



RSuite CMS

MarkLogic Database Administration

7 February

Notices

RSuite is a registered trade mark and the RSuite CMS logo is a trademark of Orbis Technologies, Inc.

MarkLogic and the MarkLogic logo are trade marks of MarkLogic Corporation.

This is a confidential document to be shared with only current RSuite customers and others designated by Orbis Technologies, Inc.

Copyright © 2017 Orbis Technologies, Inc. All rights reserved.

About this book

This manual contains procedures for backing up and restoring MarkLogic™ databases and other procedures and guidance for RSuite® customers configuring MarkLogic databases. It assumes the user is familiar with MarkLogic architecture and terminology.

Contents

- About this book.....iii
- Back up a database..... 1**
 - Before you begin 1
 - Immediate backup 1
 - Omitting forests 1
 - Perform the backup 1
 - Scheduled backup..... 2
- Restore a backed-up database 4**
 - Before you begin 4
 - Restore a backup 4
- Improving query performance 5**
- Triple store 6**
- Database Rebalancing..... 7**
- MarkLogic clusters 8**
 - Evaluator Node 8
 - Data Node 8
- Monitoring MarkLogic 10**

Back up a database

You can run a database backup immediately or schedule it for the future.

Before you begin

1. Ensure that the path to the directory that receives the backup exists for all servers that host any forests in the database. That is, all servers must have the same local directory defined, or all servers must have access to the network repository.
If necessary, create the directory.
2. Ensure the user ID running the MarkLogic™ server has Read and Write permissions to the directory.
The default user on UNIX is `daemon` and on Windows it is the local system user.

Immediate backup

Omitting forests

Before you start the backup, you have a chance to edit the list of the forests that will be included in the backup. If you deselect a forest, you might not have a completely consistent view of the database to restore. Deselect forests only if you are sure you understand the implications. To ensure the restored database is the same as it was when you backed it up, back up all of the forests associated with the database, including the Schemas and Security database forests.

Make a note of the forests it is safe to remove from the backup.

Perform the backup

Perform these steps to back up the database:

1. Log into the MarkLogic Admin interface as a user with the `admin` role.
2. Click on the **Databases** link in the left menu of the Admin Interface.
3. Click on the database name (e.g., `rsuite`) to back up, either from the tree menu or on the summary page.
4. Click on the **Backup/Restore** tab.
The Backup/Restore screen appears.
5. Enter the fully qualified path to the back-up directory in the Back up to Directory field.
6. To include replica forests in the backup, set Include Replica Forests to `TRUE`.
You have replica forests if you have configured forests for local-disk failover.
7. Click on **OK**.
8. For full backup, set Incremental Backup to `FALSE`.

9. Set Journal Archiving to FALSE.
The Confirm Backup screen appears and lists all the forests selected for backing up.
10. Deselect forests that you do not want to back up (see Omitting forests, above).
11. Click on **OK** to begin the backup.

After the backup is underway, the Admin interface redirects you to the Database Status page.

Scheduled backup

You can schedule database backups to occur once in the future or to be repeated at day-, week-, or month-based intervals. You can create any number of scheduled backups.

To schedule a backup, perform these steps:

1. Log on to the MarkLogic Admin interface as a user with the **admin** role.
2. Click on the **Databases** icon on the left tree menu.
3. On the tree menu or on the Database Summary page, select the database to back up.
The Database Configuration page appears.
4. Click on the **Scheduled Backup** link in the tree menu for the database.
The Scheduled Backup Configuration page appears.
5. On the Scheduled Backup Configuration page, delete scheduled backups you no longer need.
6. Click on the **Create** tab.
The Schedule a Database Backup page appears.
7. Ensure the user ID running the MarkLogic Server process has Read and Write permissions to the back-up directory. (See Before you begin, on page 1.)
8. Enter the fully qualified path to the back-up directory. (See Before you begin, on page 1.)
9. Choose a one-time backup or define a schedule:

For a one-time backup, enter the start date in **mm/dd/yyyy** format (for example, 07/29/2018 for July 29, 2018) and the time in 24-hour form, **hh:mm**.

To run backups at intervals of less than an hour, enter the number of minutes between backups.

To do backups at hourly intervals, enter the number of hours between backups. The Backup Minute setting specifies how many minutes after the hour the backup starts; it does not add to the interval. That is, the effective specification is in the form “Every 2 hours at 17 minutes after the hour”, not “Every 2 hours and 17 minutes”.

For daily intervals, enter the number of days between backups and the time of day.

To create a weekly schedule, enter the number of weeks between backups, check the days of the week on which to run a backup, and select the time of day for the backup to start.

To create a monthly schedule, enter the number of months between backups, select the day of the month (1-31), and select the time of day for the backup to start.

10. Enter the number of backups to keep. After that number have been saved, when a new backup is saved, the oldest backup is deleted. Enter 0 to keep an unlimited number of backups.
11. Choose whether you want to include the Security database, the Schemas database, and the Triggers database in this scheduled backup.
12. Choose whether you want the backups to include replica forests as well as the master forests.
You have replica forests if you have configured forests for local-disk failover.
13. For full backup, set Incremental Backup to FALSE.
14. Set Journal Archiving to FALSE.
15. Click on **OK** to create the backup schedule and put it into effect.

Restore a backed-up database

Before you begin

1. Determine which dated backup you wish to restore. The process defaults to the latest backup.
2. Ensure that the database being replaced has the same configuration settings as the backup being restored: index options, fragmentation, range indexes.
3. Make a note of any forests that should not be restored.

Careful: Not restoring a forest can degrade the integrity of the database. Deselect forests only if you are sure you understand the implications. The safest course is to restore all of the forests associated with the database, including the Schemas and Security database forests.

Restore a backup

To replace an entire database with a backup, perform these steps:

1. Log into the Admin interface as a user with the **admin** role.
2. Click on the **Databases** link in the left menu of the Admin interface.
3. On the tree menu or on the summary page, select the database to be replaced with a backup.
4. Select the Backup/Restore tab.
The Backup/Restore screen appears.
5. Enter the directory of the backup in the Restore from Directory field:
 - a. To restore the latest backup, enter the path to the main back-up directory
 - b. To restore a particular dated backup, include its date-stamp subdirectory.
6. To restore replica forests, set Include Replica Forests to TRUE.
 - This is possible only if the backup includes replica forests.
 - You have replica forests if you have configured forests for local-disk failover.
7. To restore the full backup, set Incremental Backup to FALSE.
8. Set journal archiving to FALSE.
9. Click on **OK**.
The Confirm Restore screen displays a list of the forests to be restored. By default, all the forests associated with a database are checked.
10. Uncheck forests that should not be restored. (See Before you begin on page 4.)
11. Click on **OK** to begin the restore operation.

Improving query performance

Perform these steps to configure MarkLogic to improve performance during queries:

1. Log into the Admin interface as a user with the **admin** role.
2. Click the **Databases** icon on the left tree menu.
3. Select a database.
4. Enable a configuration:

Directory Creation-automatic—creates a directory hierarchy to match the URI of a document or a directory that is created. This is the recommended setting, especially if you are accessing the database with a WebDAV server or if you are using it as a modules database.

Three-character Searches—creates indices to enable wildcard searches where the search pattern contains three or more consecutive non-wildcard characters, for example, “*abc”. MarkLogic recommends combining the three-character search indexes with a codepoint collation word lexicon, which makes all searches faster. When this parameter is TRUE, character searches are faster, but document loading is slower and the database files are larger.

For example, if the term “cater” appears in a document, entries for “cat”, “ate”, and “ter” are added to the index. MarkLogic also includes contextual information about whether a string was at the start or end of a word.

Trailing Wildcard Searches—creates indices to enable wildcard searches where the search pattern contains consecutive non-wildcard characters at the beginning of the string and the wildcard character at the end of the string (for example, “abc*”). When this parameter is TRUE, character searches are faster, but document loading is slower and the database files are larger.

For more information see the article. about wildcard searches on the MarkLogic Web site (<https://developer.marklogic.com/blog/wildcard-search-in-marklogic>).

Collection Lexicon—creates a lexicon of all of the collection URIs in the database. The collection lexicon allows you to quickly list all the collection URIs in the database and to perform lexicon-based queries on the URIs.

Maintain Last Modified—includes a timestamp on the properties document for each document in the database.

Triple store

MarkLogic can hold semantic triples as content in the database.

Perform these steps to configure a database for triple indexing:

1. Log into the Admin interface as a user with the **admin** role.
2. Click on the **Databases** link in the left menu of the Admin interface.
3. On the tree menu or on the summary page, click on a database name.
4. Change the value of Triple Index to TRUE.
5. Click on **OK**.

Database Rebalancing

MarkLogic has a rebalancer that keeps content distributed evenly among the database forests.

When a new forest is added to the database, the rebalancer moves documents to the new forest from the existing forests.

When a forest is retired, the rebalancer removes all of the documents from that forest and distributes them among the remaining forests.

For more information, see the article on Database rebalancing in the MarkLogic Web site (<https://docs.marklogic.com/guide/admin/database-rebalancing>).

MarkLogic clusters

In MarkLogic, when the volume of the data grows or requests to the application server increase, the number of MarkLogic servers needs to be increased. These servers can be joined in a cluster. Servers in a cluster are called “nodes”.

A MarkLogic Server cluster uses a “shared-nothing architecture”:

- no single server is in charge
- each server communicates with every other server
- each node in the cluster maintains its own copy of the configuration
- every database in the cluster, including the security database, is available to every node in the cluster

Servers can be added as demand increases. Hence, the shared-nothing architecture is highly scalable and can maintain high availability.

Although nodes in a cluster are very similar, they are distinguished by the roles they play, either evaluator node (e-node) or data node (d-node).

Evaluator Node

E-nodes evaluate XQuery programs, XCC/XDBC requests, WebDAV requests, and other server requests. If an e-node request does not require any data from a forest, it is processed entirely on the e-node.

If the request needs data, such as a document in a database, it communicates with d-nodes to obtain the data. Once it gets the content from the d-node, the e-node performs the filter portion of query processing and sends the results to the application.

Data Node

D-nodes maintain transactional integrity during insert, update, and delete operations. This transactional integrity includes forest journaling, forest recovery, back-up operations, and on-disk forest management. D-nodes also maintain indexes, retrieve content, and optimize storage in forests.

D-nodes service e-nodes when the e-nodes require content from a forest. A d-node gets the communication from an e-node, then sends the results of the index resolution back to the e-node. The d-node part of the request includes the index resolution portion of query processing. Also, each d-node performs merges for any forests hosted on that node.

It is possible for a server to act as both an e-node and a d-node. This of course is the case in a single-server installation. In a cluster, any number of servers can perform both e-node and d-node activities. In large configurations, however, it is usually better for e-nodes and d-nodes to be separate servers in the cluster.

MarkLogic Database Administration for RSuite CMS

E-nodes can be under load balancer for handling the requests to improve performance.

For more information, see the article on Clustering and Caching (<https://docs.marklogic.com/guide/concepts/clustering>) on the MarkLogic Web site.

Monitoring MarkLogic

The MarkLogic Web site has articles about monitoring activity on the MarkLogic Server:

Monitoring History Dashboard

(https://docs.marklogic.com/guide/performance/status#id_83642)

Server Logs (https://docs.marklogic.com/guide/performance/status#id_72340)

Status Screens in the Admin Interface

(https://docs.marklogic.com/guide/performance/status#id_83500)

Create Your Own Server Reports

(https://docs.marklogic.com/guide/performance/status#id_80312)