CHAPTER 10

Database Management

This chapter contains information on RDU database management and maintenance. The RDU database is the Broadband Access Center (BAC) central database. The Cisco BAC RDU requires virtually no maintenance other than to ensure availability of sufficient disk space. As the administrator, you must understand and be familiar with database backup and recovery procedures.

This chapter describes:

Understanding Failure Resiliency

The RDU database uses a technique known as Write-ahead logging to protect against database corruption that could result from unforeseen problems, such as an application crash, system failure, or power outage.

Write-ahead logging involves writing a description of any database change to a database log file before writing the change into the database files. This mechanism allows the repair of any incomplete database writes that can result from system failures.

The RDU server performs an automatic recovery each time it is started. During this recovery process, the database log files are used to synchronize the data with the database files. Database changes that were written into the database log, but not into the database, are written into the database during this automatic recovery.

In this way, write-ahead logging virtually guarantees that the database does not become corrupted when the RDU server crashes because the database is automatically repaired when the RDU server is restarted.

Write-ahead logging requires these conditions to work properly:

- You must set up the file system and physical storage so that they guarantee that the data is flushed to physical storage when requested. For example, a storage system with SDRAM-only write cache, which loses data during system failure, is not appropriate.

  However, a disk array with battery-backup write cache that guarantees that the data gets persisted, even in the event of a system failure, is acceptable. A system without battery-backed write cache should flush data disk when requested instead of performing in-memory data caching.
- You must set up the file system with a 8192-byte block size to match the RDU database block size. This setting is usually the default unless explicitly adjusted.

Database Files

The RDU database stores data in binary files using the file system you have mounted on the partition containing the files. It is essential to choose and configure a file system in a way that it is not susceptible to long recovery times after system failures.

Database files are vital to the operation of the RDU. Therefore, take extra precautions to safeguard them against accidental removal or other manual manipulation. Follow standard system administration practices to safeguard these important files.

For example, these files should always have permissions that allow only root user access. Additionally, it is a good practice to never log into your production system as a root user, but rather log in as a less privileged user and use the sudo command to perform tasks requiring root privileges.

Database Storage File

The RDU server stores its database in a file called bpr.db, which resides in the database directory. This directory is located in the BPR_DATA/rdu/db directory and is configured by specifying the BPR_DATA parameter during a component installation. See Changing Database Location, page 10-7, for additional information on moving the database.

Note The database file is normally accessed in a random fashion. You should therefore, select a disk with the fastest seek time and rotational access latency to obtain the best database performance.

Database Transaction Log Files

The RDU server stores database transaction logs in 10-MB files that are stored in the database log directory. You configure this directory during installation by specifying the BPR_DBLOG parameter. The log directory is located in the BPR_DBLOG/rdu/dblog directory. See Changing Database Location, page 10-7, for additional information on moving the transaction logs to a new directory.

Database log files are named in numeric sequence, starting at log.00000001, log.00000002, and so on.

Note The disk on which transaction logs are stored is usually accessed in a sequential manner, with data being appended to the log files. Select a disk that can efficiently handle this access pattern to achieve the best database performance. We recommend that you locate the database transaction logs directory on the fastest disk on the system. Also, ensure that 1 GB disk space is available.

Automatic Log Management

Database transaction logs files are used to store transaction data until that data is completely written into the database. After that, the transaction log data becomes redundant and the files are then automatically removed from the system. Under normal circumstances there should be only a few log files in the database transaction log directory. Over time, you will notice that older transaction logs disappear and newer ones are created.

⚠ **Caution** Database transaction logs are an integral part of the database. Manual deletion of transaction log files will result in database corruption.

Miscellaneous Database Files

The database directory contains additional files that are essential to database operation. These files, in addition to the rdu.db file, are found in the BPR_DATA/rdu/db directory and are copied as part of the database backup:

- DB_VERSION—Identifies the physical and logical version of the database and is used internally by the RDU.
- history.log—Used to log activity about essential database management tasks, such as automatic log file deletion, backup, recovery, and restore operations. In addition to providing useful historical information for the administrator, this log file is essential to RDU database operation.

Disk Space Requirements

The size of a fully populated database depends on a number of factors:

- Device objects that the RDU manages.
- Custom properties stored on each object.
- Device history records tracked for each device.

The approximate estimates for disk space required on each partition are:

- BPR_DATA, approximately 3 to 5 KB per device object
- BPR_DBLOG, at least 500 MB

⚠ **Caution** These numbers are provided as a guideline only and do not eliminate the need for normal system monitoring.

You can use the **disk_monitor.sh** tool to monitor available disk space and alert the administrator. See Using the disk_monitor.sh Tool, page 19-4, for additional information.

Handling Out of Disk Space Conditions

When the RDU server runs out of disk space, it generates an alert through the syslog facility and the RDU log. The RDU server then tries to restart automatically. When attempting to restart, the RDU server may again encounter the `out of disk space` error and attempt to restart again.

The RDU server continues trying to restart until free disk space becomes available. Once you free up some disk space on the disk that is operating near a limit, the next time the RDU restarts it will succeed.

If the size of your database grows beyond the capacity of the current disk partition, then you should move the database to a new disk or partition. For information on how to do this, see Changing Database Location, page 10-7.

Note It is a good practice to monitor disk space utilization to prevent failure. See Using the disk_monitor.sh Tool, page 19-4, for additional information.

Backup and Recovery

The RDU server supports a highly efficient backup process that can be performed without stopping the server or suspending any of its activities. Database backup and recovery involves these stages:

- Backup—Takes a snapshot of the RDU database from a live server.
- Recovery—Prepares the database snapshot for re-use.
- Restore—Copies the recovered database snapshot to the RDU server.

Automated tools are provided for each of these steps. You can use these tools in either interactive mode or silent mode, but you must have root privileges to use the tools.

Database Backup

Backup is the process of copying the database files into a backup directory. The files can then be compressed and placed on tape or other archive.

RDU database backup is highly efficient because it involves just copying files without interrupting server activity. However, because it involves accessing the RDU database disk, backup may adversely affect RDU performance. The opposite is also true; RDU activity happening during backup will adversely affect backup performance. Therefore, you should perform backups during off-peak hours.

Other than concurrent system activity, backup performance also depends on the underlying disk and file system performance. Essentially, backup will perform as fast as database files can be copied from source to target.

Use the **backupDb.sh** tool, in the BPR_HOME/rdu/bin directory, to perform database backups:

- To use this tool, you must provide the target directory where the backup files will be placed. This directory should be on a disk or partition that has available disk space equivalent to 120% of the current database file size.
- As illustrated in the following example, this tool automatically creates a time stamped subdirectory, under the directory you specify and places the backups there.

Examples

Here is an example of using the **backupDb.sh** tool:

```
# backupDb.sh /var/backup
```

where */var/backup* identifies the database backup directory.

In this example, all backup database files are stored in a directory called /var/backup/rdu-backup-09252002-130345. The last subdirectory (rdu-backup-20020925-130345) is automatically created with a current timestamp.

Note The timestamped subdirectory format is *rdu-backup-yyyyMMdd-HHmmss*. In this example, the subdirectory would be *rdu-backup-04272006-175430*, meaning that the directory contains a backup that was started at 5:54:30 pm on April 27, 2006.

The **backupDb.sh** tool also reports progress to the screen and logs its activity into *history.log*.

Note When using the **backupDb.sh** tool, you can use a **–help** option to obtain usage information. You can also use the optional **–nosubdir** flag to disable, if necessary, the automatic creation of the subdirectory.

Database Recovery

Database recovery is the process of restoring the database to a consistent state. Since backup is performed on a live RDU, the database can be changing while it is being copied. The database log files, however, ensure that the database can be recovered to a consistent state.

Recovery is performed on a snapshot of a database. In other words, this task does not involve touching the database on the live RDU server. The recovery task can be performed either immediately following a backup or prior to restoring the database to the RDU server.

Note We recommend that you perform database recovery immediately after each backup. This way, the backed up database can be more quickly restored in case of emergency.

The duration of database recovery depends on the number of database log files that were copied as part of the backup, which in turn depends on the level of RDU activity at the time of the backup. The more concurrent activity RDU experiences during the backup, the more transaction log files have to be copied as part of the backup and the longer is the recovery. Generally, recovering a database takes from 10 to 60 seconds per transaction log file.

Use the **recoverDb.sh** tool, located in the BPR_HOME/rdu/bin directory, to perform recovery of the snapshot of a database. When you use this tool, you must provide the location of the backup. This is also the directory in which the recovery will be performed.

Examples

Here is an example of using the **recoverDb.sh** tool:

```
# recoverDb.sh /var/backup/rdu-backup-20060427-130345
```

In this example, the snapshot located in the /var/backup/rdu-backup-20060427-130345 directory will be recovered to a consistent state. The progress of the recovery operation will appear on screen and the activity will be recorded in the history.log file in the snapshot directory.

Note When using the **recoverDb.sh** tool, you can use the **–help** option to obtain usage information on the tool.

Database Restore

Restoring the database is the process of copying the previously recovered database snapshot to the database location used by the RDU server. Only a database that has been previously recovered can be restored.

Since restoring the database means replacing the current RDU database, it is very important that you first properly remove and archive the old database.

Note Never delete the database you are replacing. You might need a copy of an old database to simplify future system diagnostics.

Use the **restoreDb.sh** tool, located in the BPR_HOME/rdu/bin directory, to replace the current RDU database with another database. When using this tool, you must specify an input directory. This directory must contain the recovered backup snapshot of the database to be restored to the RDU server.

Note Before running the **restoreDb.sh** tool, you must stop the RDU server. Also, remember to back up the database, then remove the database files from the rdu/db and the rdu/dblog directories.

Examples

Here is an example of running the **restoreDb.sh** tool:

```
# restoreDb.sh /var/backup/rdu-backup-20060427-130345
```

In this example, the database found in the /var/backup/rdu-backup-200604275-130345 directory is restored to the RDU server.

Note When using the **restoreDb.sh** tool, you can use the **–help** option to obtain usage information on the tool.

You must restart the RDU after the restore operation is completed. The RDU log file will contain successful startup messages.

This tool displays progress on the monitor and logs its activity in the history.log file.

Changing Database Location

You can move the database from one partition or disk to another on the same system. You might sometimes want to do this for administrative reasons. This process requires stopping the RDU server and the Cisco BAC process watchdog.

The process of changing the database location involves changing system parameters and copying the appropriate files to the new location. You can adjust one or both of the following parameters:

- *BPR_DATA*—This parameter is initially set during installation and points to a directory that is used to store the database, and many other important files, such as logs, configuration files, and so on.

  This directory also stores log data for the Cisco BAC process watchdog, the DPE (if installed on the same system), RDU, and SNMP agent, among others.

- *BPR_DBLOG*—This parameter is initially set during installation and points to the directory that stores database transaction log files.

The values for the above parameters are recorded in a file called BPR_DATA/bpr_definitions.sh. Any change to this file requires that you restart all BAC components running on the system.

To change the location of the database and transaction logs:

Step 1 Run the **/etc/init.d/bprAgent stop** command to stop the BAC process watchdog and all Cisco BAC components.

Step 2 Make a backup copy of BPR_HOME/bpr_definitions.sh file.

Step 3 Edit the file and change either or both the BPR_DATA and BPR_DBLOG parameters to new directories.

Step 4 Save the file.

Step 5 Copy or move the directory structure and contents of the original BPR_DATA and/or BPR_DBLOG directories to new location(s). If you make a copy, ensure that all file and directory permissions are preserved.

Step 6 Run the **/etc/init.d/bprAgent start** command to start the BAC process watchdog and all BAC components.

Step 7 Monitor the appropriate log files to ensure that all components have successfully started.