Transaction logging

Domino supports transaction logging for servers that run Domino 5 and later, and for databases that are in a Domino 5 or later on-disk structure.

Transaction logging captures all the changes made to a database and writes them to a transaction log. The logged transactions are then written to disk in a batch, either when resources are available or when scheduled.

A transaction is a related series of changes made to a database on a server. For example, opening a new document, adding text, and saving the document is one transaction. In this case, the transaction consists of three separate implicit API calls: NotesOpen, NoteUpdate, and NoteClose.

A transaction log is a record of changes made to Notes databases. The transaction log consists of log extents and the log control file (NLOGCTRL.LFH). A log extent is one of the log files into which the transaction logs are written. It has the form S*xxxxxxx*.TXN, where *x* character represents a seven-digit number that is unique to that server. Domino fills each extent sequentially before writing data to a new one. The records are secured using a proprietary byte-stream format. Each server has only one transaction log that captures all the changes to databases that are enabled for transaction logging.

Use transaction logging to:

- Schedule regular backups. Backups based on transaction logs are faster and easier than full database backups that do not use transaction logging.
- Recover from a media failure. If you have a media failure, you can restore the most recent full backup from tape, then use the transaction logs to add the data that was not written to disk.
- Recover from a system crash. When the server restarts, it runs through the end of the transaction logs and recovers any writes that were not made to disk at the time of the crash. Logged databases do not require a consistency check.
- Log the database views. You can avoid most view rebuilds.

To use all the features of transaction logging for backups and backup recovery, you need a third-party backup utility that uses the backup and recovery methods of the Domino C API Toolkit (Release 5 or later). For example, in the case of a media recovery, a database backup is taken with the third-party utility, while logging keeps track of updates to the database. When the database is then lost, the backup is brought up to current state by going through the transaction log and applying any updates which have happened to that databases since the database backup was taken.

Note that restart recovery does not require a third-party utility. In this case, logging goes on while updates are happening. When the server crashes then restarts, any updates which would have otherwise been lost are written to the database. This significantly reduces lost data and database corruption because of server crashes, and reduces overall restart time since the consistency check of databases is not required.

Understanding the database instance ID (DBIID)

When you enable transaction logging, Domino assigns a unique database instance ID (DBIID) to each Domino database. When Domino records a transaction in the log, it includes this DBIID. During recovery, Domino uses the DBIID to match transactions to databases.

Some database maintenance activities, such as using the Compact command with options, cause Domino to reconstruct the database in such a way that old transaction log records are no longer valid. When this happens, a new DBIID is assigned to the database. From that point on, all new transactions recorded in the log for that database use the new DBIID. After a database is assigned a new DBIID, take a new full backup of the database. The new full backup captures the database in its current state with the new DBIID. Then, if you have to restore the database, Domino needs only the new transactions that contain the new DBIID.

Domino assigns a new DBIID when:

- You enable transaction logging for the first time.
- You run the Compact task with an option -- for example, the option to reduce file size.
- You run the Fixup task on corrupted databases.
- You move a Domino database to a logged server.

See also

[How transaction logging works](#)
[Planning for transaction logging](#)
[Disabling transaction logging for a specific database](#)
[Using transaction logging for recovery](#)
[View logging](#)
[Transaction logging -- Troubleshooting](#)

Glossary   **Help on Help**

How transaction logging works

Following is a general example of transaction logging from both the administrator's and the employees' points of view.

The administrator enables transaction logging for all the databases on the servers. The administrator chooses the Archived logging style so that there is plenty of room for the transaction logs; uses a separate, mirrored device for safe and speedy storage of the transaction logs; and installs a backup utility to recover from media failures and any resulting corrupted databases.

The administrator backs up the transaction logs daily. This procedure doesn't take long because the administrator is backing up only the changes, rather than doing a full backup of all the databases on the server.

When the server crashes, it's down, but not for long. As the administrator restarts the server, it replays all the changes from the transaction logs to the databases. The server is soon back in business.

A few days later, there's a media failure. The administrator restores the corrupted databases from the most recent weekly backup and replays the changes.

The employees who use the databases do not notice any difference in how they do their work. They might notice, however, that servers are up and running more often and that there is less down time.

How changes are made to the database

Transaction logging posts all the database transactions to the log file, without waiting for the transaction to commit to disk. After being posted to the log file, the change is considered successful. The physical write process can wait until the server is less busy or occur at periodic intervals. The changes are written to disk in a batch.

What happens between the time when the transaction is posted to the log file and when the database is updated on the disk? Databases are cached in memory while they are open. The writes to the database happen to the in-memory copy of the database. They are then immediately sent to the transaction logs. Later, the memory-cached version of the database is posted to disk, updating the databases. Since the transaction log is sequential, there is no seek time, and only enough information is written to the logs to redo (or undo if necessary) the operation. In many cases, this is less information than the database write to disk.

If the database is not yet completely written to disk and you open it, you are opening the memory-cached version. If the server crashes before the version on disk has been updated with the changes, restarting the server applies the logs to the database during restart.

See also

Transaction logging
Planning for transaction logging

Planning for transaction logging

Transaction logging captures all the changes that are made to databases and writes them to a transaction log. The logged transactions are written to disk in a batch when resources are available or at specified intervals.

Use this checklist for your transaction logging planning.

- Allocate space for the log files. Use a dedicated, mirrored device, such as RAID level 1 with a dedicated controller for optimal performance and data integrity.
- Plan a backup strategy. Plan to archive the transaction logs daily using incremental backups. Schedule weekly full database backups. You will then be prepared if you have a media failure.
- Decide which servers and databases will use transaction logging. Transaction logging is available for servers running Domino 5 and later. Consider enabling transaction logging for all databases on the server.
- Select a Domino-compatible backup utility. The utility must be able to use the backup and recovery methods of the Domino C API Toolkit (Release 5 or later).
- Choose the logging style that fits your needs. Logging styles include archived, circular, and linear.
- Set up a Domino server for transaction logging.

Comparing transaction logging styles

There are three logging styles to choose from -- circular, linear, and archived. The logging style you choose is also dependent on your disk size and backup strategy.

With circular logging, Domino reuses a fixed amount of disk space (up to 4GB) for transaction logs. After the disk space is used up, Domino starts overwriting old transactions, starting with the oldest. When the space fills up, perform a backup on the databases. You may need to do daily backups to capture database changes before they are overwritten, depending on the server activity level. Use circular logging if the size of the log needed between full database backup intervals is less than 4GB.

Linear logging is like circular logging, except it allows more than 4GB. Use linear logging if the size of the log needed between full database backup intervals is greater than 4GB, and you are not using archive media.

Archived logging creates log files as needed. It simplifies backup and restoration, and provides online and partial backups. The log files are not overwritten until you archive them. With archived logging, you must have a backup utility to back up the filled log extents so that they are ready if needed. If you do not have a backup utility, the server continues to create log extents, fills up the disk space, and then panics.

See also

> Transaction logging
> How transaction logging works
> Disabling transaction logging for a specific database
> Using transaction logging for recovery
> Fixing corrupted databases
> Using Fixup
> Transaction logging -- Troubleshooting

Glossary   **Help on Help**

Setting up a Domino server for transaction logging

You can enable and set up transaction logging on any server.

1. Make sure that all the databases you want to log are in the Domino data directory, either at the root, or in a subdirectory.

2. From the Domino Administrator, click the Configuration tab, expand the Server section, and click "All Server Documents."

3. Select the Server Document for the Domino server you want to edit and then click Edit Server.

4. Click the Transactional Logging tab, complete these fields, and then save the document:

Field	Action
Transactional Logging*	Choose one: • Enabled -- To start transaction logging • Disabled (default) -- To not use transaction logging
Log path*	Enter the path name location of the transaction log. For best results, use a separate mirrored device, such as a RAID (Redundant Array of Independent Disks) level 0 or 1 device with a dedicated controller. This provides better performance and data integrity than using the default path (\LOGDIR) in the Domino data directory. **Note:** If the device is used solely for storing the transaction log, set the "Use all available space on log device" field to Yes.
Use all available space on log device	For circular and linear logging only. Choose one: • Yes -- To use all available space on the device for the transaction log. Choose Yes if you use a separate device dedicated to storing the log. • No -- To use the default or specified value in the "Maximum log space" field.
Maximum log space	For circular and linear logging only. The maximum size, in MB, for the transaction log. Default is 192MB. Maximum is 4096MB (4GB). Allocate a separate disk with at least 1024MB (1GB) of disk space for the transaction log. Domino formats at least 3 and up to 64 log files, depending on the maximum log space you allocate.
Automatic fixup of corrupt databases	Choose one: • Enabled (default) -- To run the Fixup task automatically if a database is corrupted and Domino cannot use the transaction log to recover it. Domino assigns a new DBIID and notifies the administrator that a new database backup is required. • Disabled -- To not run the Fixup task automatically. Domino notifies the administrator to run the Fixup task with the -J parameter on corrupted logged databases.
Runtime/Restart performance	This field controls how often Domino records a recovery checkpoint in the transaction log. This affects server performance as databases may be flushed from the cache to disk. To record a recovery checkpoint, Domino evaluates each active logged database to

	determine how many transactions would be necessary to recover each database after a system failure. When Domino completes this evaluation, it: • Creates a recovery checkpoint record in the transaction log that lists each open database and the starting point transaction needed for recovery • Forces database changes to be saved to disk if they have not been saved already Choose one: • Standard (default and recommended) -- To record checkpoints regularly. • Favor runtime -- To record fewer checkpoints. This option requires fewer system resources and improves server run-time performance but causes more of the log to be applied during restart. • Favor restart recovery time -- To record more checkpoints. This option improves restart recovery time because fewer transactions are required for recovery.
Logging style**	Choose one: • Circular (default) -- To re-use the log files and overwrite old transactions. • Archived (recommended) -- To re-use the log files after they are archived. A log file can be reused when it is inactive, which means that it does not contain any transactions necessary for a restart recovery. Use a third-party backup utility to copy and archive the existing log. When Domino using the existing file again to Start, Domino increments the log file name. If all the log files become inactive and are not archived, Domino creates additional log files. • Linear -- To re-use the log files and overwrite old transactions for log size greater than 4GB.

* If you change this field, you must restart the server so that the change takes effect..

** If you change this field, Domino assigns a new DBIID to each database. You must restart the server and perform another full backup.

See also

How transaction logging works
Planning for transaction logging
Detecting when a mail file exceeds its quota

Glossary   **Help on Help**

Disabling transaction logging for a specific database

After you set up transaction logging on a server, Domino logs all databases on that server. You can disable transaction logging of specific databases, but this practice is not recommended because if unlogged databases are corrupted during a system or media failure, you must run the Fixup task to recover the database.

To disable transaction logging for a specific database

1. Do one of the following to choose "Disable transaction logging":

- If you are creating a new database, use the Advanced Database Options dialog box.
- If you are working in an existing database, use the Advanced tab of the Database Properties box.
- In the Domino Administrator, select a database on the Files tab, choose Tools - Database - Advanced Properties.

2. Be sure that all users have closed the database.

3. Use the Dbcache command with the flush parameter to close the database in the database cache.

4. Open the database.

To reenable transaction logging for a specific database

Follow the steps above, but de-select "Disable transaction logging."

See also

Transaction logging
How transaction logging works
Planning for transaction logging
Using transaction logging for recovery
Compact options
Managing the database cache
Transaction logging -- Troubleshooting

Glossary   **Help on Help**

View logging

View logging provides a way to maintain consistent views in failure conditions and allows media recovery to update those views. View logging is transaction logging support for Notes views and folders. All updates to Notes views or folders are recorded in the transaction log for recovery purposes.

To enable view logging, you use Domino Designer. In Designer, open a view or folder, select the Advanced tab, and check "Logging - Include updates in transaction log."

Note If you enable view logging in a template, all databases created from that template and all databases whose designs are replaced from that template have those views logged.

See also

> Transaction logging
> How transaction logging works

Glossary   **Help on Help**

Using transaction logging for recovery

Transaction logging is an integral part of recovering from system and media failures. Using transaction logging provides insurance against system failure, but creating regular backups is essential so that you can recover data after a failure.

System failure recovery

A system failure causes the server to stop and requires you to restart the server. During restart, Domino automatically performs database recovery. The system uses the transaction logs to apply full transactions and undo partial transactions that were not written to disk for databases that were open during the system failure. Domino runs the Fixup task for:

- Databases in formats that are earlier than Domino 5
- Databases that are in Domino 5 format but have transaction logging disabled
- Corrupted databases, if you choose Yes for "Auto fixup of corrupt databases" in the Server document.

When you restart a server after a system failure, Domino automatically restores the affected databases.

Media failure recovery

A media failure causes databases to be damaged or lost. To recover, you use the third-party backup utility to restore database backups and transactions from the transaction log files. The backup utility you choose must use the backup and recovery methods of the Domino C API Toolkit (Release 5 or later).

For information on recovering after a media failure, see the documentation included with your backup utility.

See also

Fault recovery
Specifying a cleanup script for fault recovery

Glossary   **Help on Help**

Transaction logging -- Troubleshooting

These topics describe common problems with transaction logging. You can also search for solutions to common problems on the Lotus Support Services Web site at www.lotus.com/support.

Invalid transaction log path

If Domino cannot access the transaction log path, the server console displays error messages indicating: the invalid log path, databases requiring media recovery or Fixup, and a panic.

1. Check the log path to make sure it exists.

2. Check that the server has write access to the log path.

3. If the log path is correct and the device is good, restart the server. The problem should be fixed and you do not need to continue to step 4.

4. If log path is correct but the device is bad, replace the device on the log path, or edit the TRANSLOG_Path setting in NOTES.INI to point to a different log path.

>   **Note** If you edit the TRANSLOG_Path setting when you restart the server, be sure to make the same edit to the "Log path" field in the Server document. Otherwise, Domino reverts to the old path upon the next server restart.

5. Restart the server. Domino creates new log files and a control file, and assigns new DBIIDs to all Domino 5 or higher databases.

6. If "Automatic fixup of corrupt databases" is set to Yes in the Server document, the Fixup task runs on the databases that require media recovery or Fixup. Otherwise, you must run the Fixup task manually.

7. Perform full database backups.

Transaction log damaged or corrupted

If the transaction log appears to be damaged or corrupted, the server console displays error messages indicating: the log is damaged, databases requiring media recovery or Fixup, and a panic.

The error may occur because of a failed read from or write to the transaction log.

1. Restart the server to correct the error. If the damaged log error does not appear again, the log is not damaged.

2. Stop the server again so it shuts down cleanly.

3. Perform full database backups.

4. Restart the server.

>   If you continue to received the damaged log error, the active transaction log is damaged or corrupted.

5. Delete the transaction log files and the control file.

6. Restart the server. Domino creates new log files and a control file and assigns new DBIIDs to all Domino 5 or higher databases.

7. If "Automatic fixup of corrupt databases" is set to Yes in the Server document, the Fixup task runs on the databases that require media recovery or Fixup. Otherwise, you must run the Fixup task manually.

8. Perform full database backups.

If the error occurred during media recovery, an archived log file may be corrupted.

9. Restart the server to correct the problem, and then stop the server so it shuts down cleanly.

10. While the server is down, use the third-party backup utility to perform media recovery. If the archived log still cannot be used, allow database backups to be restored without the transactions in the corrupted log.

11. Perform full database backups.

12. Restart the server.

See also

[Troubleshooting the Domino system](#)

Glossary   **Help on Help**