to its current state using the previous backup and database logs.
- Enhanced performance—This is achieved because writes to disk are distributed between two separate physical disks or storage media; database writes go to one media and log records go to another.



**Note**

---

The locations of the database and database logs are specified during installation of the RDU server.

---

# Backup and Recovery

The RDU server supports both full (the complete database) and incremental (the latest database changes only) online backups. These backups can be performed without stopping the server.

To restore the database from backup, you need a snapshot of the database file and, optionally, any number of sequential log files generated since the snapshot of the database was taken.

A full online backup involves copying the database file and applicable log files to backup media while an incremental online backup only copies the log files.

While backing up the RDU does not have any direct impact on the performance of the database, because it uses the same disk and system resources, it might indirectly affect the performance of the RDU server. Consequently, you should perform any RDU backup during those times when system loads are at their lowest levels.

## Log Files

RDU server stores database logs in 10 MB files in the database log directory. While the database log file directory is configured during installation, the default log directory is:

```
<BPR_DBLOG>/rdu/dblog directory
```

where:

- <BPR\_DBLOG>—identifies the installation directory for database logs



Note

---

Log files are named in numeric sequence, starting at log.00000001. The instructions found in this chapter highlight the importance of having the log files in uninterrupted sequence.

---

For back-up purposes, log files can be in one of these states:

- Active—These are log files that contain data not yet written into the database. It is important to keep active log files until they become redundant.
- Redundant—A log file becomes redundant once its data has been written into the database. The `listRedundantLogs.sh` utility helps to identify redundant log files. These redundant log files can be deleted after appropriate backup is made. Refer to the [“Using the listRedundantLogs.sh Utility” section on page 5-5](#) for specific instruction on using the `listRedundantLogs.sh` utility.
- Removable—These are redundant files that can be removed. A log file becomes removable after either being backed up, or when the database that contains data for this log file has been backed up.

## Monitoring Disk Space

With each log file occupying 10MB of disk space it is possible that all log files could rapidly fill all of your available disk space. If database log files were never removed, they would quickly occupy approximately twice as much hard disk space as the database file itself. It is equally possible that this can happen without you being aware of the situation. To prevent this from happening, you need to establish a process for monitoring disk space. A sample script, called `disk_monitor.sh`, is provided with BPR. This script is located in the `<BPR_HOME>/rdu/sample/tools` directory.

The `disk_monitor.sh` script sets threshold values for one or more file systems. When these thresholds are surpassed, an alert is generated through the Solaris `syslog` facility, at 60 second intervals, until additional disk space is available.

**Note**


---

Cisco recommends that, at a minimum, you use the `disk_monitor.sh` script to monitor the `<BPR_DATA>` and `<BPR_DBLOG>` directories.

---

To run this script from the command line interface, enter this command:

```
# ./disk_monitor.sh (file system-directory) (x)
```

Where:

- *(file system-directory)*—identifies any directory, in a file system, that needs monitoring.
- *(x)*—identifies the percentage threshold applied to the specified file system.

For example, assume that you want to be notified when a file system (`var/bpr` for example) with database logs reaches 80% of its capacity. You would enter the command this way:

```
# ./disk_monitor.sh /var/bpr 80
```

When the disk space used by this file system on which the database logs reside reaches 80% capacity, an alert such as that shown below, is sent to syslog:

```
Dec 7 8:16:03 perf-u80-1 BPR: [ID 702911 local6.warning] Filesystem /var/bpr usage is 81%
(threshold is 80%)
```

## Database Files

The RDU server stores its database in a file called `bpr.db` which is found in the database directory. While this directory is configured during the installation process, its default location is:

```
<BPR_DATA>/rdu/db directory
```

where:

- `<BPR_DATA>` identifies the installation directory for database files.

**Note**


---

Every time you are doing a full backup, you must copy the database file to the backup media.

---

## Full Online Backup

A full online backup involves copying the database file, and applicable log files, to backup media without having to stop the RDU server. To perform a full online backup:

---

**Step 1** Identify all redundant log files using the `listRedundantLogs.sh` utility.

Refer to the [“Using the listRedundantLogs.sh Utility” section on page 5-5](#) for specific instruction on using the `listRedundantLogs.sh` utility. These files have to be identified because they must not be copied to the backup. In the case of these files, they contain data that already exists in the database file that you are going to backup.

**Step 2** Copy the database file to the backup media.




---

**Note** Prior to copying the database file, you may compress the file (creating a zip or tar file) to conserve disk space. When doing so, you must use a compression utility that does not attempt to replace the file being compressed. The compression utility should make a new file with a compressed copy of the database file.

---

**Step 3** Verify that the database file was accurately copied by performing an approximate comparison of the file sizes between the disk file and the backup file.




---

**Note** The comparison is approximate because the database file may grow as you are doing the backup due to concurrent database activity.

---

**Step 4** Delete all of the redundant log files identified in Step 1.




---

**Note** While performing this step, do not run the listRedundantLogs.sh utility. This will produce a list of log files different from that generated in Step #1. Consequently, log files that became redundant since running the utility in Step 1 may not have been copied into the newly backed up database file.

---

The database contains the changes in these log files and the database has been backed up so the files are no longer needed.




---

**Note** Be careful not to destroy any other log files.

---

**Step 5** Copy all remaining log files located in the database log directory to the backup media. If there is concurrent RDU activity, when you copy these log files, they may naturally grow in size and number. You must ensure that you copy all of the log files that existed when you started the process.

**Step 6** Verify that the files were successfully copied to the backup media.

**Step 7** Once the full backup is complete, you can safely discard any previous full and incremental backups that are not required. However, Cisco recommends that you keep at least three previous database copies as a safeguard.




---

**Note** The database directory may contain some files with a file format similar to \_\_db.001.

---

## Incremental Online Backup

An incremental online backup involves copying all redundant log files that have not yet been archived to the backup media.



### Caution

---

When creating incremental backups, it is important to exercise extreme caution to prevent loss or damage to your data.

---

To create an incremental backup:

- 
- Step 1** Create a list of all redundant log files using the `listRedundantLogs.sh` utility. Refer to the [“Using the `listRedundantLogs.sh` Utility” section on page 5-5](#) for specific instruction on using the `listRedundantLogs.sh` utility.
  - Step 2** Copy all log files, identified with the `listRedundantLogs.sh` utility, to backup media.
  - Step 3** Verify that the log files have been copied successfully.
  - Step 4** Delete the log files copied in step 2.
- 

Incremental online backups collect all redundant log files made since the previous full backup. Provided that all log files exist, in sequence, since the last database backup, the database can be recovered, however, the log files must be in sequence. For example, if you have `log.0000002`, `log.0000003` and `log.0000005`, but not `log.0000004`, you can only restore your database up to the changes reflected in `log.0000003`. Consequently, it is very important that you keep an archived version of a full backup and all subsequent incremental backups. Once a new full backup is made, you may discard any incremental backups done before the full backup.

## Using the `listRedundantLogs.sh` Utility

Both full and incremental backups require the use of the `listRedundantLogs.sh` utility that is located in the `<BPR_HOME>/rdu/bin` directory. This helps you identify the log files that contain changes already copied into the database file. The `listRedundantLogs.sh` utility is located in this directory:

```
# ./listRedundantLogs.sh [-noverbose] [?]
```

where:

- `[-noverbose]`—Prints out the minimum necessary data. No additional information or text is printed.
- `[?]`—Prints out command help information.

This command creates an output that contains a list of redundant log files. These log files reflect the changes that have already been applied to the database. The command has a `-noverbose` parameter for use when auto-parsing is required. The `-noverbose` mode cause the printed results to appear as a formatted list of log filenames.



---

**Note** You can run this command regardless of whether the RDU server is operational or not, but you must not run the command while the RDU server is starting up.

---

## Failure Recovery

The recovery process involves restoring the latest backup of the database from the backup media. Additionally, any subsequent and sequential log files, created since the last database backup, can be used to bring the database more up to date.

Recovery is accomplished using the `recoverDb.sh` utility that is located in the `<BPR_HOME>/rdu/bin` directory. This tool uses the location of the database and log file directories as its input parameters. The result of running this utility is a recovered database with a `bpr.db` filename. This file will always be located in the database directory provided. The database can be moved to the appropriate RDU server directory and after the RDU server is restarted, the recovered database is operational.

To recover from a failure:

---

**Step 1** Place the most recently backed up copy of the `bpr.rdb` file into a recovery directory.




---

**Note** For example purposes, assume this to be `/tmp/db`.

---

**Step 2** Place all log files archived since the backup from step 1 into the backup directory.




---

**Note** For example purposes, assume this to be `/tmp/dblog`.

---

As noted earlier in this chapter, log files must be sequential. Refer to the [“Log Files and Recovery” section on page 5-7](#) for additional information.

- If you are recovering from a full backup, place the log files archived during that full backup into this directory.
- If you are recovering using a combination of a full and incremental backup, you must place the incremental backup log files into this directory after the full backup files. This is necessary because at least one file from both backup types will have the same file name and it is important that the newer file (from the incremental backup) overwrites the log file resulting from the full backup.
- You can also use the log files that are available in the database installation directories, provided that they are sequential with logs from the full and incremental backup. If you use these log files, they can overwrite all log files with the same names from a previous full backup.

**Step 3** Locate the `recoverDb.sh` utility. This utility is normally found in the `<BPR_HOME>/rdu/bin` directory.

**Step 4** Run the `recoverDb.sh` utility and provide the location of database and log file directories. For example:

```
# ./recoverDb.sh /tmp/db /tmp/dblog [?]
```

Where:

- `/tmp/db`—identifies the location of the most recently backed up copy of `bpr.db`
- `/tmp/dblog`—identifies the location of the archived log files
- `[?]`—Prints out command help information.

**Step 5** When the `recoverDb.sh` utility finishes, a recovered database file (`bpr.db`) will be located in the database directory provided.




---

**Note** For example purposes, assume this to be `/tmp/db/bpr.db`.

---

**Step 6** Stop the RDU server, if it is still running.

**Step 7** Move or backup the current database file and all log files found under RDU server directories `<BPR_DB>/rdu/db` and `<BPR_DBLOG>/rdu/dblog`.



---

**Note** The database directory may contain some files with a file format similar to \_\_db.001. You can delete these and all other files in the directories previously referenced.

---

- Step 8** Delete all files from these two directories:
- <BPR\_DATA>/rdu/db
  - <BPR\_DBLOG>/rdu/dblog
- Step 9** Copy the recovered database file bpr.db to <BPR\_DB>/rdu/db. In this example, the location of the recovered database file is /tmp/db/bpr.db.
- Step 10** Start the RDU server again. Once the message indicating that the server has successfully started appears, the RDU operates with the recovered database.
- Step 11** Verify that RDU is operating successfully. After doing so, you can destroy the contents of directories used to do the recovery (/tmp/db and /tmp/dblog, in this example.) You should, however, keep the backup archive of these files.
- 

## Log Files and Recovery

As described in the [“Failure Recovery” section on page 5-5](#), you can place all log files archived since the previous database backup, into the backup directory. You can actually place a combination of archived log files and log files that you had remaining in your RDU database log directory, into the backup directory.

This allows the database to be recovered almost entirely in case of a disk failure with the database. Assuming that the database logs are on different disks, you can still use them to recover the database to the most recent date.

The only restriction in place is that all log files have to be sequential, that is, the numeric extensions of all log files have to cover a complete range of numbers. For example, if you place log files log.00010, log.00011, log.00013, log.00017 and log.00018, then the recovery will fail with an error message saying that log.00012 was not found. You could successfully recover up to log.00011 if you remove all subsequent logs files. In this example, that would include log.00017 and log.00018.

By controlling the number of log files you use during recovery, you can control how far back in time, or how recent, you recreate your database. This maybe useful if you want to return your database to some approximate point back in time.

## Running Out of Disk Space

When the RDU server runs out of disk space it normally generates an alert, through the syslog and the rdu log. The RDU server then tries to automatically restart. The RDU server continues trying to restart until free disk space is available, such as after you have compressed the rdu logs (not database logs) or they have been moved to a new disk. In the event that additional disk space cannot be created, you can perform a standard backup of the database. Doing so allows you to free some disk space by deleting certain database log files as part of the backup process.

If your database grows in size beyond the capacity of the current disk, you should perform a standard database recovery and place the recovered database on a new disk. After doing so, you must stop the RDU server and adjust the BPR\_DATA and BPR\_DBLOG variables, in the <BPR\_HOME>/bpr\_definitions.sh file, to point to the new location of the database and database file.

**Note**

---

It is a good practice, to prevent failure situations, to monitor disk space utilization. Refer to the [“Monitoring Disk Space”](#) section on page 5-2 for additional information.

---