

Administrating MarkLogic Server

MarkLogic 11

Publication date 2024-04-17
Copyright © 2024 Progress Software Corporation

All Rights Reserved

Table of Contents

1. Introduction	13
1.1. Objectives	13
1.2. Audience	13
1.3. Scope and Requirements	13
1.4. Architecture Overview	14
2. Administrative (Admin) Interface	16
2.1. Overview of the Admin Interface	16
2.2. Accessing the Admin Interface	16
2.3. Logging Off the Admin Interface	17
2.4. Creating and Managing Administrators	17
2.5. The admin-ui-user Role	17
3. Common Administrative Procedures	18
3.1. Installing and Upgrading MarkLogic Server	18
3.2. Starting MarkLogic Server	18
3.3. Stopping MarkLogic Server	18
3.3.1. From the Admin Interface	18
3.3.2. With a System Command	19
3.3.3. With a Server-Side JavaScript	19
3.3.4. With an XQuery Script	19
3.4. Restarting MarkLogic Server	19
3.4.1. From the Admin Interface	19
3.4.2. With a Server-Side JavaScript	20
3.4.3. With an XQuery Script	20
3.5. Creating and Configuring Forests and Databases	20
3.6. Creating and Configuring App Servers	20
3.7. Setting Up Users, Roles, Privileges, and Permissions	21
3.8. Loading Content into a Database	21
3.9. Running the XQuery Use Cases and Building Simple Applications	21
3.10. Backing Up and Restoring Data	21
3.11. Monitoring and Tuning Performance	22
3.12. Scripting and Scheduling Administrative Tasks	22
3.13. Configuring Clusters, Groups, and Failover	22
4. Clusters	24
4.1. Overview of Cluster Configuration	24
4.2. OpenSSL FIPS 140-2 Mode	24
4.3. Procedures for Configuring Clusters	24
4.3.1. Configuring OpenSSL FIPS 140-2 Mode	24
4.3.2. Configuring Simple Cluster Encryption	24
4.3.3. Coupling Clusters	27
4.4. Running Behind a Load Balancer or Reverse Proxy	29
4.5. Configuring a MarkLogic Application Message and Banner	30
4.5.1. Example Configuration	31
4.5.2. Configuration Reference	31
4.5.3. Example: Creating a New Configuration Document	32
4.5.4. Example: Activating/Deactivating a Configuration	32
4.5.5. Example: Modifying the Notification Dialog Text	32
4.5.6. Example: Modifying the Banner Text	33
5. Groups	34
5.1. Overview of Groups	34
5.2. Example	34
5.3. Procedures for Configuring and Managing Groups	35
5.3.1. Creating a New Group	35
5.3.2. Group Settings	36

- 5.3.3. Enabling SSL Communication over XDQP 40
- 5.3.4. Configuring an SMTP Server 43
- 5.3.5. Restarting All Hosts in a Group 43
- 5.3.6. Deleting a Group 43
- 5.4. App Server Status Page 44
- 5.5. Access the App Server Status Page 44
- 6. HTTP Servers 45
 - 6.1. Creating a New HTTP Server 45
 - 6.2. Setting Output Options for an HTTP Server 47
 - 6.3. Viewing HTTP Server Settings 48
 - 6.4. Deleting an HTTP Server 48
 - 6.5. Canceling a Request 49
- 7. XDBC Servers 50
 - 7.1. Creating a New XDBC Server 50
 - 7.2. Setting Output Options for an XDBC Server 52
 - 7.3. Viewing XDBC Server Settings 53
 - 7.4. Deleting an XDBC Server 53
 - 7.5. Canceling a Request 53
- 8. WebDAV Servers 54
 - 8.1. Accesses a Database for Read and Write, Not XQuery Execution 54
 - 8.2. WebDAV Server Security 54
 - 8.3. Directories 55
 - 8.3.1. Automatic Directory Creation in a Database Settings 55
 - 8.3.2. Properties and URIs of Directories 56
 - 8.4. Server Root Directory 56
 - 8.5. Documents in a WebDAV Server 57
 - 8.6. Procedures for Creating and Managing WebDAV Servers 57
 - 8.6.1. Creating a New WebDAV Server 57
 - 8.6.2. Setting Output Options for a WebDAV Server 59
 - 8.6.3. Viewing WebDAV Server Settings 59
 - 8.6.4. Deleting a WebDAV Server 60
 - 8.6.5. Canceling a Request 60
 - 8.7. WebDAV Clients 60
 - 8.7.1. Tested WebDAV Clients 60
 - 8.7.2. General Steps to Connect to a Server 61
 - 8.7.3. Connecting to a Web Folder Using Windows 10 File Explorer 62
 - 8.7.4. Connecting to a Web Folder Using Windows 9 or Earlier File Explorer 62
 - 8.8. Example: Setting Up a WebDAV Server to Add or Modify Documents Used by Another Server 63
- 9. ODBC Servers 65
 - 9.1. Creating a New ODBC Server 66
 - 9.2. Setting Output Options for an ODBC Server 69
 - 9.3. Viewing ODBC Server Settings 69
 - 9.4. Deleting an ODBC Server 69
 - 9.5. Canceling a Request 69
 - 9.6. ODBC Request Monitoring and Cancellation 70
- 10. Auditing Events 71
 - 10.1. Overview of Auditing 71
 - 10.1.1. Audit Log Files 71
 - 10.1.2. Restricting Audit Events 71
 - 10.1.3. Audit Successful, Unsuccessful, or Both Types of Events 72
 - 10.1.4. Enabled at the Group Level 72
 - 10.2. Auditable Events 72
 - 10.2.1. Audit Log Content 75
 - 10.2.2. Sample Audit Logs 76

10.3. Configuring Auditing for a Group	76
10.3.1. Enabling Auditing for a Group	76
10.3.2. Disabling Auditing for a Group	76
10.3.3. Configuring Auditing to Audit Certain Events and Set Up Certain Restrictions	77
11. Managing User Requests and Monitoring Login Attempts	78
11.1. Managing Concurrent User Requests	78
11.1.1. Limiting Concurrent Requests with User Request Limits	78
11.1.2. Configuring User Concurrent Request Controls	78
11.2. Setting Request Blackouts on an App Server	78
11.2.1. Configuring Request Blackouts	78
11.2.2. Deleting Request Blackouts	79
11.3. Storing and Monitoring the Last User Login Attempt	79
11.3.1. Storing Last User Login Information in a Last-Login Database	79
11.3.2. Configuring User Login Monitoring	79
11.3.3. Displaying the Last Login Information	80
12. Databases	81
12.1. Understanding Databases	81
12.1.1. Schemas and Security Databases	81
12.1.2. Modules Database	81
12.1.3. Triggers Database	82
12.1.4. Database Settings	82
12.1.5. Example of Databases in MarkLogic Server	87
12.2. Creating a New Database	88
12.3. Attaching and/or Detaching Forests to/from a Database	89
12.4. Viewing Database Settings	89
12.5. Loading Documents into a Database	89
12.6. Merging a Database	90
12.7. Reindexing a Database	90
12.8. Clearing a Database	90
12.9. Disabling a Database	91
12.10. Deleting a Database	91
12.11. Checking and Setting Permissions for a Document in a Database	92
13. Word Query Database Settings	93
13.1. Understanding the Word Query Configuration	93
13.1.1. Overview of Configuration Options	93
13.1.2. Understanding Which Elements are Included and Excluded	93
13.1.3. Adding a Weight to Boost or Lower the Relevance of an Included Element ...	95
13.1.4. Specifying an Attribute Value for an Included Element	96
13.1.5. Understanding the Index Option Configuration	96
13.2. Configuring Customized Word Query Settings	96
14. Fields Database Settings	98
14.1. Overview of Fields	98
14.2. Understanding Field Configurations	98
14.2.1. Overview of Field Configuration Options	98
14.2.2. Root and Path Fields	99
14.2.3. Metadata Fields	104
14.2.4. Understanding the Index Option Configuration	104
14.3. Field Word Lexicons and Field Value Lexicons	105
14.4. Configuring Fields	105
14.4.1. Configuring a New Path or Root Field	105
14.4.2. Configuring a New Metadata Field	106
14.4.3. Modifying an Existing Field	107
14.4.4. Creating a Range Index on a Field	107
15. Understanding and Controlling Database Merges	108

- 15.1. Overview of Merges: Merges are Good 108
 - 15.1.1. Dynamic and Self-Tuning 108
 - 15.1.2. What Happens During a Merge 108
 - 15.1.3. Dangers of Disabling Merges 108
 - 15.1.4. Merges Will Change Scores 109
- 15.2. Setting Merge Policy 109
 - 15.2.1. Overview of the Merge Policy Controls 109
 - 15.2.2. Configuring the Merge Policy 110
- 15.3. Blackout Periods for Merges 111
 - 15.3.1. Understanding Merge Blackouts 111
 - 15.3.2. Configuring Merge Blackout Periods 111
 - 15.3.3. Deleting Merge Blackout Periods 111
- 15.4. Merges and Point-in-Time Queries 112
- 15.5. Setting a Negative Merge Timestamp to Preserve Fragments for a Rolling Window of Time 112
- 15.6. Monitoring a Merge 112
 - 15.6.1. Messages in the ErrorLog.txt File 112
 - 15.6.2. Database Status Page 113
- 15.7. Explicit Merge Commands 113
 - 15.7.1. Manually Initiating a Merge 113
 - 15.7.2. Canceling a Merge 114
- 15.8. Configuring Merge Policy Rules 114
 - 15.8.1. Determining the Baseline for Your Merges 114
 - 15.8.2. If You Want to Reduce the Number of "Large" Merges 115
 - 15.8.3. Other Solutions 117
- 16. Database Rebalancing 118
 - 16.1. Overview of the Database Rebalancer 118
 - 16.2. Rebalancer Trigger Events 119
 - 16.3. Rebalancer Document Assignment Policies 119
 - 16.3.1. Bucket Assignment Policy 119
 - 16.3.2. Segment Assignment Policy 121
 - 16.3.3. Statistical Assignment Policy 121
 - 16.3.4. Range Assignment Policy 122
 - 16.3.5. Query Assignment Policy 124
 - 16.3.6. Legacy Assignment Policy 125
 - 16.3.7. Summary of Assignment Policies 126
 - 16.4. How the Rebalancer Moves Documents 126
 - 16.4.1. How Data is Moved When a Forest is Attached to the Database 127
 - 16.4.2. How Data is Moved When a Forest is Retired from the Database 127
 - 16.5. Configuring the Rebalancer on a Database 127
 - 16.6. Configuring the Rebalancer on a Forest 127
 - 16.7. Retiring a Forest from the Database 128
 - 16.8. Checking the Rebalancer Status 128
 - 16.9. How the Rebalancer Interacts with Other Database and Forest Settings 128
 - 16.9.1. Database Replication 128
 - 16.9.2. Restoring a Database from a Backup 129
 - 16.9.3. Tiered Storage 129
 - 16.9.4. Fast Locking 129
 - 16.9.5. Delete-Only and Read-Only Forests 129
 - 16.10. Rebalancer Settings after Upgrading from an Earlier Release 130
- 17. Tiered Storage 131
 - 17.1. Terms Used in This Section 131
 - 17.2. Overview of Tiered Storage 132
 - 17.3. Range Partitions 134
 - 17.4. Query Partitions 135

17.5. Partition Migration	136
17.6. Configuring a Database with Range Partitions	138
17.6.1. Defining a Range Partition Key	139
17.6.2. Creating Range Partitions	140
17.7. Configuring a Database with Query Partitions	141
17.7.1. Creating Query Partitions	142
17.7.2. Setting the Query Assignment Policy for the Query Partition	143
17.7.3. Isolating a Query Partition	144
17.7.4. Look Up Partitions Queries	145
17.8. Overview of the Tiered Storage REST API	145
17.8.1. Asynchronous Operations	146
17.8.2. Privileges	146
17.8.3. /manage/v2/databases/{id name}/partitions	146
17.8.4. /manage/v2/databases/{id name}/partitions/{name}	147
17.8.5. /manage/v2/databases/{id name}/partitions/{name}/properties	147
17.8.6. /manage/v2/databases/{id name}/partition-queries	147
17.8.7. /manage/v2/databases/{id name}/partition-queries/{partition-number}	148
17.8.8. /manage/v2/databases/{id name}/partition-queries/{partition-number}/ properties	148
17.8.9. /manage/v2/forests	148
17.8.10. /manage/v2/forests/{id name}	148
17.8.11. /manage/v2/forests/{id name}/properties	149
17.9. Common Forest and Partition Operations	149
17.9.1. Viewing Partitions	149
17.9.2. Migrating Forests and Partitions	150
17.9.3. Resizing Partitions	151
17.9.4. Transferring Partitions between Databases	152
17.9.5. Combining Forests	152
17.9.6. Retiring Forests	152
17.9.7. Taking Forests and Partitions Online and Offline	153
17.9.8. Setting the updates-allowed State on Partitions	153
17.9.9. Deleting Partitions	154
17.10. Partitions with Forest-Level Failover	154
18. Super Databases and Clusters	156
18.1. Overview	156
18.2. Creating a Super-database	158
18.3. Creating a Super-cluster	158
18.4. Viewing Super-databases and Sub-databases	158
19. Backing Up and Restoring a Database	160
19.1. Backup and Restore Overview	160
19.1.1. Consistent, Database-Level Backup	161
19.1.2. Admin Interface	162
19.1.3. Backup and Restore Transactions	162
19.1.4. Backup Directory Structure	162
19.1.5. Phases of Backup or Restore Operation	164
19.1.6. Notes about Backup and Restore Operations	165
19.2. Backing Up Databases with Journal Archiving	166
19.3. Incremental Backup	168
19.3.1. Including New Forests in Incremental Backups	169
19.3.2. Using Journal Archiving with Incremental Backups	169
19.4. Backing Up a Database	169
19.4.1. Backing Up a Database Immediately	169
19.4.2. Scheduling a Database Backup	171
19.5. Restoring a Database from a Backup	172
19.5.1. Admin Interface for Database Restore	172

19.5.2. Restoring a Database without Journal Archiving	173
19.5.3. Restoring Databases with Journal Archiving	175
19.5.4. Restoring from an Incremental Backup with Journal Archiving	176
19.5.5. Restoring to the Safe Timestamp	177
19.5.6. Restoring to a Specific Timestamp	178
19.5.7. Restoring Based on Sample Documents	179
19.5.8. Restoring a Reconfigured Database	179
19.6. Backing Up and Restoring a Database Following Local Disk Failover	182
20. Rolling Upgrades	186
20.1. Understanding Rolling Upgrades	186
20.1.1. When Cluster Has Nodes at Different Software Version Levels	187
20.1.2. Rolling Upgrade Process	187
20.1.3. Rolling Upgrade Status in the Admin Interface	187
20.1.4. Effective Version and Software Version	188
20.2. Example—Rolling Upgrade	188
20.3. Performing Rolling Upgrades	191
20.3.1. Upgrading an EC2 Instance	191
20.3.2. Rolling Upgrades Using XQuery	194
20.3.3. Rolling Upgrades on Both Production and DR Clusters	195
20.4. Rolling Back a Partial Upgrade	195
20.5. APIs for Rolling Upgrades	195
20.5.1. Admin APIs	195
20.5.2. REST Management APIs	195
20.6. Interaction with Other MarkLogic Features	195
20.7. Important Points to Note before Performing Rolling Upgrades	196
20.8. Other Upgrade Options	196
21. Hosts	197
21.1. Adding a Host to a Cluster	197
21.2. Changing the Group of the Host	197
21.3. Shutting Down or Restarting a Host	197
21.4. Clearing a Forest on a Host	198
21.5. Deleting a Forest on a Host	198
21.6. Leaving the Cluster	198
21.7. Displaying License Options	199
21.8. Changing the License Key For a Host	200
21.9. Rolling Back a Transaction	200
22. Forests	202
22.1. Understanding Forests	202
22.2. Creating a Forest	202
22.3. Making a Forest Delete-Only	204
22.4. Making a Forest Read-Only	205
22.5. Attaching and Detaching Forests Using the Forest Summary Page	206
22.6. Making Backups of a Forest	207
22.6.1. Backing Up a Forest	207
22.6.2. Scheduling a Forest Backup	208
22.7. Restoring a Forest	208
22.8. Rolling Back a Forest to a Point In Time	209
22.9. Merging a Forest	209
22.10. Clearing a Forest	209
22.11. Disabling a Forest	209
22.12. Deleting a Forest from a Host	210
22.13. Rolling Back a Prepared XA Transaction Branch	210
23. Security Administration	212
23.1. Security Entities	212
23.2. Users	214

23.2.1. Creating a User	215
23.2.2. Viewing a User Configuration	215
23.2.3. Modifying a User Configuration	215
23.2.4. Deleting a User	216
23.3. Roles	216
23.3.1. Creating a Role	217
23.3.2. Viewing a Role	217
23.3.3. Modifying a Role Configuration	217
23.3.4. Deleting a Role	218
23.4. Execute Privileges	218
23.4.1. Creating an Execute Privilege	218
23.4.2. Execute Privilege: grant-my-privilege	219
23.4.3. Viewing an Execute Privilege	219
23.4.4. Modifying an Execute Privilege	219
23.4.5. Deleting an Execute Privilege	220
23.5. URI Privileges	220
23.5.1. Creating a URI Privilege	220
23.5.2. Viewing a URI Privilege	221
23.5.3. Modifying a URI Privilege	221
23.5.4. Deleting a URI Privilege	221
23.6. Amps	221
23.6.1. Creating an Amp	222
23.6.2. Viewing an Amp	222
23.6.3. Modifying an Amp	223
23.6.4. Deleting an Amp	223
23.7. Protected Collections	223
23.7.1. Creating a Protected Collection	224
23.7.2. Viewing a Protected Collection	224
23.7.3. Removing a Permission from a Protected Collection	224
23.7.4. Deleting a Protected Collection	224
23.8. Certificate Templates	224
23.9. Realm	225
23.9.1. Setting the Realm	225
23.9.2. Changing the Realm	225
24. Text Indexing	227
24.1. Text Indexes	227
24.1.1. Understanding the Text Index Settings	227
24.1.2. Viewing Text Index Configuration	231
24.1.3. Configuring Text Indexes	231
24.2. Phrasing and Element-Word-Query Boundary Control	231
24.2.1. Phrasing Control	231
24.2.2. Element Word Query Throughs	232
24.2.3. Procedures	233
24.3. Query Behavior with Reindex Settings Enabled and Disabled	234
24.3.1. Understanding the Reindexer Enable Settings	234
24.3.2. Query Evaluation According to the Lowest Common Denominator	235
24.3.3. Reindexing Does Not Apply to Point-In-Time Versions of Fragments	235
24.3.4. Example Scenario	235
25. Range Indexes and Lexicons	237
25.1. Understanding Range Indexes	237
25.2. Using Range Indexes for Value Lexicons	239
25.3. Understanding Word Lexicons	239
25.4. Understanding Path Range Indexes	240
25.4.1. Limitations on Index Path Expressions	240
25.4.2. Examples of Index Path Expressions	240

- 25.4.3. Testing the Validity of an Index Path Expression 241
 - 25.4.4. Using Namespace Prefixes in Index Path Expressions 241
 - 25.5. Viewing Element Range Index Settings 241
 - 25.6. Defining Element Range Indexes 242
 - 25.7. Viewing Attribute Range Index Settings 243
 - 25.8. Defining Attribute Range Indexes 243
 - 25.9. Viewing Path Range Index Settings 244
 - 25.10. Defining Namespace Prefixes Used in Path Range Indexes and Fields 244
 - 25.11. Defining Path Range Indexes 244
 - 25.12. Viewing Element Range Index Settings 245
 - 25.13. Defining Element Word Lexicons 245
 - 25.14. Viewing Attribute Word Lexicon Settings 246
 - 25.15. Defining Attribute Word Lexicons 246
 - 25.16. Defining Value Lexicons 247
 - 25.17. Deleting Range Indexes or Lexicons 247
 - 25.18. Defining Field Range Indexes 248
- 26. Fragments 249
 - 26.1. Choosing a Fragmentation Strategy 249
 - 26.1.1. Fragment Roots 250
 - 26.1.2. Fragment Parents 250
 - 26.2. Defining Fragment Roots 250
 - 26.3. Defining Fragment Parents 251
 - 26.4. Viewing Fragment Rules 251
 - 26.5. Deleting Fragment Rules 252
- 27. Namespaces 253
 - 27.1. Defining Namespaces for a Group 253
 - 27.2. Defining Namespaces for an HTTP, ODBC, or XDBC Server 253
 - 27.3. Viewing Namespace Settings for a Group 254
 - 27.4. Viewing Namespace Settings for an HTTP, ODBC, or XDBC Server 254
 - 27.5. Deleting Namespaces for a Group 254
 - 27.6. Deleting Namespaces for an HTTP, ODBC, or XDBC Server 254
- 28. Understanding and Defining Schemas 255
 - 28.1. Understanding Schemas 255
 - 28.2. Procedures For Defining Schemas 256
 - 28.2.1. Adding a Schema Definition for a Group 256
 - 28.2.2. Adding a Schema Definition for an HTTP, ODBC, or XDBC Server 256
 - 28.2.3. Viewing Schema Definitions for a Group 257
 - 28.2.4. Viewing Schema Definitions for an HTTP, ODBC, or XDBC Server 257
 - 28.2.5. Deleting a Schema Definition for a Group 258
 - 28.2.6. Deleting a Schema Definition for an HTTP, ODBC, or XDBC Server 258
- 29. Log Files 259
 - 29.1. Application and System Log Files 259
 - 29.2. Understanding the Log Levels 259
 - 29.3. Configuring System Log Files 259
 - 29.4. Configuring Application Log Files 260
 - 29.5. Viewing the System Log 260
 - 29.6. Viewing the Application and System File Logs 261
 - 29.7. Viewing Access Log Files 261
 - 29.8. Viewing Crash Log Files 262
- 30. Scheduling Tasks 263
 - 30.1. Understanding Scheduled Tasks 263
 - 30.2. Scheduling a Module for Invocation 263
 - 30.3. Selecting a Task Type 264
 - 30.3.1. Scheduling per Minute 264
 - 30.3.2. Scheduling per Hour 264

30.3.3. Scheduling per Day and Time	265
30.3.4. Scheduling per Week, Day, and Time	265
30.3.5. Scheduling per Month, Day, and Time	266
30.3.6. Scheduling Once Invocation on a Calendar Date and Time	266
31. Appendix A: 'Hot' versus 'Cold' Admin Tasks	268
31.1. Groups	268
31.2. HTTP, ODBC, XDBC, and WebDAV Servers	269
31.3. Databases	269
31.4. Hosts	269
31.5. Forests	269
31.6. Mimetypes	269
31.7. Security	270
32. Appendix B: Pre-defined Execute Privileges	271
33. Appendix C: Pre-defined Roles	303
33.1. admin	303
33.2. admin-builtins	303
33.3. admin-configuration-delete	304
33.4. admin-configuration-read	304
33.5. admin-configuration-write	304
33.6. admin-default	304
33.7. admin-default-internal	304
33.8. admin-module-internal	304
33.9. admin-module-read-internal	304
33.10. admin-module-read-invoke	304
33.11. admin-transform	305
33.12. admin-ui-user	305
33.13. alert-admin	305
33.14. alert-execution	305
33.15. alert-internal	305
33.16. alert-user	305
33.17. app-builder	305
33.18. app-builder-internal	305
33.19. app-user	305
33.20. application-plugin-registrar	305
33.21. appservices-internal	306
33.22. cpf-restart	306
33.23. custom-dictionary-admin	306
33.24. custom-dictionary-user	306
33.25. custom-language-admin-read	306
33.26. custom-language-admin-write	306
33.27. dls-admin	306
33.28. dls-internal	306
33.29. dls-user	306
33.30. domain-management	307
33.31. filesystem-access	307
33.32. flexrep-admin	307
33.33. flexrep-eval	307
33.34. flexrep-internal	307
33.35. flexrep-user	307
33.36. flexrep-user-change	307
33.37. graphql-internal	307
33.38. hadoop-internal	307
33.39. hadoop-user-all	308
33.40. hadoop-user-read	308
33.41. hadoop-user-write	308

33.42. harmonized-reader	308
33.43. harmonized-updater	308
33.44. healthcheck-user	308
33.45. infostudio-admin-internal	308
33.46. infostudio-internal	309
33.47. infostudio-user	309
33.48. manage	309
33.49. manage-admin	309
33.50. manage-admin-internal	310
33.51. manage-internal	310
33.52. manage-schematron-user	310
33.53. manage-user	310
33.54. merge	310
33.55. network-access	310
33.56. optic-reader-internal	310
33.57. ort-user	311
33.58. pii-reader	311
33.59. pipeline-execution	311
33.60. pipeline-management	311
33.61. pki	311
33.62. plugin-internal	311
33.63. ps-internal	311
33.64. ps-user	311
33.65. qconsole-internal	311
33.66. qconsole-user	311
33.67. query-view-admin	312
33.68. redaction-internal	312
33.69. redaction-user	312
33.70. rest-admin	312
33.71. rest-admin-internal	312
33.72. rest-extension-user	312
33.73. rest-internal	312
33.74. rest-reader	312
33.75. rest-reader-internal	312
33.76. rest-writer	312
33.77. rest-writer-internal	312
33.78. search-internal	312
33.79. security	312
33.80. security-internal	314
33.81. sparql-update-user	314
33.82. sql-execution	314
33.83. tde-admin	314
33.84. tde-view	314
33.85. temporal-admin	314
33.86. temporal-internal	314
33.87. tiered-storage-admin	314
33.88. tiered-storage-internal	315
33.89. trigger-management	315
33.90. view-admin	315
33.91. view-admin-internal	315
33.92. welcome-internal	315
33.93. xa	315
33.94. xa-admin	315
33.95. xinclude	316
34. Technical support	317

35. Copyright 318

1. Introduction

MarkLogic Server is a powerful NoSQL database for harnessing your digital content base, complete with Enterprise features demanded by real world, mission-critical applications. MarkLogic enables you to build complex applications that interact with large volumes of content in XML, SGML, HTML, JSON, and other popular content formats. The unique architecture of MarkLogic ensures that your applications are both scalable and high-performance, delivering query results at search-engine speeds while providing transactional integrity over the underlying content repository. MarkLogic can be configured for a distributed environment, enabling you to scale your infrastructure through hardware expansion.

1.1. Objectives

This document describes administrative tasks required to manage the operation of MarkLogic on your system.

1.2. Audience

This document is intended for a technical audience and MarkLogic system administrators.

1.3. Scope and Requirements

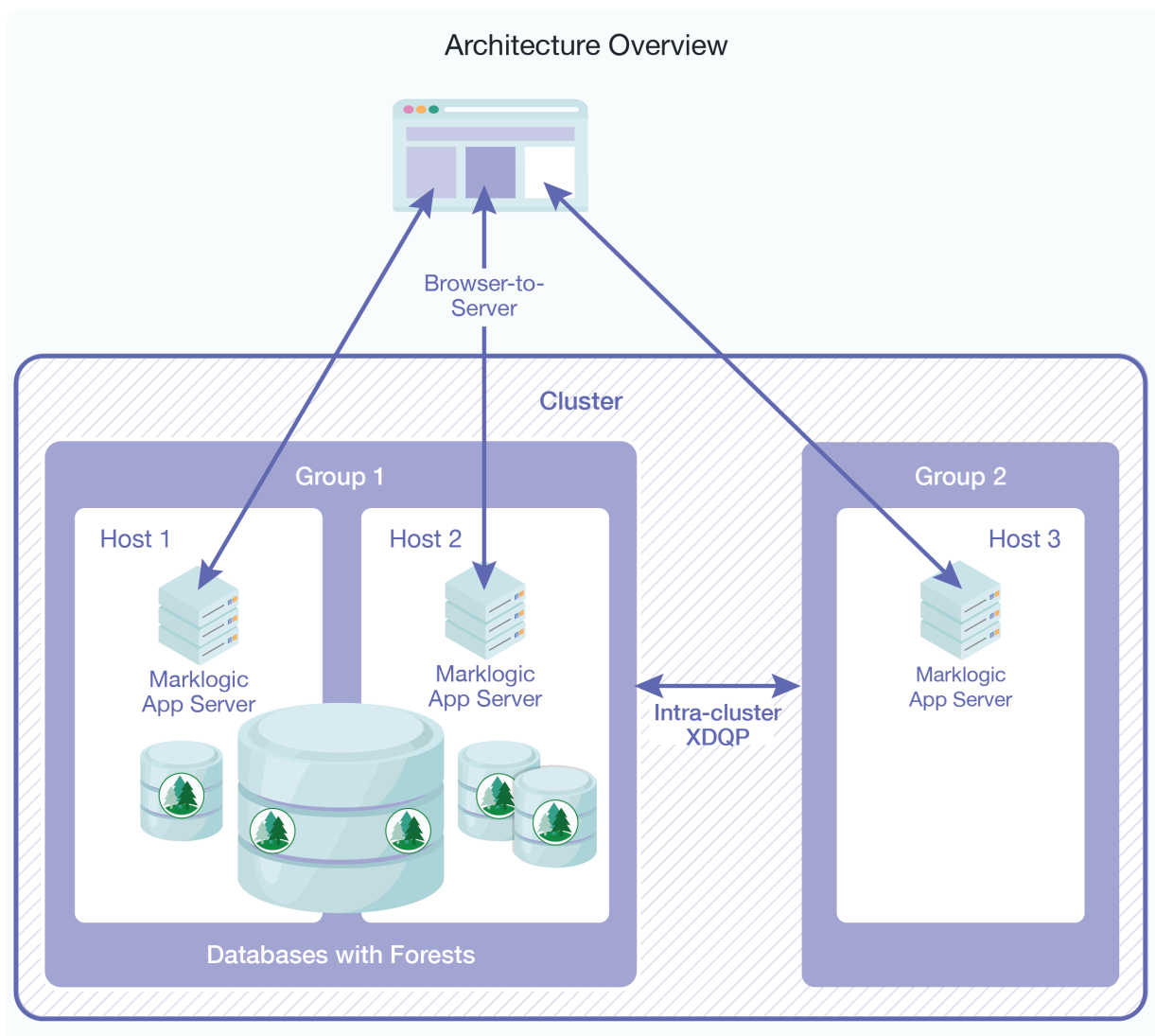
This guide explains administrative tasks for running MarkLogic on all platforms. For details on supported platforms, see the [Installation Guide](#) and [What's New in MarkLogic 11](#).

This document explains only the administrative tasks for the software. To learn how to install the software or get started using it, see these documents:

- [Installation Guide](#)
- [Getting Started with MarkLogic Server](#)

This document assumes that you have successfully completed all the tasks in [Getting Started with MarkLogic Server](#). If not, be sure to complete these basic tasks before doing any administrative work for MarkLogic Server. For a list of features in this release, a list of known incompatibilities with previous releases, and a list of all MarkLogic documentation, see [What's New in MarkLogic 11](#).

1.4. Architecture Overview



The figure shows a conceptual diagram of a simple MarkLogic Server deployment. Each host runs an instance of MarkLogic Server with its configured App Servers. One or more forests of a database may reside on a host. Hosts that do not have forests are functioning as e-nodes. One or more hosts can be in a group. One or more groups make up a cluster.

Applications communicate with MarkLogic over the network. Groups in a cluster communicate using XDQP. Clusters can communicate with other clusters using inter-cluster XDQP. Each of the three communication pathways can be configured to use TLS or SSL. The TLS and SSL protocols can be configured to use FIPS 140-2 approved cryptographic functions. FIPS mode is the default. For more details, see [Section 4.2, "OpenSSL FIPS 140-2 Mode" \[24\]](#).

For more information, see [Hosts](#) or one of the topics below:

- [HTTP Servers](#)
- [XDBC Servers](#)
- [WebDAV Servers](#)
- [ODBC Servers](#)
- [Groups](#)
- [Clusters](#)
- [Databases](#)

- [Forests](#)

2. Administrative (Admin) Interface

The MarkLogic Server Administrative Interface (or Admin Interface) is used to configure the MarkLogic Server software on your system.

2.1. Overview of the Admin Interface

With the Admin Interface, you can complete any of the following tasks:

- Managing basic software configuration
- Creating and configuring groups
- Creating and managing databases
- Creating and managing new forests
- Backing up and restoring forest content
- Creating and managing new web server and Java-language access paths
- Creating and managing security configurations
- Tuning system performance
- Configuring namespaces and schemas
- Checking the status of resources on your systems

The Admin Interface is implemented as a MarkLogic Server web application. By default, it runs on port 8001 of your hosts. If you have completed the basic tasks in [Getting Started with MarkLogic Server](#), then accessing the Admin Interface requires that you enter a username and password. After you have been authenticated, you should not need to re-enter your username and password to complete any of the other tasks outlined in this guide during the current session.

Some configuration changes require the server to restart to reflect the changes. Configuration changes that do not require the server to restart to reflect the changes are defined as “hot”. In a clustered deployment, “cold” tasks will require all of the hosts in the cluster to restart their instance of MarkLogic in order to reflect the changes. In a single-server deployment, “cold” tasks will cause MarkLogic to restart in order to reflect the changes. For a list of which tasks are “hot” and which are “cold,” see [Section 31, “Appendix A: ‘Hot’ versus ‘Cold’ Admin Tasks” \[268\]](#).

2.2. Accessing the Admin Interface

Only authorized administrators can log into the Admin Interface. An authorized administrator is a user who has the `admin` role or has the `admin-ui-user` role. Authorized administrators with the `admin` role have access to all administrative tasks in MarkLogic Server; therefore, authorized administrators are trusted personnel and are assumed to be non-hostile, appropriately trained, and following proper administrative procedures.

Users with the `admin-ui-user` role may view the Admin Interface but do not have access to data or the ability to make administrative changes. For more about the `admin-ui-user` role, see [Section 2.5, “The admin-ui-user Role” \[17\]](#).

To access the Admin Interface, follow these steps:

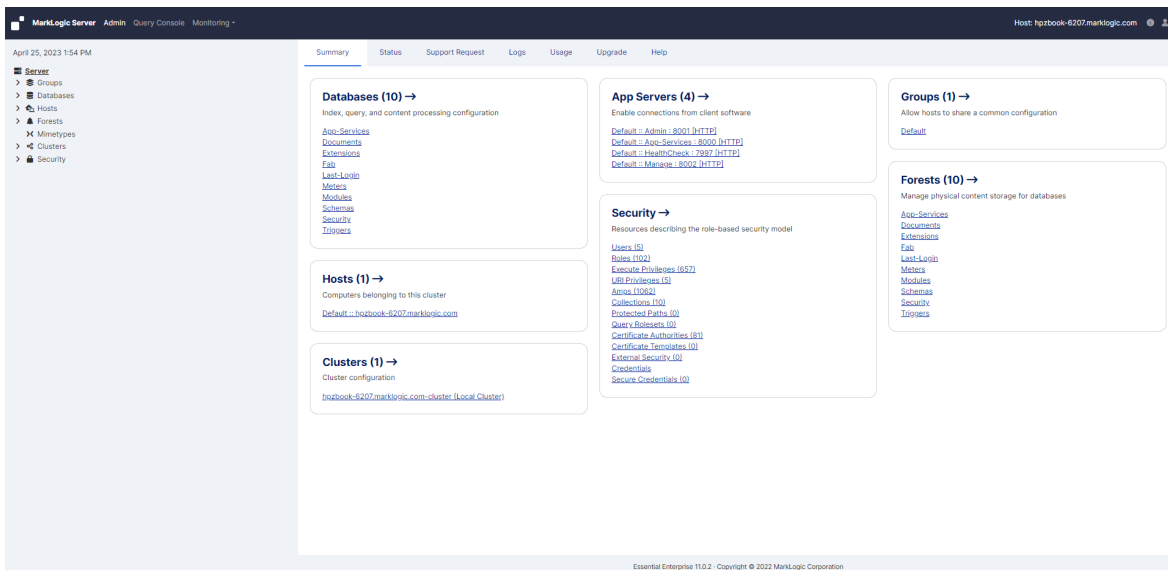
1. [Start MarkLogic Server](#).
2. Open <http://localhost:8001/>.




NOTE

If you are not accessing the Admin Interface from the same system on which MarkLogic Server is running, you will have to use the IP address or domain name of the server instead of localhost.

- Log in with your admin username and password. The summary screen for the Admin Interface appears:



 **NOTE** If you have already logged on as an admin user during this session, you do not have to log in again.

From the summary screen, you can see and click on many of the items configured in MarkLogic Server. The summary screen displays all of the databases, app servers, groups, forests, security objects, and hosts configured for your system. If you click on any object or category, the Admin Interface takes you to a more detailed page for the object or category.

2.3. Logging Off the Admin Interface

To log off the Admin Interface, close the browser window. This action is sufficient to end the current session and force the user to authenticate again.

2.4. Creating and Managing Administrators

MarkLogic Server administrators have the `admin` role. Users with the `admin` role, known as authorized administrators, are trusted personnel and are assumed to be non-hostile, appropriately trained, and following proper administrative procedures. For the procedures for creating, managing, and removing administrators, see [Section 23, "Security Administration" \[212\]](#).

2.5. The admin-ui-user Role

The `admin-ui-user` role has a set of base privileges that are required in order to use the Admin Interface. The role allows read-only access to the Admin Interface, but does not grant access to data, security configuration, or write access to the server configuration.

If an `admin-ui-user` attempts to perform an action for which they do not have the privilege, an error message appears. The error message contains the privilege or privileges needed to perform the action.

3. Common Administrative Procedures

This section describes some of the common administrative procedures for MarkLogic Server and where you can find more details on each procedure.

3.1. Installing and Upgrading MarkLogic Server


MarkLogic Server runs on a variety of platforms. For a list of supported platforms and installation procedures, see the *Installation Guide*.

For issues and procedures related to upgrading MarkLogic Server, see these sections:

- [Upgrading from Previous Releases](#) and [Upgrades and Database Compatibility](#) in the *Installation Guide*.
- [Upgrading a Cluster to a New Maintenance Release of MarkLogic Server](#) in the *Scalability, Availability, and Failover Guide*.

3.2. Starting MarkLogic Server

To start MarkLogic Server, use the appropriate system command for your platform:

Platform	Steps
Windows 10	<ol style="list-style-type: none"> 1. In the Windows taskbar, click in the search area. 2. Type <code>MarkLogic</code>. 3. Right-click on Start MarkLogic Server. 4. Select Run as administrator. 5. When prompted, allow the app to make changes to your device.
Windows	Select Start > Programs > MarkLogic Server > Start MarkLogic Server . <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;">  <p>NOTE When you start MarkLogic Server from the Start menu, the Windows service configuration for MarkLogic Server is set to start automatically. Also, to start the service, right-click the Start MarkLogic Server link in the Start menu and choose Run as Administrator, then choose to allow the action.</p> </div>
Red Hat Linux	As the <code>root</code> user, enter the following command: <pre>service MarkLogic start</pre>
Mac OS X	Select System Preferences > MarkLogic to open the MarkLogic control window. Click Start MarkLogic Server .

3.3. Stopping MarkLogic Server

This section contains the ways to stop MarkLogic Server.

3.3.1. From the Admin Interface

To stop the server from the Admin Interface, follow these steps:

1. Click the **Hosts** icon on the left tree menu.
2. Click on the name of the host you want to shut down.
3. Click the **Status** tab.
4. Click **Shutdown**.
5. A confirmation message displays while shutting down. Click **OK** to shut down the server.

**NOTE**

MarkLogic Server must be running in order for you to use the Admin Interface. Once you have stopped the server, you will no longer be able to access the Admin Interface until you start MarkLogic Server again; to restart the server, run the system command for your platform as described in [Section 3.2, “Starting MarkLogic Server” \[18\]](#).

3.3.2. With a System Command

You can stop MarkLogic Server with the appropriate system command for your platform:

Platform	Command
Microsoft Windows	Select Start > Programs > MarkLogic Server > Stop MarkLogic Server
	<div data-bbox="592 801 660 878" data-label="Image"></div> <div data-bbox="699 797 780 828" data-label="Section-Header">NOTE</div> <div data-bbox="699 828 1324 907" data-label="Text"> <p>If you are using Windows, to stop the service you must right-click the Stop MarkLogic Server link in the Start menu and choose Run as Administrator, then choose to allow the action.</p> </div>
Red Hat Linux	<code>service MarkLogic stop</code>
Mac OS X	<code>~/Library/StartupItems/MarkLogic stop</code>

3.3.3. With a Server-Side JavaScript

This Server-Side JavaScript stops MarkLogic Server:

```
'use strict';
xdmp.shutdown(null, "Shutting Down MarkLogic Server")
```

See `xdmp.shutdown()` for more details.

3.3.4. With an XQuery Script

This XQuery script stops MarkLogic Server:

```
xquery version "1.0-ml";
xdmp:shutdown((), "Shutting Down MarkLogic Server")
```

See `xdmp:shutdown()` for more details.

3.4. Restarting MarkLogic Server

This section contains the ways to restart MarkLogic Server.

3.4.1. From the Admin Interface

To restart the server from the Admin Interface, follow these steps:

1. Click the **Hosts** icon on the left tree menu.
2. Under **Hosts**, click on the name of the host you want to restart.
3. Click the **Status** tab.
4. Click **Restart**.
5. A confirmation message displays while restarting. Click **OK** to restart MarkLogic Server.

**NOTE**

The restart operation normally completes within a few seconds. It is possible, however, for it to take longer under some conditions (for example, if the Security database needs to run recovery or if the connectivity between hosts in a cluster is slow). If it takes longer than a few seconds for MarkLogic Server to restart, than the Admin Interface might return a 503: Service Unavailable message. If you encounter this situation, wait several seconds and then reload the Admin Interface.

3.4.2. With a Server-Side JavaScript

This Server-Side JavaScript restarts MarkLogic Server:

```
'use strict';
xdmp.restart(null, "Restarting MarkLogic Server")
```

See `xdmp.restart()` for more details.

3.4.3. With an XQuery Script

This XQuery script restarts MarkLogic Server:

```
xquery version "1.0-ml";
xdmp:restart((), "Restarting MarkLogic Server")
```

See `xdmp:restart()` for more details.

3.5. Creating and Configuring Forests and Databases

MarkLogic Server stores XML, JSON, XQuery, and JavaScript data in [forests](#). App Servers connect to a [database](#) that, in turn, accesses one or more forests.

Several types of [auxiliary databases](#) are created when you install MarkLogic Server, which are described in [Section 12.1, “Understanding Databases” \[81\]](#). This section outlines the general procedures for creating a database to store your documents.

To create a database to store your documents, follow these steps:

1. Create one or more forests, as described in [Section 22.2, “Creating a Forest” \[202\]](#). Depending on your storage, performance, and availability needs, you may want to create multiple forests, each on a separate host. See the for details.
2. Follow the procedure described in [Section 12.2, “Creating a New Database” \[88\]](#) to create your database. Until you understand all of the database settings, you need only provide a name for the database in the Database Name field. You can leave all of the other fields in the Database Specification in their default state.
3. Attach your forests to the database, as described in [Section 12.3, “Attaching and/or Detaching Forests to/from a Database” \[89\]](#).

3.6. Creating and Configuring App Servers

An application is executed on an App Server, which is configured with a specific database and port number. Once you have created a database, you can create an App Server. MarkLogic Server allows you to create three types of App Servers to support different types of applications:

- HTTP App Servers for executing XQuery or JavaScript, and servicing HTTP requests from a client, like a web server. For information on creating and configuring an HTTP App Server, see [Creating a New HTTP Server](#).

- XDBC App Servers for Contentbase Connector (XCC) applications that use the Java XCC libraries. For information on creating and configuring an XDBC App Server, see [XDBC Servers](#).
- WebDAV App Servers for accessing a MarkLogic Server database via a WebDAV client. For information on creating and configuring a WebDAV App Server, see [Section 8.6, “Procedures for Creating and Managing WebDAV Servers” \[57\]](#).
- ODBC App Servers for accessing a MarkLogic Server database via a SQL client. For information on creating and configuring an ODBC App Server, see [ODBC Servers](#).

To secure your App Server using SSL, see [Section 5.3.3, “Enabling SSL Communication over XDQP” \[40\]](#).

3.7. Setting Up Users, Roles, Privileges, and Permissions

MarkLogic Server provides a rich set of security objects that enable you to control user access to documents and applications, which are described in [Securing MarkLogic Server](#) and in [Section 23, “Security Administration” \[212\]](#) in this guide.

In addition to the Security pages in the Admin Interface, there are also XQuery, JavaScript, and REST functions you can use in scripts to set up and manage security objects.

3.8. Loading Content into a Database

Here are three ways to load content into a database:

- Use the load document functions. See [Loading Content into MarkLogic Server](#).
- Set up a WebDAV server and client such as Windows Explorer. See [Simple Drag-and-Drop Conversion](#) in the *Content Processing Framework Guide* for information on how to configure a WebDAV server to work with Windows Explorer.
- Use an XCC application. See [Using the Sample Applications](#) in *Developing with XCC*.

3.9. Running the XQuery Use Cases and Building Simple Applications

To test your MarkLogic Server configuration, Follow the procedure in [Exploring the Use Cases in Getting Started with MarkLogic Server](#). The procedure uses Query Console to evaluate the W3C XQuery use cases.

For procedures on building a simple XQuery application, see [Sample XQuery Application that Runs Directly Against an App Server](#) in *Getting Started with MarkLogic Server*. For more in-depth information, see the [Application Developer's Guide](#). If you are writing a Java application that communicates with MarkLogic Server through the XCC API, see [Developing with XCC](#).

3.10. Backing Up and Restoring Data

You can make backups of a database, as described in [Section 19.4, “Backing Up a Database” \[169\]](#), which backs up all of the forests in the database. You can also create backups of individual forests used by a database, as described in [Section 22.6, “Making Backups of a Forest” \[207\]](#).

There are a number of key differences between database-level and forest-level backups. A database-level backup, by default, backs up all of the forests in the database to the specified directory. Each time a database backup is initiated, a new set of backup data is created in that directory. With a forest-level backup, each forest must be backed up to a separate directory. In addition, each incremental backup of a forest is added onto the previous backup data. A forest backup also has additional logic that checks to see if any of its stands have changed before overwriting the backup of the earlier stand. Only the stands that have changed are overwritten.

Along with full backups, you can use incremental backups and journal archiving to create backups that enable you to recover your database to a specific point in time. For details, see [Section 19, “Backing Up and Restoring a Database” \[160\]](#).

You can restore an entire database from a database backup, as described in [Section 19.5.2, “Restoring a Database without Journal Archiving” \[173\]](#). You can restore an individual forest from either a database backup, as described in [Section 19.5.2, “Restoring a Database without Journal Archiving” \[173\]](#), or from an individual forest backup, as described in [Section 22.7, “Restoring a Forest” \[208\]](#).

3.11. Monitoring and Tuning Performance

For information on how to monitor the performance of MarkLogic Server, see [Monitoring MarkLogic Server Performance](#) in the *Query Performance and Tuning Guide*.

Factors that impact system performance include these:

- The configuration of MarkLogic Servers, as described in [Scalability Considerations in MarkLogic Server](#) in the *Scalability, Availability, and Failover Guide*.
- Merges, as described in [Section 15.1, “Overview of Merges: Merges are Good” \[108\]](#).
- Fragment size, as described in [Section 26, “Fragments” \[249\]](#).
- Index configuration, as described in [Section 24, “Text Indexing” \[227\]](#).
- Range indexes, as described in [Section 25, “Range Indexes and Lexicons” \[237\]](#).
- Reindexing your database, as described in [Section 12.7, “Reindexing a Database” \[90\]](#).
- Database memory and journal settings, as described in [Memory and Journal Settings \[85\]](#).
- **Database** field, configuration, as described in [Section 14, “Fields Database Settings” \[98\]](#).
- Log levels, as described in [Section 29.2, “Understanding the Log Levels” \[259\]](#).
- Trace Events set in the Diagnostics page on the left tree menu, under the group name.

For details on how to tune your applications for maximum performance, see the [Query Performance and Tuning Guide](#).

3.12. Scripting and Scheduling Administrative Tasks

MarkLogic Server includes built-in and library modules that enable you to write XQuery, JavaScript, and REST scripts that perform administrative tasks on MarkLogic Server. The functions provided by these modules enable you to script most administrative procedures.

For example, the Admin Library Module (`admin.xqy`) enables you to write scripts that create or modify databases, forests, App Servers, set up SSL security, and so on. The Security Library Module (`security.xqy`) provides a set of functions that enable you to create scripts that set up security entities. The `xdmp` built-in functions enable you to do forest and database backup/restore operations, as well as other database and forest management operations.

For a general overview of scripting administrative tasks, see [Scripting Administrative Tasks in MarkLogic Server](#) in the *Scripting Administrative Tasks Guide*. All of the available administrative functions are described in the [XQuery and XSLT Reference Guide](#) and in [MarkLogic REST API Reference](#).

You can schedule administrative scripts to be invoked at specific intervals or times, as described in [Section 30, “Scheduling Tasks” \[263\]](#).

3.13. Configuring Clusters, Groups, and Failover

A single instance of MarkLogic Server running on a single machine is called a **host**. You can configure multiple hosts into a **cluster** as described in the *Scalability, Availability, and Failover Guide*. Within a cluster, you can create **groups** of similarly configured hosts as described in [Section 5, “Groups” \[34\]](#). Different configurations of grouped hosts are useful when different groups of hosts perform different tasks or have different system capabilities.

If a host goes down, its duties can be resumed by another host in the cluster. MarkLogic provides support for failover, which allows the forest to automatically mount to a different host in the event of a forest's primary host going offline. For details on configuring forests for failover, see [High Availability](#)

[of Data Nodes with Failover](#) and [Configuring Shared-Disk Failover for a Forest](#) in the *Scalability, Availability, and Failover Guide*.

4. Clusters

This section describes cluster configuration. A *cluster* is a set of hosts that work together.

4.1. Overview of Cluster Configuration

In MarkLogic clusters, a common configuration is to have one group defined for the *evaluator* nodes (hosts that service query requests) and another group defined for the *data* nodes (hosts to which forests are attached).

The Cluster configuration page found in the Admin Interface enables you to configure FIPS 140-2 mode for a cluster and to couple local and foreign clusters. For a description of each configuration option, see the help tab of the group configuration page in the Admin Interface. For a discussion of how clustering works in MarkLogic Server, see [Clustering in MarkLogic Server](#) in the *Scalability, Availability, and Failover Guide*.

4.2. OpenSSL FIPS 140-2 Mode

MarkLogic Server uses FIPS-capable OpenSSL to implement the Secure Sockets Layer (SSL v3) and Transport Layer Security (TLS v1) protocols. When you install MarkLogic Server, FIPS mode is enabled by default and SSL RSA keys are generated using secure FIPS 140-2 cryptography. This implementation disallows weak ciphers and uses only FIPS 140-2 approved cryptographic functions. Should your applications experience any difficulty running in SSL FIPS-mode, you can disable FIPS-mode using the Admin Interface as described below.

For more information on the OpenSSL FIPS 140-2 cryptographic capabilities, refer to the documentation provided by the OpenSSL Project at <https://www.openssl.org/docs/fips.html>.

4.3. Procedures for Configuring Clusters

This section discusses how to configure clusters in MarkLogic Server.

4.3.1. Configuring OpenSSL FIPS 140-2 Mode

When FIPS 140-2 mode is enabled, the OpenSSL library is initialized into FIPS 140-2 mode at system startup. Note that this is the default behavior of MarkLogic Server. If FIPS mode is enabled or disabled on a running system, the OpenSSL library is reconfigured appropriately without requiring a server restart. When the FIPS mode setting changes and secure XDQP is configured, all XDQP connections are dropped and reestablished.

To configure a cluster to run in FIPS 140-2 mode, follow these steps:

1. Log into the Admin Interface.
2. Click the **Clusters** icon on the left tree menu.
3. Select the local cluster.
4. Click the **Configure** tab to open the **Local Cluster** page.
5. To enable SSL FIPS, select **true**.
6. Click **OK** to save the changes.

4.3.2. Configuring Simple Cluster Encryption

This section describes configuring a cluster's simple encryption-at-rest options for the internal Key Management System (KMS) that the MarkLogic Server installation process sets up for MarkLogic to use by default.



NOTE

- Adding or changing any encryption information will require you to restart all the hosts in the cluster.
- To set up a [more complex encryption-at-rest scenario](#) than what is described in this section--such as [using an external KMS](#)--see *Securing MarkLogic Server*.

To access the **Keystore** page, where you can configure the simple encryption-at-rest options described in the subsections, follow these steps:

1. Click **Clusters** in the left navigation tree.
2. Click the name of the cluster you want to configure.
3. Click the **Keystore** Tab. The **Keystore** page appears.



NOTE

Only the controls boxed in red are discussed in this section. Before using any other controls, see [Securing MarkLogic Server](#).

Encrypting Data, Configuration, and Log Files

You can use your Key Management Service (KMS) to encrypt your data, configuration, and log files at the cluster level. By default, all encryption is off.



NOTE

Adding or changing any encryption information will require you to restart all the hosts in the cluster.

To encrypt data, configuration, or log files, follow these steps:

1. [Access the Keystore page](#).
2. At the top of the page, choose the encryption options you want:

Field	Description
Data Encryption	<p>Specifies whether or not encryption is enabled for user data. Choose among 3 options:</p> <ul style="list-style-type: none"> • <code>force</code>: Causes all data in all databases in this cluster to be encrypted--even if a particular databases's Data Encryption setting is <code>off</code>. • <code>default-on</code>: Causes all data in all databases in this cluster to be encrypted--unless a particular database's Data Encryption setting is <code>off</code>. Then that database's data will not be encrypted. • <code>default-off</code>: Causes all data in all databases in this cluster not to be encrypted--unless a particular database's Data Encryption setting is <code>on</code>. Then that database's data will be encrypted. <p>See Encrypt a Database to turn on a database's Data Encryption setting and Turn off Encryption for a Database to turn it off.</p>
Config Encryption	<p>Specifies whether or not encryption is enabled for configuration files.</p>

Field	Description
Logs Encryption	Specifies whether or not encryption is enabled for log files.
Audit Log Encryption	[v11.1.0 and up] Specifies whether or not encryption is enabled for the audit log file even when Logs Encryption is disabled. (If Logs Encryption is enabled, the audit log file is encrypted regardless of this setting.)

3. Click **OK**. Your settings are saved, and the **Summary** tab for the local cluster appears.

**NOTE**

For more about MarkLogic encryption at rest and the internal KMS, see [Configuring Encryption at Rest](#) in *Securing MarkLogic Server*.

Setting the Internal KMS Backup Option

You can choose whether to include or exclude the internal KMS in backups. By default, it is included.

**NOTE**

Adding or changing any encryption information will require you to restart all the hosts in the cluster.

To change the backup option, follow these steps:

1. [Access the Keystore page](#).
2. On the **Internal KMS** tab, for the **Backup Option** field, choose either `include` or `exclude` from the dropdown.
3. Click **OK**. Your settings are saved, and the **Summary** tab for the local cluster appears.

**NOTE**

For more about MarkLogic encryption at rest and the internal KMS, see [Configuring Encryption at Rest](#) in *Securing MarkLogic Server*.

Changing the Internal KMS Password

You can change the password for your internal KMS. The initial password was set during the initial configuration of this cluster.

To change the internal KMS password, follow these steps:

1. [Access the Keystore page](#).
2. At the bottom of the **Internal KMS** tab, click **Change password**. The **Change Password** page appears.
3. Enter the current password into the first field.

4. Enter the new password into the second field.
5. Confirm the new password by entering it again into the third field.
6. Click **OK**. Your new password is set, and the **Summary** tab for the local cluster appears.

**NOTE**

For more about MarkLogic encryption at rest and the internal KMS, see [Configuring Encryption at Rest](#) in *Securing MarkLogic Server*.

Synchronizing the Internal KMS Keys

You can synchronize the internal KMS keys with the enveloped keys on MarkLogic Server. This is useful when you use encryption at rest.

**NOTE**

Adding or changing any encryption information will require you to restart all the hosts in the cluster.

To synchronize the internal KMS keys, follow these steps:

1. [Access the Keystore page](#).
2. At the bottom of the **Internal KMS** tab, click **Synchronize Keys**. The **Synchronize Keys** confirmation page appears.
3. Click **OK** to confirm that you want to synchronize the MarkLogic Server keys with your internal KMS. Your internal KMS keys are synchronized, and the **Summary** tab for the local cluster appears.

**NOTE**


For more about MarkLogic encryption at rest and the internal KMS, see [Configuring Encryption at Rest](#) in *Securing MarkLogic Server*.


4.3.3. Coupling Clusters

You can use the Admin Interface to couple local and foreign clusters to enable inter-cluster communication.

**NOTE**

The foreign cluster must be running the same version of MarkLogic as the local cluster.

 **NOTE**
The procedure described in this section must be repeated for every cluster.

 **NOTE**
If you are using [OAuth external security](#) with JSON Web Tokens (JWT) and you are setting up a foreign cluster with your Security database replicated to the foreign cluster, then you must sync your local and foreign internal keystores after creating OAuth external security objects on your local cluster. Sync them by using `xdmp:keystore-export()` to export the internal keystore from your local cluster then using `xdmp:keystore-import()` to import the internal keystore into your foreign cluster.

Before coupling clusters, you must specify a bootstrap host for each cluster. By default, the name of the cluster is that of the bootstrap host. You must edit the cluster name in the **Local Cluster Configuration** on each local cluster to be coupled with foreign clusters.


To couple clusters, follow these steps:

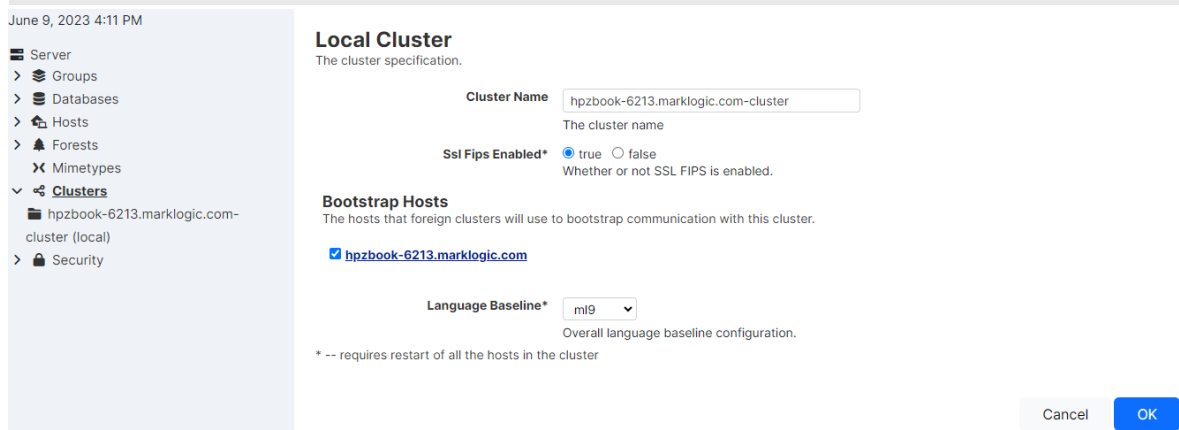
1. Click **Clusters** on the left tree menu.
2. Under **Clusters**, click the name of the cluster.
3. On the **Local Cluster** page, enter the cluster name.

Each cluster to be coupled must have one or more bootstrap hosts that stores the configuration information needed to establish an initial connection to foreign clusters. You must identify the bootstrap hosts in each cluster before attempting any of the configuration procedures described in this section.

The clusters in a production system will typically have more than one bootstrap host to ensure availability. When establishing an initial connection with a local cluster, a foreign cluster will connect to the first available bootstrap host.

4. Under **Bootstrap Hosts**, select one or more hosts to serve as the bootstrap hosts for this cluster.

 **NOTE**
It is best to choose the host that hosts your security forest as your bootstrap host. If you have configured your security forest for local disk failover, then also choose the host that hosts your replica security forest as a bootstrap host.



June 9, 2023 4:11 PM

- Server
 - > Groups
 - > Databases
 - > Hosts
 - > Forests
 - > Mimetypes
 - > **Clusters**
 - hpzbook-6213.marklogic.com-cluster (local)
 - > Security

Local Cluster
The cluster specification.

Cluster Name:
The cluster name

Ssl Fips Enabled* true false
Whether or not SSL FIPS is enabled.

Bootstrap Hosts
The hosts that foreign clusters will use to bootstrap communication with this cluster.

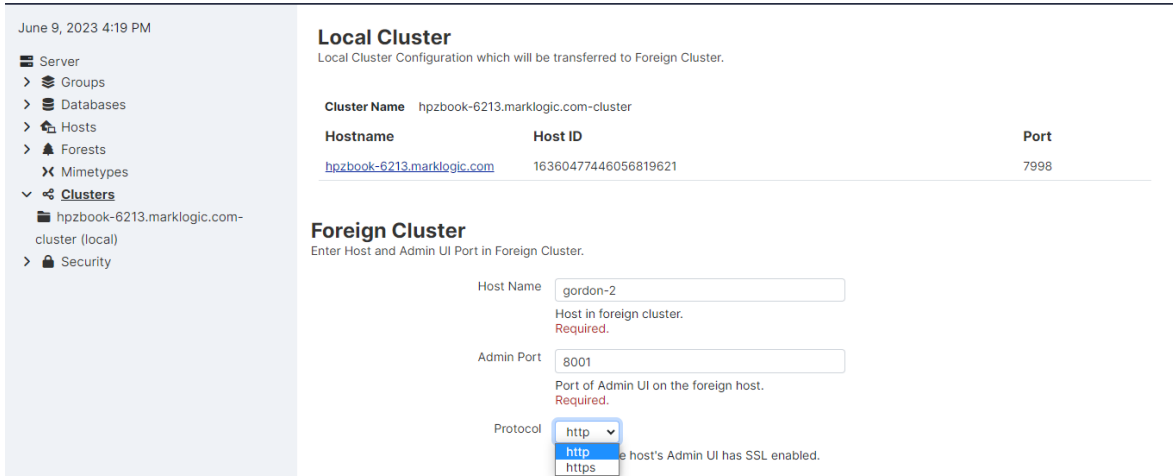
[hpzbook-6213.marklogic.com](#)

Language Baseline*
Overall language baseline configuration.

* -- requires restart of all the hosts in the cluster

Cancel

5. Click **OK** to save the configuration.
The remaining steps in this procedure describe how to “couple” a foreign cluster configuration to the bootstrap host on your local cluster. If you have designated more than one bootstrap host on your local cluster, pick any one of them.
6. Click the **Couple** tab.
7. In the **Foreign Cluster** portion of the page, enter the host name for any host in the foreign cluster to be coupled. You can also specify the admin port (if necessary) and the communication protocol to be used between the clusters (HTTP or HTTPS). When you have SSL enabled on the Admin App Server on the bootstrap host in the foreign cluster, set the **Protocol** field to `https`.



8. Click **OK**.
9. In the **Foreign Cluster Configuration** page, if you are using SSL for inter-cluster communication, configure the SSL security settings and timeout values.
10. In the **Verify Add Foreign Cluster** page confirm all of the settings are correct and click **OK**.
11. Click **ok** for any subsequent validate screens.
12. When validation is complete, the **Summary** window appears and displays the summary for the **Foreign Cluster** configuration. **Bootstrapped** indicates whether the foreign cluster configuration has been received by the local cluster. **Last Bootstrap** indicates the last time the foreign cluster configuration was received by the local cluster. Initially the status of **Bootstrapped** may appear as `false` and **Last Bootstrap** as `never`. Refresh your browser page to see the current status.

4.4. Running Behind a Load Balancer or Reverse Proxy

Starting with MarkLogic 11.1.0, the Admin UI, Query Console, Monitoring Dashboard, and History can be run behind a reverse proxy or load balancer and accessed via path-based routing. Additionally, the MarkLogic clients (Java, Node.js, XCC) and MarkLogic Content Pump (mlcp) can connect to a MarkLogic cluster through a load balancer or reverse proxy configured with path-based routing to MarkLogic app servers.

Accessing a MarkLogic cluster running behind a reverse proxy or load balancer does not require any configuration on the MarkLogic side. However, the MarkLogic UIs now support HTTP headers that can be used to specify the paths that are used to access each of them.

Header	Value
X-ML-ADM-Path	Path for accessing the Admin UI
X-ML-QC-Path	Path for accessing Query Console
X-ML-MNG-Path	Path for accessing the monitoring and management applications

When configuring a reverse proxy or load balancer, listener ports and paths are configured to map those ports and paths to target hosts and ports in the MarkLogic cluster. For example, a reverse proxy could be configured to map the following paths to ports:

Port	Path	Target Port
443	/ml_8000	8000
443	/ml_8001	8001
443	/ml_8002	8002
443	/my_app	8010

To access the MarkLogic UIs, the reverse proxy needs to be configured to add the following HTTP headers to the requests sent to MarkLogic:

```
X-ML-ADM-Path    "/ml_8001" ;
X-ML-QC-Path     "/ml_8000" ;
X-ML-MNG-Path    "/ml_8002" ;
```

Additionally, to prevent CSRF issues when accessing the MarkLogic UIs, the reverse proxy or load balancer needs to be configured to add the following HTTP headers to the requests sent to the MarkLogic UI ports (8000, 8001, 8002 by default):

Header	Value
Host	<proxy_server_host>:<proxy_server_port>
Referer	<proxy_server_protocol>://<proxy_server_host>:<proxy_server_port>
Origin	empty

Where `proxy_server_host` is the hostname of the reverse proxy or load balancer, `proxy_server_port` is the external port of the reverse proxy or load balancer (e.g. 80, 443, etc.) and `proxy_server_protocol` is the protocol that is used to access the reverse proxy or load balancer (e.g. http or https).

The clients libraries and tools support specification of an optional basepath connection option that would be used when accessing MarkLogic app servers running behind the reverse proxy or load balancer. See the documentation for each client library or tool for details.

Limitations

- HTTPS is supported with path-based routing. HTTP is not.
- Digest auth is not supported.
- Certificate-based authentication cannot be used when terminating TLS at the load balancer/proxy.

4.5. Configuring a MarkLogic Application Message and Banner

This topic describes how to configure your cluster to display a notification dialog and an application banner when users navigate to one of the built-in MarkLogic application pages, such as Query Console or the Monitoring Dashboard.

Administrators might want to use this feature in situations like these:

- To notify users of important system status changes, such as a planned outage.
- To make it easy for users to distinguish among MarkLogic clusters, such as testing versus production environments.

The notification dialog is only displayed to each user once per host from which he or she connects to a MarkLogic application. If the notification message changes, the dialog will be displayed again, next time the user navigates to one of the affected applications.

The UI configuration can be updated with the `admin:ui-set-banner` API call. More information about this API can be found at [admin:ui-set-banner](#).

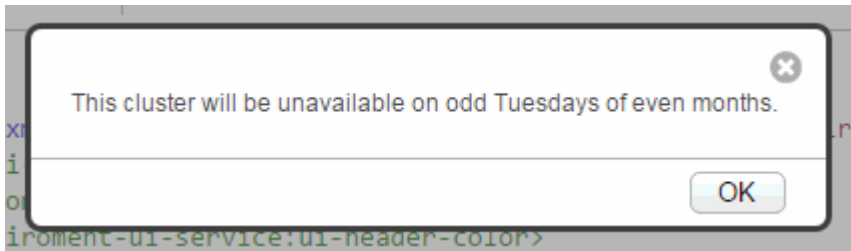
4.5.1. Example Configuration

This example is based on the following configuration. (Whitespace has been added to improve readability.) For more details on the structure and meaning of the elements, see [Section 4.5.2, “Configuration Reference” \[31\]](#).

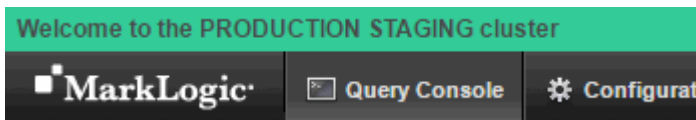
```
<env-ui:environment-ui xml:lang="zxx"
  xmlns:env-ui="http://marklogic.com/environment-ui">
  <env-ui:ui-active>true</env-ui:ui-active>
  <env-ui:ui-label>Welcome to the PRODUCTION STAGING cluster</env-ui:ui-label>
  <env-ui:ui-header-color>#33CC99</env-ui:ui-header-color>
  <env-ui:ui-header-text-color>#000000</env-ui:ui-header-text-color>
  <env-ui:ui-message>
    This cluster will be unavailable on odd Tuesdays of even months.
  </env-ui:ui-message>
</env-ui:environment-ui>
```

This configuration has the following effects on the UI of applications such as Query Console and the Monitoring Dashboard:

- The first time a user navigates to one of the built-in MarkLogic applications, MarkLogic displays the following dialog. The text comes from the `ui-message` configuration element.



- After the user dismisses the dialog, the configured banner is displayed at the top of the application page. The text comes from the `ui-label` configuration element, and the banner colors come from the `ui-header-color` and `ui-header-text-color` elements.



When no UI customization is active, no banner is displayed.

4.5.2. Configuration Reference

The `/cluster-ui-settings.xml` document in the App-Services database must have this structure. All elements are required:

```
<env-ui:environment-ui xml:lang="zxx"
  xmlns:env-ui="http://marklogic.com/environment-ui">
  <env-ui:ui-active>boolean</env-ui:ui-active>
  <env-ui:ui-label>banner_text</env-ui:ui-label>
  <env-ui:ui-header-color>color_code</env-ui:ui-header-color>
  <env-ui:ui-header-text-color>color_code</env-ui:ui-header-text-color>
  <env-ui:ui-message>notification_dialog_text</env-ui:ui-message>
</env-ui:environment-ui>
```

This table describes the child elements in more detail:

Element Local Name	Description
<code>ui-active</code>	Set to true for the configuration to take effect. Set to false to return to the default behavior (no notification dialog or banner).
<code>ui-label</code>	Text to be displayed in the banner.
<code>ui-header-color</code>	The background color of the banner.
<code>ui-header-text-color</code>	The color of the message text in the banner.

Element Local Name	Description
ui-message	The message to be displayed in the notification dialog box. The message is displayed to user only once (per host from which the user connects to the cluster), unless you update the configuration with a new message.

4.5.3. Example: Creating a New Configuration Document

Use this example to create an entirely new configuration document, rather than replacing just a portion of the existing configuration. For incremental changes, see the remaining examples.

Follow this procedure to create a new configuration using the template configuration that is installed with MarkLogic. Note that the template configuration is not active by default.

1. Read the template configuration from the App-Services database to get a baseline for your changes. This script reads the default:

```
xquery version "1.0-ml";
fn:doc('/cluster-ui-settings.xml')
```

2. Modify the configuration to meet your requirements.
3. Insert the new configuration into the App-Services database. This script is an example:

```
xquery version "1.0-ml";
let $new-config := (: YOUR CONFIG ELEM HERE :)
return xdmp:document-insert('/cluster-ui-settings.xml', $new-config)
```

4. Navigate to one of the built-in MarkLogic applications to observe your changes. For example, navigate to Query Console (<http://host:8000/qconsole>). If you already had one of the applications open in your browser, reload the page.

If you do not get a dialog or see the banner, there is likely an error in your configuration. MarkLogic validates your configuration against the schema in `INSTALL_DIR/Config/environment-ui.xsd`.

4.5.4. Example: Activating/Deactivating a Configuration

Use the following script to activate or deactivate a configuration. Run the script in Query Console against the App-Services database.

```
xquery version "1.0-ml";
declare namespace env-ui = "http://marklogic.com/environment-ui";
(: Set this var to false to deactivate, true to activate :)
let $state := fn:false()
let $env-ui-node :=
  fn:doc('/cluster-ui-settings.xml')/env-ui:environment-ui
return
  if (exists($env-ui-node)) then
    xdmp:node-replace(
      $env-ui-node/env-ui:ui-active,
      <env-ui:ui-active>{$state}</env-ui:ui-active>)
  else ()
(: Reload Query Console to see your changes :)
```

4.5.5. Example: Modifying the Notification Dialog Text

Use the following script to change the text displayed in the notification dialog box. Changing the text causes the dialog to be displayed to users the next time they navigate to one of the built-in MarkLogic applications.

Run this script in Query Console against the App-Services database.

```
xquery version "1.0-ml";
declare namespace env-ui = "http://marklogic.com/environment-ui";
(: Set this variable to your new notification :)
let $new-message := "This is your new message."
let $env-ui-node :=
  fn:doc('/cluster-ui-settings.xml')/env-ui:environment-ui
return
  if (exists($env-ui-node)) then
    xdm:node-replace(
      $env-ui-node/env-ui:ui-message,
      <env-ui:ui-message>{$new-message}</env-ui:ui-message>)
  else ()
(: Reload Query Console to see your changes. :)
```

When you reload Query Console, the notification dialog box should be displayed. It should contain your new message.

4.5.6. Example: Modifying the Banner Text

Use the following script to change the text in the banner that appears at the top of each built-in MarkLogic application page. Run the script in Query Console against the App-Services database.

```
(:This set the UI banner for the qconsole and admin GUI:)
xquery version "1.0-ml";
import module namespace admin = "http://marklogic.com/xdmp/admin" at "/MarkLogic/admin.xqy";

let $banner-options := map:map()
=> map:with("active", fn:true())
=> map:with("label", "Welcome to the PRODUCTION STAGING cluster")
=> map:with("headerColor", "#33CC99")
=> map:with("headerTextColor", "#000000")
=> map:with("message", "This cluster will be unavailable on odd Tuesdays of even months.")
return admin:ui-set-banner($banner-options)
```

5. Groups

This section describes how to use the Admin Interface to create and configure groups. For details on how to create and configure groups programmatically, see [Creating and Configuring Groups](#) in the *Scripting Administrative Tasks Guide*.

5.1. Overview of Groups

The basic definitions for group, host, and cluster are the following:

- A *group* is a set of similarly configured hosts within a cluster.
- A *host* is an instance of MarkLogic Server running on a single machine.
- A *cluster* is a set of hosts that work together.

For single-node configurations, you can only use one group at a time (because there is only one host). For clusters configurations with multiple hosts, you can have as many group configurations as makes sense in your environment.

Groups allow you to have several configurations, each of which applies to a distinct set of hosts. Different configurations are often needed when different hosts perform different tasks, or when the hosts have different system capabilities (disk space, memory, and so on). In cluster configurations, a common configuration is to have one group defined for the *evaluator* nodes (hosts that service query requests) and another group defined for the *data* nodes (hosts to which forests are attached).

HTTP, ODBC, XDBC, and WebDAV servers are defined at the group level and apply to all hosts within the group. Schemas and namespaces can also be defined at the group level to apply group-wide.

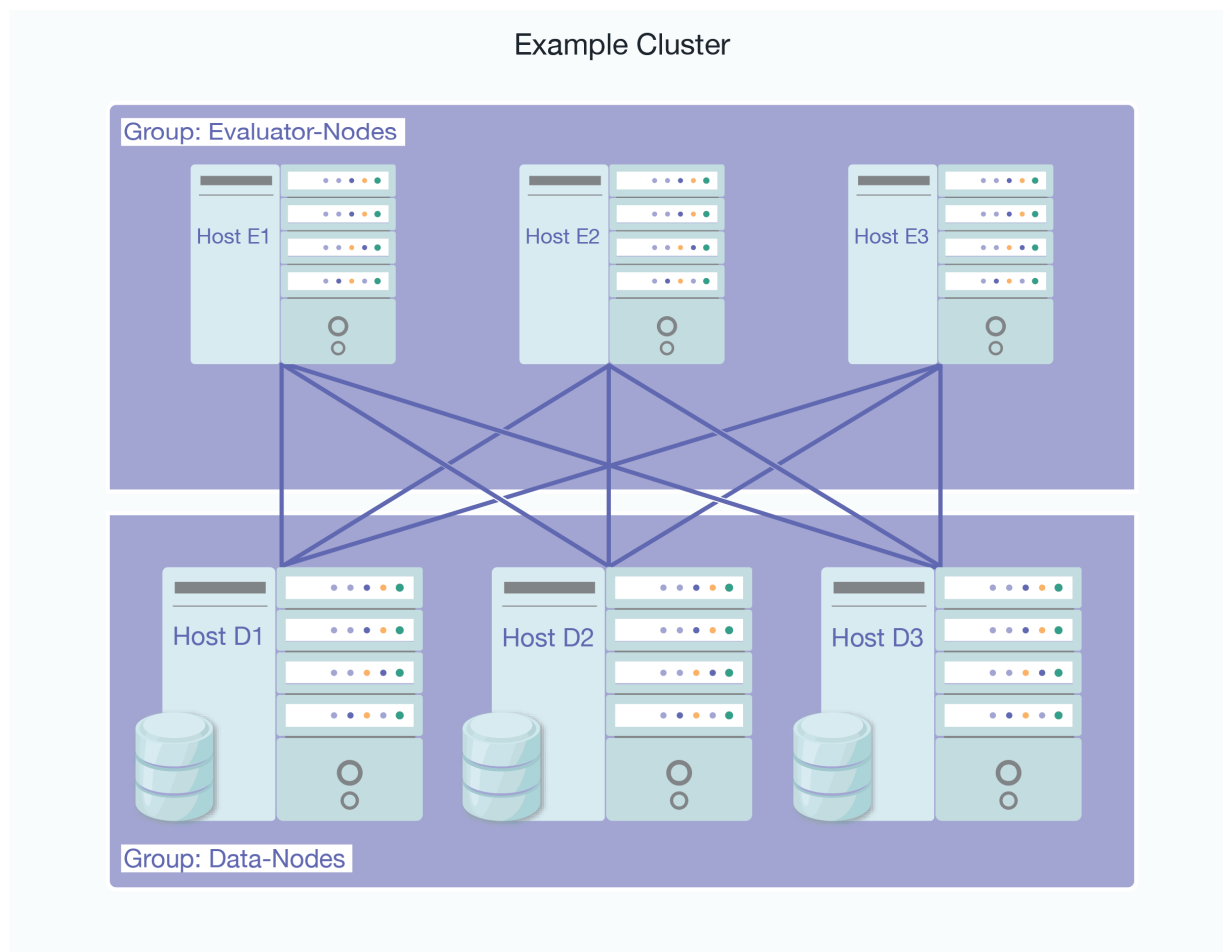
The Configure tab of the Group Administration section of the Admin Interface enables you to define configuration information for memory settings, SMTP server settings, and other configuration settings. The values for the settings are set at installation time based on your system memory configuration at the time of the installation. For a description of each configuration option, see the Help tab of the Group Administration section of the Admin Interface.

5.2. Example

The relationships between a cluster, a group, and a host in MarkLogic Server may be best illustrated with an example.

In this example, each machine is set up as a host within the example cluster. Specifically, hosts `E1`, `E2`, and `E3` belong to a group called `Evaluator-Nodes`. They are configured with HTTP servers and/or XDBC servers to run user applications. All hosts in the `Evaluator-Nodes` group have the same MarkLogic Server configuration.

Hosts `D1`, `D2`, and `D3` belong to a group called `Data-Nodes`. Hosts in the `Data-Nodes` group are configured with data forests and interact with the nodes in the `Evaluator-Nodes` group to service data requests. See the sections on databases, forests, and hosts for details on configuring data forests.



For more information about clusters, see the [Scalability, Availability, and Failover Guide](#).



NOTE

If you are administering a single-host MarkLogic environment, the host is automatically added to a Default group during the installation process. You will only have one host in the group and will not be able to add other hosts to the group.

5.3. Procedures for Configuring and Managing Groups

This section discusses the procedures for configuring and managing groups.

5.3.1. Creating a New Group

To create a new group, follow these steps:

1. Log into the Admin Interface.
2. Click the **Groups** icon on the left tree menu.
3. Click the **Create** tab. The **Group** page appears.
4. Go to the **Group Name** field and enter a shorthand name for the group. MarkLogic Server will use this name to refer to the group.
5. You can set the **cache sizing** method to enable you to manually set the settings for your caches, or have MarkLogic automatically set the cache settings:

- If you select **automatic**, MarkLogic automatically sizes the caches based on the available memory resources allocated at startup time.
- If you select **enode**, MarkLogic also automatically sizes the caches, but the sizes are tuned for better memory utilization of an Evaluation Node.
- If you select **dnode**, MarkLogic automatically sizes the caches as well, but the sizes are tuned for better memory utilization of a Data Node.

The `automatic`, `enode`, and `dnode` methods are necessary when running MarkLogic in a container, but are also applicable when running MarkLogic in other environments. When the Cache Sizing method is set to `automatic`, `enode`, or `dnode`, all manual cache settings, such as List Cache Size, Compressed Tree Cache Size and so on, can be set in the group configuration, but are not used until the Cache Sizing method is set to `manual`.

If you set the Cache Sizing method to `manual`, you can change cache size values, such as List Cache Size, Compressed Tree Cache Size, Expanded Tree Cache Size, and so on, or leave the defaults.



NOTE

Switching the Cache Sizing method from `manual` to `automatic` restarts MarkLogic Server. Switching from `automatic` to `manual` restarts MarkLogic Server if the current configuration does not match the saved configuration. Otherwise, MarkLogic Server does not restart.

6. Notes on other fields:

- **System Log Level** specifies the minimum log level messages sent to the operating system. Log levels are listed in decreasing level of log details. You may change the system log level or leave it at the default level.
- **File Log Level** specifies the minimum log level messages sent to the log file. Log levels are listed in decreasing level of log details. You may change the file log level or leave it at the default level.
- **Rotate Log Files** specifies how often to start a new log file. You may change this field or use the default value provided.
- **Keep Log Files** specifies how many log files are kept. You may change this field or use the default value provided.
- Set **Failover Enable** to true if you want to enable failover for the hosts in the group. To use failover, you must also enable failover for individual forests. If you set **failover enable** to false, failover is disabled for all the hosts in the group, regardless of their forest configurations.
- The **Xdqp Ssl Enabled** option and **Xdqp Ssl ciphers** field are to enable SSL for XDQP.

7. Click **OK**.



NOTE

For information about auditing, including how to configure various audit events, see [Section 10, “Auditing Events” \[71\]](#).

Adding a group is a “hot” administrative task; the changes are reflected immediately without a restart.




5.3.2. Group Settings






To access the settings for a particular group, follow these steps:


1. Log into the Admin Interface.
2. Click the **Groups** icon on the left tree menu.

3. Click the group for which you want to view settings.
4. Click the **Configure** tab.

This table contains the Group settings:

Field	Description
Group Name	The name of the group.
Cache Sizing	The cache sizing method. When the method is automatic, the cache size and cache partitions are computed automatically and the manual cache configuration settings are ignored. When the method is enode, the cache size and cache partitions are also computed automatically but they are tuned for better memory utilization of an Evaluation Node. The manual cache configuration settings are ignored when the method is enode. When the method is dnode, the cache size and cache partitions are also computed automatically but they are tuned for better memory utilization of a Data Node. The manual cache configuration settings are ignored when the method is dnode. When the method is manual, the manual cache configuration settings are used.
Compressed Tree Read Size	The size of the block for random access when reading compressed tree files.
Triple Cache Timeout	The time, in seconds, that a cached triple index page can be unused before being eligible to be flushed from the cache. Larger values can potentially cause more memory to be used for by the triple cache. Smaller values can potentially cause more time to be used reloading triple index pages.
Triple Value Cache Timeout	The time, in seconds, that a cached triple value index page can be unused before being eligible to be flushed from the cache. Larger values can potentially cause more memory to be used for by the triple value cache. Smaller values can potentially cause more time to be used reloading triple value index pages.
List Cache Size	The amount of memory to dedicate to caching termlist data for all on-disk stands. <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 10px; margin-top: 10px;">  NOTE This setting only appears when cache sizing is set to manual. </div>
List Cache Partitions	The number of independent list cache partitions to allocate. More partitions allow more concurrency, but make each individual cache partition smaller, which could make it more likely for the cache to fill up. The default is determined based on the amount of memory on your system and should work well for most installations. If you see a lot of CPU under-utilization under heavy concurrent query loads then raising this value can improve performance. The server may use fewer or more than the configured partitions to keep partition sizes between 2048 and 8192 megabytes. <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 10px; margin-top: 10px;">  NOTE This setting only appears when cache sizing is set to manual. </div>
Compressed Tree Cache Size	The amount of memory to dedicate to caching tree data in compressed form for all on-disk stands. <div style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 10px; margin-top: 10px;">  NOTE This setting only appears when cache sizing is set to manual. </div>

Field	Description
Compressed Tree Cache Partitions	<p>The number of independent compressed tree cache partitions to allocate. More partitions allow more concurrency, but make each individual cache partition smaller, which could make it more likely for the cache to fill up. The default is determined based on the amount of memory on your system and should work well for most installations. If you see a lot of CPU under-utilization under heavy concurrent query loads then raising this value can improve performance. The server may use fewer or more than the configured partitions to keep partition sizes between 512 and 8192 megabytes.</p> <div data-bbox="432 450 1385 584" style="background-color: #f0f0f0; padding: 10px;">  <p>NOTE This setting only appears when cache sizing is set to manual.</p> </div>
Expanded Tree Cache Size	<p>The amount of memory to dedicate to caching tree data in expanded form for the query evaluator.</p> <div data-bbox="432 689 1385 824" style="background-color: #f0f0f0; padding: 10px;">  <p>NOTE This setting only appears when cache sizing is set to manual.</p> </div>
Expanded Tree Cache Partitions	<p>The number of independent expanded tree cache partitions to allocate. More partitions allow more concurrency, but make each individual cache partition smaller, which could make it more likely for the cache to fill up. The default is determined based on the amount of memory on your system and should work well for most installations. If you see a lot of CPU under-utilization under heavy concurrent query loads then raising this value can improve performance. The server may use fewer or more than the configured partitions to keep partition sizes between 1024 and 8192 megabytes.</p> <div data-bbox="432 1055 1385 1189" style="background-color: #f0f0f0; padding: 10px;">  <p>NOTE This setting only appears when cache sizing is set to manual.</p> </div>
Triple Cache Size	<p>The amount of memory to dedicate to caching triple data for all on-disk stands.</p> <div data-bbox="432 1294 1385 1429" style="background-color: #f0f0f0; padding: 10px;">  <p>NOTE This setting only appears when cache sizing is set to manual.</p> </div>
Triple Cache Partitions	<p>The number of independent triple cache partitions to allocate. More partitions allow more concurrency, but make each individual cache partition smaller, which could make it more likely for the cache to fill up. The default is determined based on the amount of memory on your system and should work well for most installations. If you see a lot of CPU under-utilization under heavy concurrent query loads, then raising this value can improve performance. The server may use fewer or more than the configured partitions to keep partition sizes between 1024 and 8192 megabytes.</p> <div data-bbox="432 1659 1385 1794" style="background-color: #f0f0f0; padding: 10px;">  <p>NOTE This setting only appears when cache sizing is set to manual.</p> </div>

Field	Description
Triple Value Cache Size	<p>The amount of memory to dedicate to caching triple value data for all on-disk stands.</p> <div style="border: 1px solid gray; padding: 10px; margin-top: 10px;">  <p>NOTE This setting only appears when cache sizing is set to manual.</p> </div>
SmtP Relay	The network location (host:port) of the SMTP server. This server is used for all SMTP requests issued through the xdmp:email built-in function. The default port number of the SMTP server is 25. For details, see Section 5.3.4, “Configuring an SMTP Server” [43] .
SmtP Timeout	The time, in seconds, before an SMTP request times out and issues an error.
Http User Agent	The User-agent string issued when making HTTP requests from an App Server in the group.
Http Timeout	The time, in seconds, before an HTTP request times out.
Xdqp Timeout	The time, in seconds, before a request between a MarkLogic Server evaluator node (the node from which the query is issued) and a MarkLogic Server data node (the node from which the forest data is retrieved) times out.
Triple Value Cache Partitions	The number of independent triple value cache partitions to allocate. More partitions allow more concurrency, but make each individual cache partition smaller, which could make it more likely for the cache to fill up. The default is determined based on the amount of memory on your system and should work well for most installations. If you see a lot of CPU under-utilization under heavy concurrent query loads, then raising this value can improve performance. The server may use fewer or more than the configured partitions to keep partition sizes between 512 and 8192 megabytes.
Host Timeout	The time, in seconds, before a MarkLogic Server host-to-host request times out. The host-to-host requests are used for communication between nodes in a MarkLogic Server cluster.
Host Initial Timeout	The time, in seconds, that an instance of MarkLogic Server will wait for another node to come online when the cluster first starts up before deciding that the node is down, and initiating failover for any forests that are assigned to that offline host.
Retry Timeout	The time, in seconds, before a MarkLogic Server stops retrying a request.
Module Cache Timeout	The time, in seconds, that a cached module can be unused before being flushed from the cache. Larger values can potentially cause more memory to be used for cached modules. Smaller values can potentially cause more time to be used reloading uncached modules.
System Log Level	The minimum log level messages sent to the operating system. Log levels are listed in decreasing level of log details. You may change the system log level or leave it at the default level.
File Log Level	The minimum log level messages sent to the log file. Log levels are listed in decreasing level of log details. You may change the file log level or leave it at the default level.
Rotate Log Files	Specifies how often to start a new log file. You may change this field or use the default value provided.
Keep Log Files	Specifies how many log files are kept. You may change this field or use the default value provided.
Failover Enable	Set to true if you want to enable failover for the hosts in the group. To use failover, you must also enable failover for individual forests. If you set Failover Enable to false, failover is disabled for all the hosts in the group, regardless of their forest configurations.
Xdqp-Ssl-Enabled	Specifies whether SSL is enabled for XDQP. For details, see Section 5.3.3, “Enabling SSL Communication over XDQP” [40] .
Xdqp-Ssl-Allow-Sslv3	Specifies whether the SSL v3 protocol is allowed for XDQP.
Xdqp-Ssl-Allow-Tls	Specifies whether the Transport Layer Security protocol is allowed for XDQP.
Xdqp-Ssl Ciphers	The SSL ciphers that may be used.
Background I/O Limit	The maximum megabytes per second that a host may use for background I/O (merge, backup, restore). A value of 0 means no limit.
Metering Enabled	Specifies if usage metering is enabled for this group. When usage metering is enabled, a small amount of statistics about resources being used is saved to the meters database.
Performance Metering Enabled	Specifies if performance metering is enabled for this group. When enabled, performance statistics are stored in the Meters database to enable historic views of cluster performance.
Meters Database	The name of the database in which usage metering and historic performance data will be stored.
Performance Metering Period	The performance metering period in minutes.

Field	Description
Performance Metering Retain Raw	The number of days raw performance metering data is retained.
Performance Metering Retain Hourly	The number of days hourly performance metering data is retained.
Performance Metering Retain Daily	The number of days daily performance metering data is retained.
Telemetry-Log-Level	The minimum log level for log messages collected and sent by telemetry. For details, see Configure Telemetry in the Admin Interface in the <i>Monitoring MarkLogic Guide</i> .
Telemetry-Metering	The set of Metering data collected by telemetry. For details, see Configure Telemetry in the Admin Interface in the <i>Monitoring MarkLogic Guide</i> .
Telemetry-Config	The frequency of Config file changes collected by telemetry. For details, see Telemetry in the <i>Monitoring MarkLogic Guide</i> .
Telemetry Usage	Indicates whether the usage data is collected by telemetry.
Telemetry Proxy	The URL of the proxy used by telemetry. Proxy URL should start with <code>https://</code> , for example, <code>https://proxy.marklogic.com:8080</code> . If you don't specify the port number, it assumes the proxy server is listening on port 8080. For details, see Telemetry in the <i>Monitoring MarkLogic Guide</i> .
S3 Domain	The internet domain name of the simple storage service. The default value is <code>s3.amazonaws.com</code> . To access a different simple storage service that is API compatible with Amazon S3, specify it here.
S3 Protocol	The network protocol to use when accessing the simple storage service. The default is <code>https</code> . To use a more secure protocol when accessing the simple storage service, choose <code>https</code> .
S3 Server Side Encryption	The method of data encryption for data at rest on the simple storage service. The default is <code>aes256</code> . To encrypt data by custom AWS KMS key, choose <code>aws:kms</code> . You must use <code>https</code> to access an object protected by AWS KMS.
S3 Server Side Encryption Kms Key	The custom AWS KMS key of encryption for data at rest on the simple storage service. If you choose <code>kms:key</code> encryption and want to use your own KMS key, this field is required. Otherwise the default KMS key is used. The AWS KMS key must be in the same region as the S3 bucket.
S3 Proxy	The URL of the proxy server to access S3. The proxy URL should start with <code>https://</code> (for example, <code>https://proxy.marklogic.com:8080</code>). If you don't specify the port number, MarkLogic assumes the proxy server is listening on port 8080.
Azure Storage Proxy	The URL of the proxy server to access Azure Blob Storage. The proxy URL should start with <code>https://</code> (for example, <code>https://proxy.marklogic.com:8080</code>). If you don't specify the port number, MarkLogic assumes the proxy server is listening on port 8080.
Security Database	The security database where global security data are kept for hosts in this group. This database is where Amazon Web Services access keys and secret keys are kept for use with the simple storage service.
Temporary Directory	The directory used for disk-based sorting/joining, converters, and plugins.

5.3.3. Enabling SSL Communication over XDQP

This image shows the options related to configuring SSL for intra-cluster XDQP communication on the **Groups** screen:

- To enable encrypted SSL communication between hosts in the group, on the **Group** screen, set **Xdqp Ssl Enabled** to `true`. All communications to and from hosts in the group will be secured—even if the other end of the socket is in a group that does not have SSL enabled.
- The SSL keys and certificates used by the hosts are automatically generated when you install or upgrade MarkLogic Server. No outside authority signs the certificates used between hosts communicating over the internal XDQP connections in a cluster. Such certificates are self-signed and trusted by each host in the cluster. See [Keeping XDQP Certificates Up to Date](#) to keep these certificates up to date.



NOTE

If you are enabling this feature well after initially installing MarkLogic Server, use the first API described in [Keeping XDQP Certificates Up to Date](#) to make sure all certificates are valid.

- For details on configuring SSL communication between web browsers and App Servers, see [Configuring SSL on App Servers](#) in *Securing MarkLogic Server*. For details on configuring FIPS 140-2 mode for SSL communication, see [Section 4.2, “OpenSSL FIPS 140-2 Mode” \[24\]](#).

Keeping XDQP Certificates Up to Date

When a host is initialized, MarkLogic automatically generates a self-signed certificate for it in case you [enable XDQP SSL](#). These SSL certificates are good for 10 years.



WARNING

To keep an XDQP SSL-configured host running, you must renew its certificate before the old one expires.

At the host level, by default, MarkLogic detects when each host's certificate expires within 3 months and [logs a warning message](#) like this, alerting you to renew its certificate:

2023-07-26 15:39:58.791 Warning: XDQP host certificate will expire in 9 day(s). Please renew it using `admin:host-renew-xdqp-certificate` and `admin:host-activate-new-xdqp-certificate`.

At the cluster level, MarkLogic provides 3 APIs to keep these XDQP certificates up to date. The following table describes each API, and the outline following the table describes how you could use them to keep your certificates up to date:

API	Action
<code>admin.hostNeedRenewXdqpCertificate()</code>	<p>Obtains a cluster-wide list of hosts whose XDQP certificates expire within the specified time frame:</p> <ul style="list-style-type: none"> The default time frame is 3 months. The time frame that you specify is how often you must use this API to check for expiring certificates. That is, if you specify 12 months for your time frame, you must also use this API every 12 months to catch any expiring certificates so that you can update them. If the API returns an empty sequence, no certificates expire within your specified time frame. This action does not require all hosts in the cluster to be online.
<code>admin.hostRenewXdqpCertificate()</code>	<p>Generates a new XDQP certificate for any host within a cluster whose current certificate expires within the specified time frame:</p> <ul style="list-style-type: none"> The default time frame is 3 months. Use the same time frame with this API as you used with <code>admin.hostNeedRenewXdqpCertificate()</code>. Calling this API on one host in a cluster automatically generates new certificates for any other hosts in that cluster whose certificates also expire within the specified time frame. You must make sure that all hosts in the cluster are online: <ul style="list-style-type: none"> If a host is offline or goes offline, this API returns an error message like this one: <pre>[1.0-m1] XDMP-HOSTOFFLINE: xdmp:renew-host-certificate(ho:host-id("16773022918143520398"), "-----BEGIN CERTIFICATE-----&#10;MIICzjCCAbagAwIBAgIJAMgG7xp+keyY...) -- Host is offline or not responding</pre> If you see this error message, you must call the API again to complete this process.
<code>admin.hostActivateNewXdqpCertificate()</code>	<p>Activates new XDQP certificates generated with <code>admin.hostRenewXdqpCertificate()</code> for any hosts in the cluster:</p> <ul style="list-style-type: none"> You must generate new XDQP certificates before attempting to activate them. Calling this API on one host in a cluster automatically activates any other new certificates for hosts in that cluster. This action does not require all hosts in the cluster to be online.

Here is an outline of how to use these APIs periodically (that is, every 3 months or every year or every whatever time frame you specify in the APIs) to keep your certificates up to date:

- Use `admin.hostNeedRenewXdqpCertificate()` on one host in the cluster to find any hosts on that cluster whose certificates expire within your chosen time frame. If the API returns an empty sequence, skip the rest of these steps.
- During a maintenance window, update any expiring certificates:
 - Make sure that all hosts in the cluster are online.
 - Use `admin.hostRenewXdqpCertificate()` on one host in the cluster to generate new certificates for any hosts in that cluster whose certificates expire within your chosen time frame:
 - Use the same time frame here as you used in [Step 1](#).
 - Check for the error message that this API returns to indicate that a host is or has gone offline. If the message occurs, make sure that all hosts are online and call the API again.

- c. Use `admin.hostActivateNewXdqpCertificate()` on one host in the cluster to activate any new certificates for all hosts in that cluster.
3. Make sure that all expiring certificates have been updated by using `admin.hostNeedRenewXdqpCertificate()` again and checking that it returns an empty sequence.

5.3.4. Configuring an SMTP Server

The installation process configures an SMTP server based on the environment at installation time. A single SMTP server is configured for all of the hosts in a group. The SMTP configuration is used when applications use `xdmp.email()` (JavaScript) or `xdmp:email()` (XQuery).

To change the SMTP server or the SMTP timeout for the system (the time after which SMTP requests fail with an error), follow these steps:

1. Log into the Admin Interface.
2. Click the **Groups** icon on the left tree menu.
3. Click the name of the group you want to configure SMTP for either from the menu tree or from the **Summary** tab.
4. In the **SmtP Relay** field, enter the hostname for your SMTP server.
5. In the **SmtP Timeout** field, enter the time (in seconds) after which requests will time out.
6. Click **OK**.

Changing any SMTP settings is a hot operation; the server does not need to restart to reflect your changes.

5.3.5. Restarting All Hosts in a Group

To restart all the hosts in a group, follow these steps:

1. Log into the Admin Interface.
2. Click the **Groups** icon in the left tree menu.
3. Click the name of the group containing the host you want to restart either from the menu tree or from the **Summary** tab.
4. Click the **Status** tab.
5. Click **Restart**.
6. A confirmation message displays. Click **OK** to restart all of the hosts in the MarkLogic Server group.



NOTE

The restart operation normally completes within a few seconds. It is possible, however, for it to take longer under some conditions (for example, if the Security database needs to run recovery or if the connectivity between hosts in a cluster is slow). If it takes longer than a few seconds for MarkLogic Server to restart, than the Admin Interface might return a `503: Service Unavailable` message. If you encounter this situation, wait several seconds and then reload the Admin Interface.

5.3.6. Deleting a Group

You must drop all hosts assigned to a group before you can delete a group.

To delete a group, follow these steps:

1. Log into the Admin Interface.
2. Click the **Hosts** icon on the left tree menu.
3. Check that there is not a host assigned to the group you wish to delete. All hosts assigned to a group must be dropped before the group can be deleted. Dropping a host from a group does not drop the host from the cluster.
4. Click the **Groups** icon on the left tree menu.
5. Click the name of the group you want to delete, either from the menu tree or from the **Summary** tab.
6. Click **Delete**.
7. A confirmation message displays. Click **OK** to permanently delete the group.

Deleting a group is a hot operation; the server does not need to restart to reflect your changes.

5.4. App Server Status Page

The App Server Status page lists information about the activity of an app server. The page contains this information:

- Host - The name of the host. Click the link for more detailed information.
- Threads - The number of threads currently in use by the application server for the host.
- Requests - The number of requests being processed by this host through the application server.
- Updates - The number of update queries being processed by this host through the application server.
- Queued - The number of requests waiting to be processed in the application server queue.
- Average Time - The average query response times for queries being processed.
- Request Rate - The number of queries processed per second.
- Oldest Request - The elapsed time of the longest-running query.
- Hits - The number of times queries could use the expanded tree cache.
- Misses - The number of times queries could not use the expanded tree cache.
- Ratio - The ratio of expanded tree cache hits to total times the cache was accessed.

5.5. Access the App Server Status Page

To access the App Server Status page:

1. Log into the Admin Interface.
2. Click **Groups** on the left tree menu.
3. Click the appropriate group (for example, **Default**).
4. Click **App Servers** on the left tree menu.
5. Click the appropriate app server.
6. Click the **Status** tab.

6. HTTP Servers

This section describes how to use the Admin Interface to create and configure HTTP servers.



NOTE

To create and configure HTTP servers programmatically, see [Creating and Configuring App Servers](#) in the *Scripting Administrative Tasks Guide*.

MarkLogic Server enables you to write web applications by connecting sets of XML or JSON content to HTTP servers that can access server-side XQuery, JavaScript, and REST programs. These applications can return XHTML, XML, or JSON content to a browser or other HTTP-enabled client application.

HTTP servers are defined at the group level and are accessible by all hosts within the group. Each HTTP server provides access to a set of XQuery programs that reside within a specified directory structure. Each host in the group must have access to the directory structure or mirror the directory structure along with the program files. An HTTP server executes the server-side programs against the database to which it is connected.

HTTP servers follow the MarkLogic Server security model, as do WebDAV, ODBC, and XDBC servers. The server authenticates access to those programs using user IDs and passwords stored in the security database for that HTTP server. (Each HTTP server is connected to a database, and each database is in turn connected to a security database in which security objects such as users are stored.)

HTTP servers execute code, either from a specified location on the file system or from a Modules database.

Granular access control to the system and to the data is achieved through the use of privileges and permissions. For details on configuring security objects in MarkLogic Server, see [Section 23, “Security Administration” \[212\]](#). For conceptual information on the MarkLogic Server security model, see [Securing MarkLogic Server](#).

6.1. Creating a New HTTP Server

To create a new server, follow these steps:

1. Log into the Admin Interface.
2. Click the **Groups** icon.
3. Click the group in which you want to define the HTTP server (for example, **Default**).
4. Under the group name, click **App Servers** on the left tree menu.
5. Click the **Create HTTP** tab. The **Http Server** page appears.
6. In the **Server Name** field, enter a shorthand name for this HTTP server. MarkLogic Server will use this name to refer to this server on display screens in the Admin Interface.
7. In the **Root** field, enter the name of the directory in which you will store your programs. If the **Modules** field is set to a database, then the root must be a directory URI in the specified modules database.

If the **Modules** field is set to **(file system)**, then the root directory is either a fully-qualified pathname or is relative to the directory in which MarkLogic Server is installed. The following table shows the default installation directory for each platform:

Platform	Program Directory
Microsoft Windows	C:\Program Files\MarkLogic
Red Hat Linux	/opt/MarkLogic
Mac OS X	~/Library/MarkLogic

**NOTE**

Unless you specify a shared drive, all hosts in the group will need to have a copy of the programs in the directory specified above.

**WARNING**

Do not create HTTP server root directories named Docs, Data, or Admin. These directories are reserved by MarkLogic Server for other purposes. Creating HTTP server root directories with these names can result in unpredictable behavior of the server and may also complicate the software upgrade process.

8. In the **Port** field, enter the port number through which you want to make this HTTP server available. The port number must not be assigned to any other HTTP, ODBC, XDBC, or WebDAV server.
9. In the **Modules** field, select the database to use as the modules database for your documents, or leave it at the default of storing your modules on the file system. For information on what a modules database is, see [Section 12.1.2, “Modules Database” \[81\]](#).
10. In the **Database** field, select the database to be accessed by this HTTP server. Multiple HTTP, ODBC, XDBC, and WebDAV servers can access the same database.
11. Scroll down to the **Authentication** field. Select an authentication scheme, as described in [Types of Authentication](#) in *Securing MarkLogic Server*. The default is **digest**, which uses encrypted passwords.

If you select application-level authentication, you will also need to fill in a default user. Any one accessing the HTTP server is automatically logged in as the default user until the user logs in explicitly.

**WARNING**

If you use an admin user (admin) as the default user (an authorized administrator with the admin role), then everyone who uses this App Server is automatically a user with the admin role, which effectively turns off security for this App Server.

12. Scroll to the **Privilege** field near the bottom of the screen. This field represents the privilege needed to access (login to) the server.
13. Set any other properties for this App Server, as appropriate to your needs:
 - **Last Login** and **Display Last Login** are described in [Section 11.3, “Storing and Monitoring the Last User Login Attempt” \[79\]](#).
 - **Backlog** specifies the maximum number of pending connections allowed on the HTTP server socket.
 - **Threads** specifies the maximum number of App Server threads allocated to this port by each server in the cluster.
 - **Request Timeout** specifies the maximum number of seconds before a socket receives a timeout for the first request.
 - **Keep Alive Timeout** specifies the maximum number of seconds before a socket receives a timeout for subsequent requests over the same connection.
 - **Session Timeout** specifies the maximum number of seconds before an inactive session times out.

- **Max Time Limit** specifies the upper bound for any request's time limit. No request may set its time limit (for example with `xdmp:set-request-time-limit()`) higher than this number. The time limit, in turn, is the maximum number of seconds allowed for servicing a query request. The App Server gives up on queries which take longer, and returns an error.
 - **Default Time Limit** specifies the default value for any request's time limit, when otherwise unspecified. A request can change its time limit using `xdmp:set-request-time-limit()`. The time limit, in turn, is the maximum number of seconds allowed for servicing a query request. The App Server gives up on queries which take longer, and returns an error.
 - **Static Expires** adds an "expires" HTTP header for static content to expire after this many seconds.
 - **Pre-Commit Trigger Limit** specifies the maximum number of pre-commit triggers a single statement against this App Server can invoke. For more information on triggers, see [Using Triggers to Spawn Actions](#) in the *Application Developer's Guide*.
 - **Pre-Commit Trigger Depth** specifies the maximum depth (how many triggers can cause other triggers to fire, which in turn cause others to fire, and so on) for pre-commit triggers that are executed against this App Server. For more information on triggers, see [Using Triggers to Spawn Actions](#) in the *Application Developer's Guide*.
 - **Collation** specifies the default collation for queries run in this appserver. This will be the collation used for string comparison and sorting if none is specified in the query. For details, see [Encodings and Collations](#) in the *Search Developer's Guide*.
 - **Concurrent Request Limit** specifies the maximum number of requests any user may have running at a specific time. 0 indicates no maximum. For details, see [Section 11.1, "Managing Concurrent User Requests"](#) [78].
 - **Log Errors** specifies whether to log uncaught errors for this App Server to the `ErrorLog.txt` file. This is useful to log exceptions that might occur on an App Server for later debugging.
 - **Debug Allow** specifies whether to allow requests against this App Server to be stopped for debugging, using the MarkLogic Server debugging APIs.
 - **Default Xquery Version** specifies the default XQuery language for this App Server if an XQuery module does explicitly declare its language version.
 - **Multi Version Concurrency Control** specifies how strict queries behave about getting the latest timestamp. This only affects query statements, not update statements. For details about queries and transactions in MarkLogic Server, see [Understanding Transactions in MarkLogic Server](#) in the *Application Developer's Guide*.
 - The **Error Handler** and **Url Rewriter** fields are described in [Controlling App Server Access, Output, and Errors](#) in the *Application Developer's Guide*.
 - The properties associated with SSL support are described in [Configuring SSL on App Servers in Securing MarkLogic Server](#).
14. Scroll to the top or bottom and click **OK**.

The HTTP server is now created. Creating an HTTP server is a "hot" admin task; the changes take effect immediately. For information and setup instructions for managing user sessions and/or keeping track of login attempts, see [Section 11, "Managing User Requests and Monitoring Login Attempts"](#) [78].

6.2. Setting Output Options for an HTTP Server

For each HTTP Server, you can set various default output options. These output options affect how data returned from the App Server is serialized. You can also set these options at the query level to override any default options. You can set serialization options to override the App Server defaults in XQuery with the `declare option XQuery` prolog, and in XSLT using the `<xsl:output>` instruction. For details on setting the serialization options in XQuery, see [Declaring Options in the XQuery and XSLT Reference Guide](#). For XSLT output details, see the XSLT specification <http://www.w3.org/TR/xslt20#serialization>.

To specify defaults for the App Server, follow these steps:

1. Log into the Admin Interface.
2. Click the **Groups** icon in the left tree menu.
3. Click the group which contains the HTTP server you want to view (for example, **Default**).
4. Click **App Servers**. to expand the heading.
5. Underneath **App Servers**, click the App Server to expand the heading
6. Underneath the App Server heading, click **Output Options**. The **Output Options** page appears:

April 27, 2023 12:16 PM

Configure Help

Cancel OK

Output Options
Serialization parameters.

Output Sgml Character Entities Output SGML character entities.

Output Encoding The default output encoding.

Output Method Output method.

Output Byte Order Mark The output sequence of octets is to be preceded by a Byte Order Mark.

Output Cdata Section Namespace Uri Namespace URI of the "cdata section localname" specified below.

Output Cdata Section Localname Element localname or list of element localnames to be output as CDATA sections.

Output Doctype Public A public identifier to use on the emitted DOCTYPE.

Output Doctype System

7. Set any options that you want to control for this App Server.
8. Click **OK** to save your changes.

For more details about App Server output, see [Controlling App Server Access, Output, and Errors](#) in the *Application Developer's Guide*.

6.3. Viewing HTTP Server Settings

To view the settings for a particular HTTP server, follow these steps:

1. Log into the Admin Interface
2. Click the **Groups** icon on the left tree menu.
3. Expand the **App Servers** heading by clicking on it.
4. Locate the HTTP server for which you want to view settings, either in the tree menu or on the summary page.
5. Click the link for the HTTP server.
6. View the settings.

6.4. Deleting an HTTP Server

To delete the settings for an HTTP server, follow these steps:

1. Log into the Admin Interface
2. Click the **Groups** icon in the left tree menu.
3. Click the group which contains the HTTP server you want to delete (for example, **Default**).
4. Underneath the group name, click **App Servers**.
5. Locate the HTTP server you want to delete, either in the tree menu or on the summary page.
6. Click the name of the HTTP server.
7. Click **Delete**.
8. A confirmation message displays. Confirm the delete and click **OK**.

Deleting an HTTP server is a “cold” admin task; the server restarts to reflect your changes.

6.5. Canceling a Request

To cancel a long-running request (for example, a long-running query statement or update statement), follow these steps:

1. Click **Groups** in the left tree menu.
2. Click the group which contains the server that has the request to cancel (for example, **Default**).
3. Navigate to the App Server in which the request was issued, either from the tree menu or from the summary page.
4. Click the **Status** tab.
5. At the bottom right of the **Group Status** page, click the **cancel** link on the row for the query you want to cancel.
6. Click **OK** on the **Cancel Request confirmation** page. If the request is already completed when the confirmation page occurs, the page will indicate that the request cannot be found.

The request is canceled and the App Server Status page appears again.

7. XDBC Servers

This section describes how to use the Admin Interface to create and configure XDBC servers.



NOTE

To create and configure XDBC servers programmatically, see [Creating and Configuring App Servers](#) in the *Scripting Administrative Tasks Guide*.

XDBC (XML Database Connector) servers are defined at the group level and are accessible by all hosts within the group. Each XDBC server provides access to a specific forest, and to a library (root) of XQuery programs that reside within a specified directory structure. Applications execute by default against the database that is connected to the XDBC server.

XDBC Servers allow XML Contentbase Connector (XCC) applications to communicate with MarkLogic Server. XCC is an API used to communicate with MarkLogic Server from Java middleware applications. XDBC servers also allow old-style XDBC applications to communicate with MarkLogic Server, although XDBC applications cannot use certain 3.1 and newer features (such as point-in-time queries). Both XCC and XDBC applications use the same wire protocol.

XQuery requests submitted via XCC return results as specified by the XQuery code. These results can include XML and a variety of other data types. It is the XCC application's responsibility to parse, process and interpret these results in a manner appropriate to the variety of data types available. There are a number of publicly available libraries for assisting with this task, or you may write your own code. In order to accept connections from XCC-enabled applications, MarkLogic Server must be configured with an XDBC Server listening on the designated port. Each XDBC Server connects by default to a specific database within MarkLogic Server, but XCC provides the ability to communicate with any database in the MarkLogic Server cluster to which your application connects (and for which you have the necessary permissions and privileges).

XDBC servers follow the MarkLogic Server security model, as do HTTP and WebDAV servers. The server authenticates access to those programs using user IDs and passwords stored in the security database for that XDBC server. (Each XDBC server is connected to a database, and each database is in turn connected to a security database in which security objects such as users are stored.)

Granular access control to the system and to the data is achieved through the use of privileges and permissions. For details on configuring security objects in MarkLogic Server, see [Section 23, "Security Administration" \[212\]](#). For conceptual information on the MarkLogic Server security model, see [Securing MarkLogic Server](#).

7.1. Creating a New XDBC Server

To create a new server, follow these steps:

1. Log into the Admin Interface.
2. Click the group in which you want to define the XDBC server (for example, **Default**).
3. Click **App Servers** on the left tree menu.
4. Click the **Create XDBC** tab. The **Xdbc Server** page appears:
5. In the **XDBC Server Name** field, enter a shorthand name for this XDBC server. MarkLogic Server will use this name to refer to this server on display screens in the Admin Interface.

6. In the **Root** field, enter the name of the directory in which you will store your XQuery programs. If the **modules** field is set to a database, then the root must be a directory URI in the specified modules database.

If the **Modules** field is set to (**file system**), then the root directory is either a fully-qualified pathname or is relative to the directory in which MarkLogic Server is installed. The following table shows the default installation directory for each platform:

Platform	Program Directory
Microsoft Windows	C:\Program Files\MarkLogic
Red Hat Linux	/opt/MarkLogic
Mac OS X	~/Library/MarkLogic



NOTE

Unless you specify a shared drive, all hosts in the group will need to have a copy of the XQuery programs in the directory specified above.



WARNING

Do not create XDBC server root directories named Docs, Data, or Admin. These directories are reserved by MarkLogic Server for other purposes. Creating XDBC server root directories with these names can result in unpredictable behavior of the server and may also complicate the software upgrade process.

7. In the **Port** field, enter the port number through which you want to make this XDBC server available.
The port number must not be assigned to any other XDBC, HTTP, or WebDAV server.
8. In the **Modules** field, select the database to use as the modules database for your XQuery documents, or leave it at the default of storing your XQuery modules on the file system. For information on what a modules database is, see [Section 12.1.2, “Modules Database” \[81\]](#).
9. In the **Database** field, select the database to be accessed by this XDBC server. Multiple HTTP, XDBC, and WebDAV servers can access the same database.
10. Scroll to the **Authentication** field and select an authentication scheme, as described in [Types of Authentication](#) in *Securing MarkLogic Server*. The default is digest, which uses encrypted passwords.
11. Scroll to the **Privilege** field near the bottom of the screen. This field represents the privilege needed to access (login to) the server. You may leave this field blank.
A user accessing the XDBC server must have the execute privilege selected in order to access the XDBC server (or be a member of the `admin` role).
12. Set any other properties for this App Server, as appropriate to your needs:
- **Last Login** and **Display Last Login** are described in [Section 11.3, “Storing and Monitoring the Last User Login Attempt” \[79\]](#).
 - **Backlog** specifies the maximum number of pending connections allowed on the HTTP server socket.
 - **Threads** specifies the maximum number of App Server threads.
 - **Request Timeout** specifies the maximum number of seconds before a socket receives a timeout for the first request.
 - **Keep Alive Timeout** specifies the maximum number of seconds before a socket receives a timeout for subsequent requests over the same connection.
 - **Session Timeout** specifies the maximum number of seconds before an inactive session times out.
 - **Max Time Limit** specifies the upper bound for any request's time limit. No request may set its time limit (for example with `xdmp:set-request-time-limit()`) higher than this number. The

- time limit, in turn, is the maximum number of seconds allowed for servicing a query request. The App Server gives up on queries which take longer, and returns an error.
- **Default Time Limit** specifies the default value for any request's time limit, when otherwise unspecified. A request can change its time limit using `xdmp:set-request-time-limit()`. The time limit, in turn, is the maximum number of seconds allowed for servicing a query request. The App Server gives up on queries which take longer, and returns an error.
 - **Pre-commit Trigger Limit** specifies the maximum number of pre-commit triggers a single statement against this App Server can invoke. For more information on triggers, see [Using Triggers to Spawn Actions](#) in the *Application Developer's Guide*.
 - **Pre-commit Trigger Depth** specifies the maximum depth (how many triggers can cause other triggers to fire, which in turn cause others to fire, and so on) for pre-commit triggers that are executed against this App Server. For more information on triggers, see [Using Triggers to Spawn Actions](#) in the *Application Developer's Guide*.
 - **Collation** specifies the default collation for queries run in this appserver. This will be the collation used for string comparison and sorting if none is specified in the query. For details, see [Encodings and Collations](#) in the *Search Developer's Guide*.
 - **Log Errors** specifies whether to log uncaught errors for this App Server to the ErrorLog.txt file. This is useful to log exceptions that might occur on an App Server for later debugging.
 - **Debug Allow** specifies whether to allow requests against this App Server to be stopped for debugging, using the MarkLogic Server debugging APIs.
 - **Profile Allow** specifies whether to allow requests against this App Server to be profiled, using the MarkLogic Server profiling APIs. For details, see [Profiling Requests to Evaluate Performance](#) in the *Query Performance and Tuning Guide*.
 - **Default XQuery Version** specifies the default XQuery language for this App Server if an XQuery module does explicitly declare its language version.
 - The properties associated with SSL support are described in [Configuring SSL on App Servers](#) in *Securing MarkLogic Server*.

The new XDBC server is created. Creating an XDBC server is a “hot” admin task; the changes take effect immediately. For information and setup instructions for managing user sessions and/or keeping track of login attempts, see [Section 11, “Managing User Requests and Monitoring Login Attempts” \[78\]](#).

7.2. Setting Output Options for an XDBC Server

For each XDBC Server, you can set various default output options. These output options affect how data returned from the App Server is serialized. You can also set these options at the query level to override any default options. You can set serialization options to override the App Server defaults in XQuery with the `declare option XQuery` prolog, and in XSLT using the `<xsl:output>` instruction. For details on setting the serialization options in XQuery, see [Declaring Options](#) in the *XQuery and XSLT Reference Guide*. For XSLT output details, see the [XSLT specification](#).

To specify defaults for the App Server, follow these steps:

1. Click the **Groups** icon in the left tree menu.
2. Click the group which contains the XDBC server you want to view (for example, **Default**).
3. Click the **App Servers** icon on the left tree menu.
4. Select the App Server to edit.
5. Select the **Output Options** link in the left tree menu. The **Output Options Configuration** page displays.
6. Set any options that you want to control for this App Server.
7. Click **OK** to save your changes.

For more details about App Server output, see [Controlling App Server Access, Output, and Errors](#) in the *Application Developer's Guide*.

7.3. Viewing XDBC Server Settings

To view the settings for an XDBC server, follow these steps:

1. Click the **Groups** icon.
2. Click the group which contains the XDBC server you want to view (for example, **Default**).
3. Click the **App Servers** icon on the left tree menu.
4. Locate the XDBC server for which you want to view settings, either in the tree menu or on the summary page.
5. Click the icon or link for the XDBC server.
6. View the settings.

7.4. Deleting an XDBC Server

To delete the settings for an XDBC server, follow these steps:

1. Log into the Admin Interface.
2. Click on the **Groups** icon.
3. Click on the group which contains the XDBC server you want to delete (for example, **Default**).
4. Click **App Servers** on the left tree menu.
5. Locate the XDBC server to be deleted, either in the tree menu or on the summary page.
6. Click the icon for this XDBC server.
7. Click **Delete**.
8. A confirmation message displays. Confirm the delete and click **OK**.

Deleting an XDBC server is a “cold” admin task; the server restarts to reflect your changes.

7.5. Canceling a Request

To cancel a long-running request (for example, a long-running query statement or update statement), follow these steps:

1. Click **Groups** in the left tree menu.
2. Click the group which contains the server that has the request to cancel (for example, **Default**).
3. Navigate to the App Server in which the request was issued, either from the tree menu or from the summary page.
4. Click the **Status** tab.
5. At the bottom right of the **Group Status** page, click the **cancel** link on the row for the query you want to cancel.
6. Click **OK** on the **Cancel Request confirmation** page. If the request is already completed when the confirmation page occurs, the page will indicate that the request cannot be found.

The request is canceled and the App Server Status page appears again.

8. WebDAV Servers

This section describes how to use the Admin Interface to create and configure WebDAV servers.



NOTE

To create and configure WebDAV servers programmatically, see [Creating and Configuring App Servers](#) in the *Scripting Administrative Tasks Guide*.

WebDAV (Web-based Distributed Authoring and Versioning) is a protocol that extends the HTTP protocol to provide the ability to write documents through these HTTP extensions. You need a WebDAV client to write documents, but you can still read them through HTTP (through a web browser, for example). For information about WebDAV clients supported in MarkLogic Server, see [Section 8.7, “WebDAV Clients” \[60\]](#). For general information about WebDAV and the WebDAV protocol, see <http://webdav.org>.

A WebDAV server in MarkLogic Server is similar to an HTTP server with these important differences:

- WebDAV servers cannot execute XQuery code.
- WebDAV servers support the WebDAV protocol to allow WebDAV clients to have read and write access (depending on the security configuration) to a database.
- A WebDAV server only accesses documents and directories in a database; it does not access the file system directly.

8.1. Accesses a Database for Read and Write, Not XQuery Execution

In MarkLogic Server, WebDAV servers are defined at the group level and apply to all hosts within the group. Each WebDAV server provides access to a single database for reading and writing (dependent on the needed security permissions). When a document is read or written via WebDAV, all of its associated data, such as properties, metadata, collections, and so on are also transferred with the document.

In the Admin Interface, you configure a WebDAV server to access a database. Documents stored in that database are accessible for reading via HTTP. The database is also accessible via WebDAV clients for reading, modifying, deleting, and adding documents. When you add a document via a WebDAV client (by dragging and dropping, for example), you are actually loading a document directly into the database.

When accessing a database via a WebDAV server, you cannot execute XQuery code. Unlike an HTTP server, there is no Modules database for a WebDAV server. You can, however, configure a database as the Modules database of an HTTP, ODBC, or XDBC server and you can configure the same database for access from a WebDAV server. Then, you can edit code from the WebDAV server that executes from an HTTP, ODBC, or XDBC server. For an example of this configuration, see [Section 8.8, “Example: Setting Up a WebDAV Server to Add or Modify Documents Used by Another Server” \[63\]](#).

8.2. WebDAV Server Security

WebDAV servers follow the MarkLogic Server security model, as do HTTP, ODBC, and XDBC servers. The server authenticates users with user IDs and passwords stored in the security database for that WebDAV server, and the server controls access to objects in the database with privileges and roles.

(Each WebDAV server is connected to a database, and each database is in turn connected to a security database in which security objects such as users are stored.)

You can configure application-level security if you want everyone who accesses the WebDAV server to effectively log in as the same user with no password. For example, if you want everyone to log in as *guest*, where *guest* has both read and write privileges and has a predefined set of default privileges, set the authentication scheme to application-level and set the default user to *guest*.



NOTE

Because users who have write permissions to the database on a WebDAV server can load documents into the database via a WebDAV client, be sure to configure appropriate default permissions on those users so that documents they load (for example, by dragging and dropping files into a WebDAV folder) have the needed permissions for other users to read and write, according to your security policy. You can achieve such granular access control to the system and to the data through the use of privileges and permissions. For information on using security features in MarkLogic Server, see [Section 23, “Security Administration” \[212\]](#) and the sections related to security in the [Application Developer’s Guide](#).

8.3. Directories

A WebDAV directory is analogous to a file system directory. A directory must exist in order to view (via a WebDAV client) any documents in that directory (just like in a filesystem, where you must navigate to a directory in order to access any files in that directory). Each document in a directory has a URI that includes the directory URI as a prefix. Also, each directory visible from a WebDAV server must have the WebDAV root as its prefix, and there must exist a directory with the WebDAV root in the database.

For example, if you have a WebDAV root of `http://marklogic.com/`, then the URI of all documents and all directories must begin with that root in order to be visible from a WebDAV client. Also, the directory with a URI `http://marklogic.com/` must exist in the database. Therefore, a document with a URI of `http://marklogic.com/file.xml` is visible from this WebDAV server, and a directory with a URI of `http://marklogic.com/dir/` is also visible. A directory with a URI of `/dir/` and a document with a URI of `/dir/file.xml` is not visible from this server, however, because its URI does not begin with the WebDAV root.

For more details on directories and properties, see [Property Documents and Directories](#) in the *Application Developer’s Guide*.

8.3.1. Automatic Directory Creation in a Database Settings

In the configuration for a database in the Admin Interface, there is a directory creation setting. The directory creation setting specifies whether directories are created automatically when you create a document.

If you are using a WebDAV server to load documents into a database, we recommend you use the Admin Interface to set the directory creation setting for your database to `automatic`. If you create a WebDAV server that accesses a database with directory creation set to `automatic`, the root directory (required in order to access the database via a WebDAV client) is automatically created. Automatic directory creation also helps if you are loading documents manually (using `xdmp:document-load()`, for example) whose URIs include directory hierarchies that do not exist in the database. Any directory implied by a URI is automatically created with directory creation set to `automatic`.

You can also manually create and delete directories in XQuery using `xdmp:directory-create()` and `xdmp:directory-delete()`.

For details on all of the directory creation settings, see [Basic Administrative Settings \[82\]](#).

8.3.2. Properties and URIs of Directories

A directory is stored as a properties document in a MarkLogic Server database. Like a document, a directory has a URI, but the URI must end in a forward slash (/). Use the `xdmp:document-properties("uri_name")` function to retrieve the properties document for a URI, or `xdmp:document-properties()` to retrieve all of the properties documents in the database.

Properties are in the `http://marklogic.com/xdmp/property` namespace. When you create a directory (either automatically or manually), the system creates a properties document in the database with a child element named `directory`. For example, if you have a directory in your database with a URI `/myCompany/marketing/`, the following query return the following results:

```
xdmp:document-properties("/myServer/Marketing/")
=>
<prop:properties xmlns:prop="http://marklogic.com/xdmp/property">
  <prop:directory/>
</prop:properties>
```

The properties document returned does not contain the URI of the directory, but just an empty element (`prop:directory`) indicating the existence of a directory.

`xdmp:document-properties()` returns the properties documents for all documents in the database. Whenever there is a directory element in the properties document, there is a directory in the database, and calling `xdmp:node-uri()` on that element returns the URI of the directory. For example, the following query returns the URIs for all of the directories in a database:

```
declare namespace prop="http://marklogic.com/xdmp/property"
for $x in xdmp:document-properties()/prop:properties/prop:directory
return <directory-uri>{xdmp:node-uri($x)}</directory-uri>
```



NOTE

It is possible to create a document with a URI that ends in a forward slash (/). To avoid confusion with directory URIs, the best practice is to avoid creating documents with URIs that end in a forward slash.

8.4. Server Root Directory

Each WebDAV server has a concept of a *root*. The root is the top-level directory accessible from the server; you can access any documents or directories in the database that are children of the root. The root therefore serves as a prefix for all document and directory URIs accessible through the WebDAV server. You enter the WebDAV root in the Admin Interface. The root can be any valid URI. The root should always end with a forward slash (/), and if it does not, the Admin Interface will append one to the string provided.

The root should be a unique string that can serve as the top of a directory structure. It is common practice to use a WebDAV root of the form `http://<company_domain>/`, but that is not required. The following are some examples of WebDAV roots:

```
http://myCompany/marketing/
```

```
/myCompany/marketing/
```

**NOTE**

Directories cannot end in two forward slashes (//). Therefore, you cannot create a directory with a URI `http://`. If you specify a root of `http://myCompany` for a WebDAV server and `directory creation` is set to `automatic` in the database, a directory with the URI `http://myCompany/` is automatically created in the database.

Whatever the root, any documents accessible through the WebDAV server must have URIs that begin with the root. Also, any documents created through a WebDAV client (for example, by dragging and dropping into a web folder) will be loaded with URIs beginning with the WebDAV root.

For example, a document with URI `/myCompany/marketing/strategy.doc` is accessible (given the necessary security permissions) via the WebDAV server with the root `/myCompany/marketing/`, and you can create that document by dragging a document named `strategy.doc` into a web folder configured to access the WebDAV server described above.

**NOTE**

When a WebDAV client accesses a WebDAV server whose database has `directory creation` set to `automatic`, if the WebDAV root directory does not exist in that database, it is automatically created. The directory is created with no permissions, so it will only be readable by users with the `admin` role. For other users to be able to use the WebDAV server, you should add appropriate read permissions to the directory (with `xdmp:document-add-permissions`, for example). For details on document and directory permissions, see [Securing MarkLogic Server](#).

8.5. Documents in a WebDAV Server

The main purpose of a WebDAV server is to make it easy for people to store, retrieve, and modify documents in a database. The documents can be any type, whether they are text documents such as `.txt` files or source code, binary documents such as image files or Microsoft Word files, or XML documents. Because the documents are stored in a database, you can create applications that use the content in those documents for whatever purpose you need. You can also use the database backup and restore features to easily back up the content in the database.

8.6. Procedures for Creating and Managing WebDAV Servers

Use the procedures in this section to create and manage WebDAV servers.

8.6.1. Creating a New WebDAV Server

To create a new server, follow these steps:

1. Click **Groups** in the left tree menu.
2. Click the group in which you want to define the WebDAV server (for example, **Default**).
3. Click **App Servers** in the left tree menu.
4. Click the **Create WebDAV** tab.
The **WebDAV Server** page appears.
5. Go to the **Server Name** field and enter a shorthand name for this WebDAV server.
MarkLogic Server will use this name to refer to this server on display screens and in user interface controls.

6. Go to the **Root** field and enter the name of WebDAV root. This root is a string that represents the top-level of the WebDAV URI hierarchy. Any document accessible through this WebDAV server must have a URI that begins with this root string. For more details on the root, see [Server Root Directory](#).

If the root directory does not contain a forward slash, the Admin Interface adds one for you.

7. Go to the **Port** field and enter the port number through which you want to make this WebDAV server available. The port number must not be assigned to any other server.
8. Go to the **Database** field and select the database to be accessed by this WebDAV server. Multiple HTTP, ODBC, XDBC, and WebDAV servers can be connected to the same database.



NOTE

If you are using a database with a WebDAV server, the directory creation setting on the database should be set to `automatic`, which will automatically create the root directory and other directories for any documents added to the database (if the directory does not already exist). For more information on directories, see [Directories](#).

9. Scroll to the **Authentication** field. Select an authentication scheme, as described in [Types of Authentication](#) in *Securing MarkLogic Server*. The default is `digest`, which uses encrypted passwords.

If you select application-level authentication, you will also need to fill in a Default User. Any one accessing the App Server is automatically logged in as the Default User until the user logs in explicitly.



WARNING

If you use an admin user (`admin`) as the Default User (an authorized administrator with the `admin` role), then everyone who uses this App Server is automatically a user with the `admin` role, which effectively turns off security for this App Server.

10. Scroll to the **Privilege** field near the bottom of the screen. This field represents the privilege needed to access (login) the server. You may leave this field blank.
11. Set any other properties for this App Server, as appropriate to your needs:
 - **Last Login** and **Display Last Login** are described in [Storing and Monitoring the Last User Login Attempt](#).
 - **Backlog** specifies the maximum number of pending connections allowed on the HTTP server socket.
 - **Threads** specifies the maximum number of App Server threads.
 - **Request Timeout** specifies the maximum number of seconds before a socket receives a timeout for the first request.
 - **Keep Alive Timeout** specifies the maximum number of seconds before a socket receives a timeout for subsequent requests over the same connection.
 - **Session Timeout** specifies the maximum number of seconds before an inactive session times out.
 - **Max Time Limit** specifies the upper bound for any request's time limit. No request may set its time limit (for example with `xdmp:set-request-time-limit()`) higher than this number. The time limit, in turn, is the maximum number of seconds allowed for servicing a query request. The App Server gives up on queries which take longer, and returns an error.
 - **Default Time Limit** specifies the default value for any request's time limit, when otherwise unspecified. A request can change its time limit using `xdmp:set-request-time-limit()`. The time limit, in turn, is the maximum number of seconds allowed for servicing a query request. The App Server gives up on queries which take longer, and returns an error.
 - **Static Expires** adds an "expires" HTTP header for static content to expire after this many seconds.

- **Pre-commit Trigger Limit** specifies the maximum number of pre-commit triggers a single statement against this App Server can invoke. For more information on triggers, see [Using Triggers to Spawn Actions](#) in the *Application Developer's Guide*.
- **Pre-commit Trigger Depth** specifies the maximum depth (how many triggers can cause other triggers to fire, which in turn cause others to fire, and so on) for pre-commit triggers that are executed against this App Server. For more information on triggers, see [Using Triggers to Spawn Actions](#) in the *Application Developer's Guide*.
- **Collation** specifies the default collation for queries run in this appserver. This will be the collation used for string comparison and sorting if none is specified in the query. For details, see [Encodings and Collations](#) in the *Search Developer's Guide*.
- **Concurrent Request Limit** specifies the maximum number of requests any user may have running at a specific time. 0 indicates no maximum. For details, see [Managing Concurrent User Requests](#).
- **Log Errors** specifies whether to log uncaught errors for this App Server to the ErrorLog.txt file. This is useful to log exceptions that might occur on an App Server for later debugging.
- **Debug Allow** specifies whether to allow requests against this App Server to be stopped for debugging, using the MarkLogic Server debugging APIs.
- **Profile Allow** specifies whether to allow requests against this App Server to be profiled, using the MarkLogic Server profiling APIs. For details, see [Profiling Requests to Evaluate Performance](#) in the *Query Performance and Tuning Guide*.
- **Default XQuery Version** specifies the default XQuery language for this App Server if an XQuery module does not explicitly declare its language version.
- **Multi Version Concurrency Control** specifies how strict queries behave about getting the latest timestamp. This only affects query statements, not update statements. For details about queries and transactions in MarkLogic Server, see [Understanding Transactions in MarkLogic Server](#) in the *Application Developer's Guide*.
- The properties associated with SSL support are described in [Configuring SSL on App Servers](#) in *Securing MarkLogic Server*.

12. Scroll to the top or bottom and click **OK**.

The new WebDAV server is added. Adding a WebDAV server is a “hot” admin task.

8.6.2. Setting Output Options for a WebDAV Server

For each WebDAV Server, you can set various default output options. These output options affect how data returned from the App Server is serialized. You can also set these options at the query level to override any default options. You can set serialization options to override the App Server defaults in XQuery with the `declare option XQuery` prolog, and in XSLT using the `<xsl:output>` instruction. For details on setting the serialization options in XQuery, see [Declaring Options](#) in the *XQuery and XSLT Reference Guide*. For XSLT output details, see the [XSLT specification](#).

To specify defaults for the App Server, follow these steps:

1. Log into the Admin Interface.
2. Click the **Groups** icon in the left tree menu.
3. Click the group which contains the WebDAV server you want to view (for example, **Default**).
4. Click **App Servers** on the left tree menu.
5. Click the App Server to edit.
6. Click **Output Options** in the left tree menu. The **Output Options** page appears.
7. Set any options that you want to control for this App Server.
8. Click **OK** to save your changes.

For more details about App Server output, see [Controlling App Server Access, Output, and Errors](#) in the *Application Developer's Guide*.

8.6.3. Viewing WebDAV Server Settings

To view the settings for a WebDAV server, follow these steps:

1. Log into the Admin Interface.
2. Click the **Groups** icon on the left tree menu.
3. Click the group which contains the WebDAV server you want to view (for example, **Default**).
4. Click the **App Servers** icon on the left tree menu.
5. Locate the WebDAV server for which you want to view settings, either in the tree menu or on the summary page.
6. Click the name of the WebDAV server.
7. View the settings.

8.6.4. Deleting a WebDAV Server

To delete the settings for a WebDAV server, follow these steps:

1. Log into the Admin Interface.
2. Click the **Groups** icon on the left tree menu.
3. Click the group which contains the WebDAV server (or example, **Default**).
4. Click the WebDav server icon on the left tree menu.
5. Click **Delete**. A confirmation message appears.
6. Confirm the delete and click **OK**.

Deleting a WebDAV server is a “cold” admin task; the server restarts to reflect your changes.

8.6.5. Canceling a Request

To cancel a long-running request (for example, a long-running query statement or update statement), follow these steps:

1. Click **Groups** in the left tree menu.
2. Click the group which contains the server that has the request to cancel (for example, **Default**).
3. Navigate to the App Server in which the request was issued, either from the tree menu or from the summary page.
4. Click the **Status** tab.
5. At the bottom right of the **Group Status** page, click the **cancel** link on the row for the query you want to cancel.
6. Click **OK** on the **Cancel Request confirmation** page. If the request is already completed when the confirmation page occurs, the page will indicate that the request cannot be found.

The request is canceled and the App Server Status page appears again.

8.7. WebDAV Clients

A WebDAV client allows you to log into a WebDAV server to read, modify, insert, add, or delete documents. This section lists the supported WebDAV clients for MarkLogic Server and provides some general and specific procedures.

8.7.1. Tested WebDAV Clients

The following table lists WebDAV clients that have been tested with MarkLogic Server:

WebDAV Client	How to Get It	Notes
Windows Explorer	Part of Windows 10 in many configurations	Allows drag and drop from Windows. For instructions on setting up the Windows Explorer client, see Connecting to a Web Folder Using Windows 10 File Explorer . or Connecting to a Web Folder Using Windows 9 or Earlier File Explorer . Some Windows clients require digest authentication.
PerlDAV	PerlDAV -- A WebDAV client library for Perl5	A command line, perl-based WebDAV client. Designed to be scriptable and to allow you to send individual WebDAV calls.
XML Spy	Altova Software	Allows you to open, edit, and save XML files in XML Spy. Use the File > Open URL menu item in XML Spy.

WebDAV Client	How to Get It	Notes
jEdit DAV plug-in	Available on the the MarkLogic Developer site .	Allows you to view and edit database documents in jEdit 4.2. This version is available from developer.marklogic.com

For detailed information on these clients, see the documentation accompanying these products.



NOTE

Directory and document names in WebDAV (and in MarkLogic Server databases) are case-sensitive, but some WebDAV clients (Windows Explorer, for example) are not case-sensitive. While Windows recognizes case, it treats the directory named `NewFolder` as the same directory as one named `newFolder`. Therefore, directory or document names that differ only in case might cause confusion when using Windows Explorer or other case-insensitive WebDAV clients. If possible, avoid assigning names to directories or documents that differ only by case (for example, `NewFolder` vs `newFolder`).

Windows WebDAV clients will cause two transactions upon initial document creation: the first is a 0-length WebDAV PUT resulting in a new 0-length document, and the second is an update to the 0-length document. If you are using CPF (or other applications that use triggers), this will fire both the create trigger (when the initial 0-length document is created) and the update trigger (when the document is updated with its contents). When using Windows WebDAV clients with CPF applications, make sure that your CPF actions for create and update are designed to work correctly for this behavior. In most cases, having the same action for create and update will be sufficient, but in some cases, you might need to write an action that checks for a 0-length document and does something special with it.

8.7.2. General Steps to Connect to a Server

Each WebDAV client has its own way of connecting to a WebDAV server, but these are the general steps to connect to a WebDAV server:

1. Start the WebDAV client.
2. Enter the connection information for the WebDAV server. This includes the servername and port number of the WebDAV server. For example, if you have a WebDAV server running on port 9001 on a machine named `marklogic.myCompany.com`, enter `http://marklogic.myCompany.com:9001/` in the appropriate place for your WebDAV client.
3. If prompted, enter a username and password for the WebDAV server. You will be prompted for a username or password unless you have configured application-level security.



NOTE

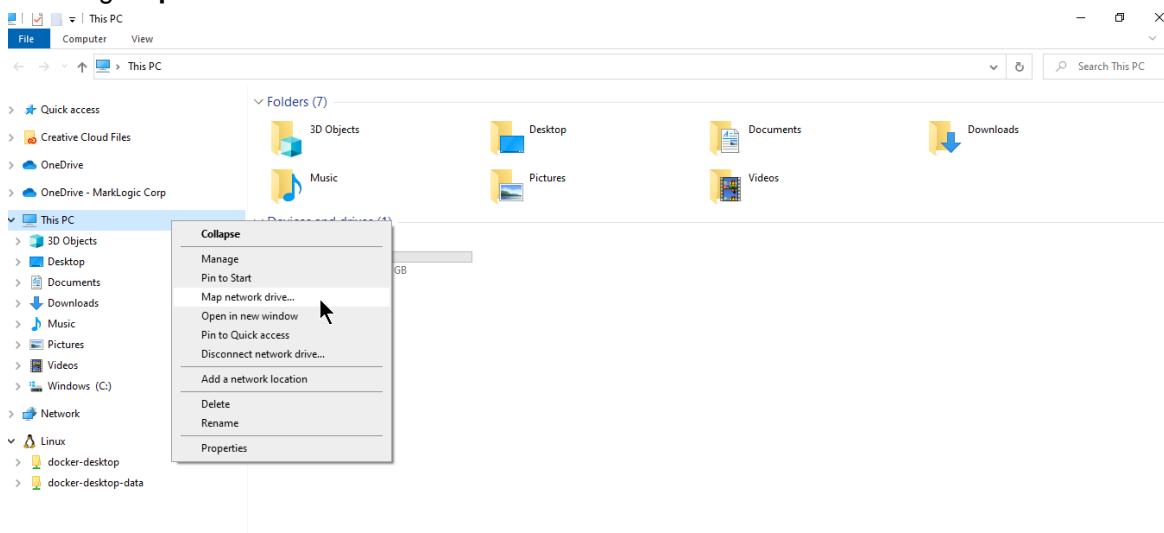
The user who logs into the WebDAV server must have the needed privileges (granted via roles) to access the documents and directories under the WebDAV root directory. Also, if you want the WebDAV user to create documents under the WebDAV root, then that user must have the needed URI privileges (granted via roles) to create documents under the root. The lack of any needed privileges and/or permissions can cause the WebDAV login or other WebDAV activities to fail. For details on URI privileges and document permissions, see [Securing MarkLogic Server](#).

- Use whatever browsing mechanism the client supports to add, remove, or modify documents and directories. For example, in Windows Explorer, double click on folders to expand them, drag and drop documents into folders, rename documents and directories, and so on.

8.7.3. Connecting to a Web Folder Using Windows 10 File Explorer

To connect to a Web Folder in Windows 10, follow these steps:

- Launch File Explorer by holding the `Windows Key + E`.
- Map a network drive by looking in the left pane of File Explorer and right-clicking on **This PC** and selecting **Map network drive...**



- In the **Drive** field, select a drive letter.
- Click on **Connect to a Web site that you can use to store your documents and pictures**.
- Click **Next**.
- Click **Choose a custom network location**.
- Click **Next**.
- Enter or select the Internet or network address.
- Click **Next**



NOTE

If required, you may be prompted for a username and password. Enter the values as appropriate and click **Next**.

- In the **Type a name for this network location** field, enter a name.
- Click **Next**.
- A confirmation screen appears. Click **Finish**.

You can now use this folder like other Windows folders to drag and drop documents, rename documents, and so on. When you drag and drop a file into a WebDAV folder connected to a MarkLogic Server WebDAV server, you will actually load that document into the database.

8.7.4. Connecting to a Web Folder Using Windows 9 or Earlier File Explorer

To connect to a Web Folder in Windows 9 or earlier, follow these steps:

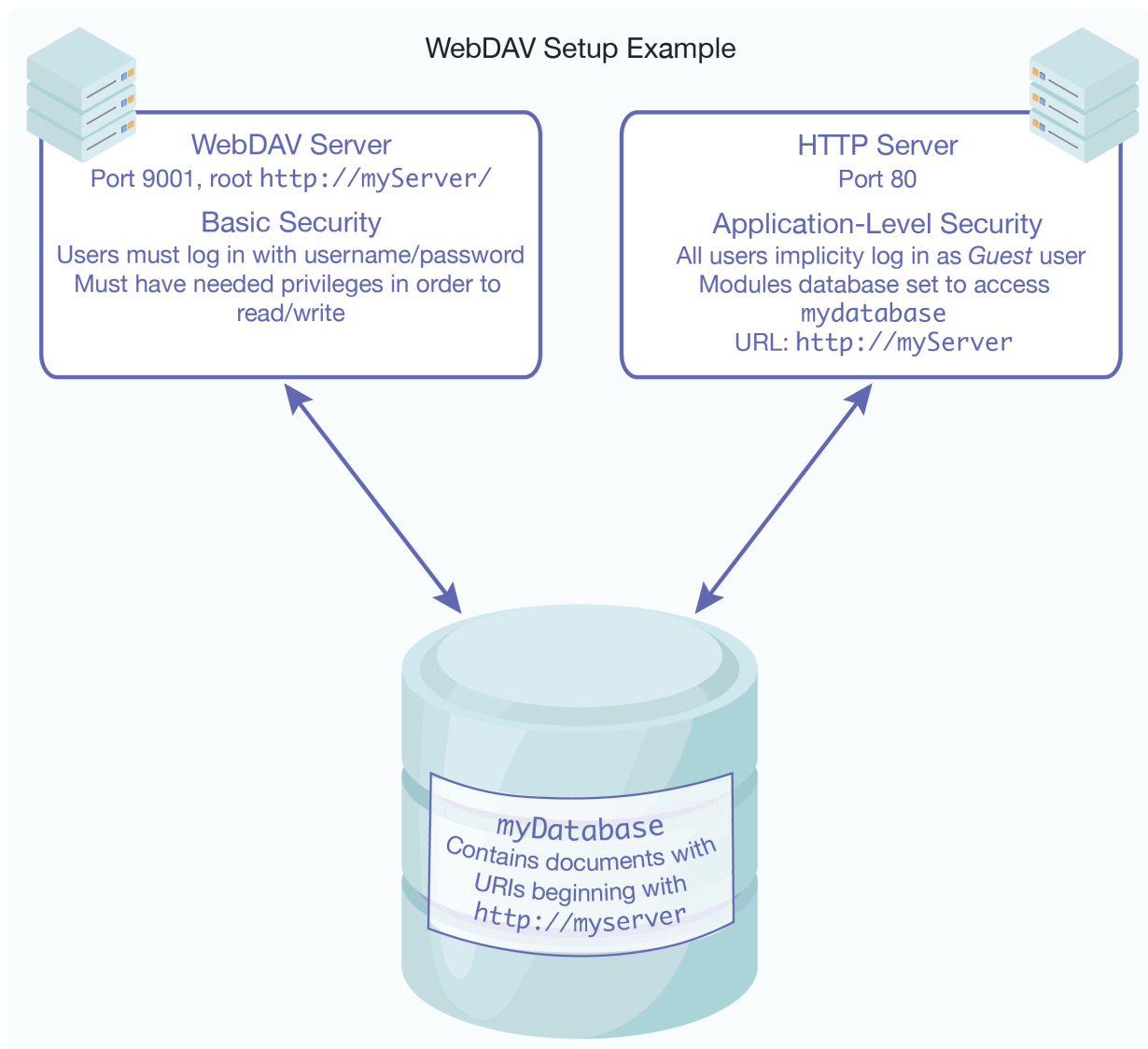
- Double-click the **My Network Places** icon on your desktop.
- In My Network Places, double-click the **Add Network Places** icon.
- In the **Add Network Place** Wizard, enter your WebDAV server address and port number. For example, if you have a WebDAV server running on port 9001 on a machine named `marklogic.myCompany.com`, enter `http://marklogic.myCompany.com:9001/`

4. Click **Next**.
5. If prompted, enter your username and password for the WebDAV server.
6. Enter a name for the network place and click **Finish**.

You can now use this folder like other Windows folders to drag and drop documents, rename documents, and so on. When you drag and drop a file into a WebDAV folder connected to a MarkLogic Server WebDAV server, you will actually load that document into the database.

8.8. Example: Setting Up a WebDAV Server to Add or Modify Documents Used by Another Server

You can use a WebDAV server to provide privileged users write access to a database (via a WebDAV client). That database, in turn, might also be used as a Modules database in one or more other servers (HTTP, ODBC, WebDAV, and/or XDBC) to provide read and execute access. Consider the scenario shown in the following figure:



In this scenario, all users can view the content by going to the URL `http://myServer/` in their web browsers. No password is needed to access this server because it is set up with application-level security, using a default user named *Guest*. The *Guest* user only has read permissions. If there is content that you do not want the *Guest* user to access, load that content with privileges that the *Guest* user does not have.

Meanwhile, users with the proper privileges can log in through a WebDAV client to access the WebDAV server at port 9001. Because the WebDAV server is configured with basic security, users are prompted for a username and password when they access the server through the WebDAV client (or through a web browser connected to port 9001). From the WebDAV client, they can add documents, edit documents, or read documents according to the database security policy.

For information about a Modules database, see [Section 12.1.2, “Modules Database” \[81\]](#).

9. ODBC Servers

This section describes how to use the Admin Interface to create and configure ODBC servers.



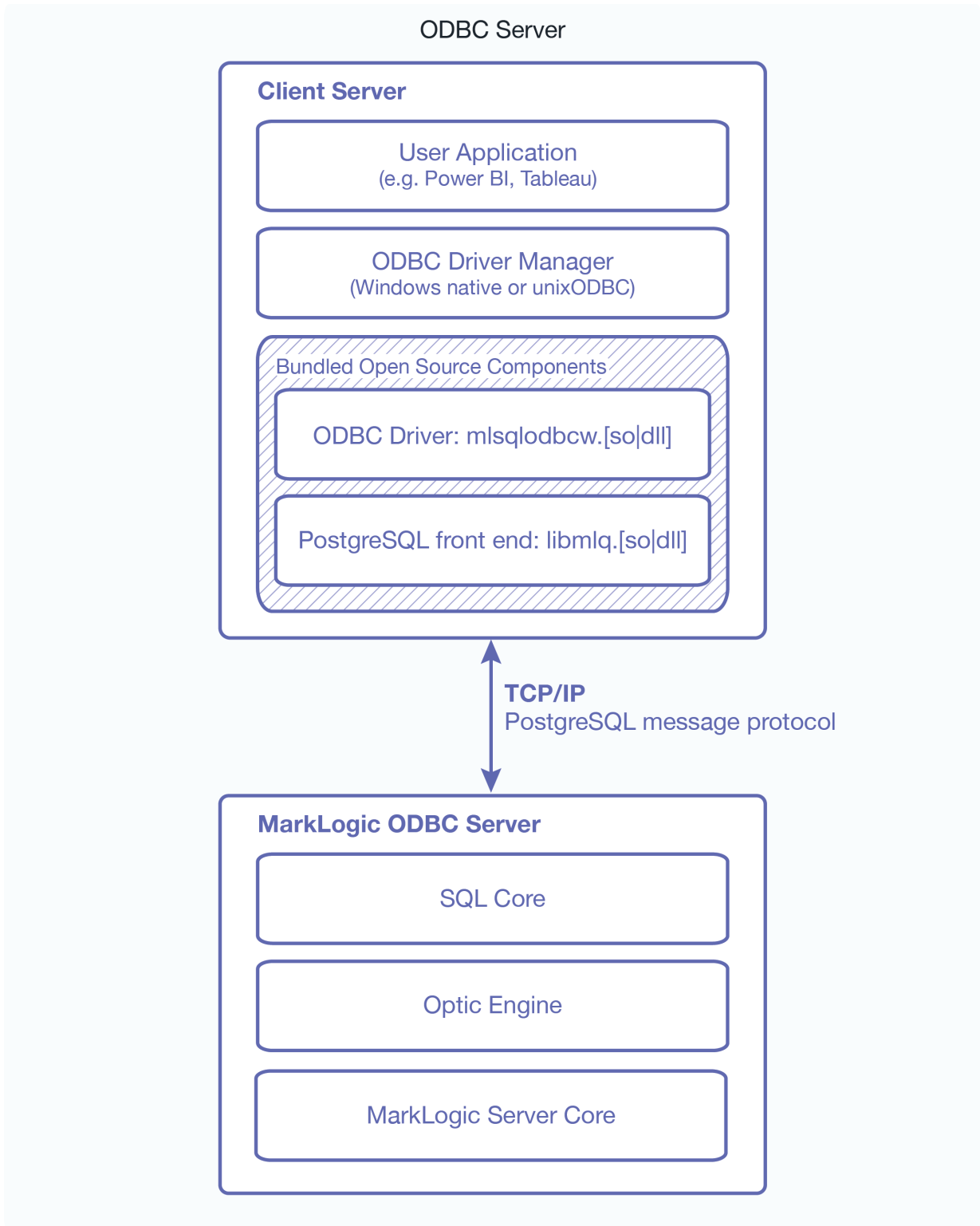
NOTE

To create and configure ODBC servers programmatically, see [Creating and Configuring App Servers](#) in the *Scripting Administrative Tasks Guide*.

An ODBC server is one of several components that support SQL queries to MarkLogic Server.

The basic purpose of an ODBC server is to return relational-style data resident in MarkLogic Server in response to SQL queries. The ODBC server returns data in tuple form and manages server state to support a subset of SQL and ODBC statements from Business Intelligence (BI) tools.

As shown in the figure below, an ODBC server connects with a PostgreSQL front end on the client by means of the PostgreSQL message protocol. The ODBC server accepts SQL queries from the PostgreSQL front end and returns the relational-style data needed by the BI applications to build reports.



9.1. Creating a New ODBC Server

To create a new server, follow these steps:

1. Click **Groups** in the left tree menu.
2. Click the group in which you want to define the ODBC server (for example, **Default**).
3. Click **App Servers** in the left tree menu.
4. Click the **Create ODBC** tab at the top right. The **Odbc server** page appears:

5. In the **Odbc Server Name** field, enter a shorthand name for this ODBC server. MarkLogic Server will use this name to refer to this server on display screens in the Admin Interface.
6. In the **Root** field, enter the name of the directory in which you will store your data. If the **Modules** field is set to a database, then the root must be a directory URI in the specified modules database. If the **Modules** field is set to **file system**, then the root directory is either a fully-qualified pathname or is relative to the directory in which MarkLogic Server is installed. The following table shows the default installation directory for each platform:

Platform	Program Directory
Microsoft Windows	C:\Program Files\MarkLogic
Red Hat Linux	/opt/MarkLogic
Mac OS X	~/Library/MarkLogic

**NOTE**

Unless you specify a shared drive, all hosts in the group will need to have a copy of the XQuery programs in the directory specified above.

**WARNING**

Do not create ODBC server root directories named Docs, Data, or Admin. These directories are reserved by MarkLogic Server for other purposes. Creating ODBC server root directories with these names can result in unpredictable behavior of the server and may also complicate the software upgrade process.

7. In the **Port** field, enter the port number through which you want to make this ODBC server available. The default PostgreSQL listening socket port is 5432. The port number must be unique to this ODBC server and must not be assigned to any other ODBC, HTTP, XDBC or WebDAV server.
8. In the **Modules** field, select the database to use as the modules database for your XQuery documents, or leave it at the default of storing your XQuery modules on the file system. For information on what a modules database is, see [Section 12.1.2, “Modules Database” \[81\]](#).
9. In the **Database** field, select the database to be accessed by this ODBC server. This database should be set up with the range indexes and schema views to support the SQL application. For details on how to set up a database to support SQL applications, see the [SQL Data Modeling Guide](#). Multiple ODBC, HTTP, XDBC, and WebDAV servers can access the same database.
10. Scroll to the **Authentication** field. Select an authentication scheme, as described in [Types of Authentication](#) in *Securing MarkLogic Server*. The default is **digest**, which uses encrypted passwords.
If you select **application-level in authentication**, you will also need to fill in a **Default User**. Anyone accessing the ODBC server is automatically logged in as the Default User until the user logs in explicitly.

**WARNING**

If you use an admin user (admin) as the Default User (an authorized administrator with the `admin` role), then everyone who uses this App Server is automatically a user with the `admin` role, which effectively turns off security for this App Server.

11. Scroll to the **Privilege** field near the bottom of the screen. This field represents the privilege needed to access (login to) the server. You may leave this field blank.
A user accessing the ODBC server must have the `execute` privilege selected in order to access the ODBC server. If you chose application-level authentication above, you should ensure that the default user has the selected privilege.

12. Set any other properties for this App Server, as appropriate to your needs:
 - **Last Login** and **Display Last Login** are described in [Section 11.3, “Storing and Monitoring the Last User Login Attempt” \[79\]](#).
 - **Backlog** specifies the maximum number of pending connections allowed on the ODBC server socket.
 - **Threads** specifies the maximum number of App Server threads.
 - **Request Timeout** specifies the maximum number of seconds before a socket receives a timeout for the first request.
 - **Keep Alive timeout** specifies the maximum number of seconds before a socket receives a timeout for subsequent requests over the same connection.
 - **Session Timeout** specifies the maximum number of seconds before an inactive session times out.
 - **Max Time Limit** specifies the upper bound for any request's time limit. No request may set its time limit (for example with `x Demp:set-request-time-limit()`) higher than this number. The time limit, in turn, is the maximum number of seconds allowed for servicing a query request. The App Server gives up on queries which take longer, and returns an error.
 - **Default Time Limit** specifies the default value for any request's time limit, when otherwise unspecified. A request can change its time limit using `x Demp:set-request-time-limit()`. The time limit, in turn, is the maximum number of seconds allowed for servicing a query request. The App Server gives up on queries which take longer, and returns an error.
 - **Static Expires** adds an "expires" ODBC header for static content to expire after this many seconds.
 - **Pre-commit Trigger Limit** specifies the maximum number of pre-commit triggers a single statement against this App Server can invoke. For more information on triggers, see [Using Triggers to Spawn Actions](#) in the *Application Developer's Guide*.
 - **Pre-commit Trigger Depth** specifies the maximum depth (how many triggers can cause other triggers to fire, which in turn cause others to fire, and so on) for pre-commit triggers that are executed against this App Server. For more information on triggers, see [Using Triggers to Spawn Actions](#) in the *Application Developer's Guide*.
 - **Collation** specifies the default collation for queries run in this appserver. This will be the collation used for string comparison and sorting if none is specified in the query. For details, see [Encodings and Collations](#) in the *Search Developer's Guide*.
 - **Concurrent Request Limit** specifies the maximum number of requests any user may have running at a specific time. 0 indicates no maximum. For details, see [Section 11.1, “Managing Concurrent User Requests” \[78\]](#).
 - **Log Errors** specifies whether to log uncaught errors for this App Server to the ErrorLog.txt file. This is useful to log exceptions that might occur on an App Server for later debugging.
 - **Debug Allow** specifies whether to allow requests against this App Server to be stopped for debugging, using the MarkLogic Server debugging APIs.
 - **Profile Allow** specifies whether to allow requests against this App Server to be profiled, using the MarkLogic Server profiling APIs. For details, see [Profiling Requests to Evaluate Performance](#) in the *Query Performance and Tuning Guide*.
 - **Default XQuery Version** specifies the default XQuery language for this App Server if an XQuery module does explicitly declare its language version.
 - **Multi Version Concurrency Control** specifies how strict queries behave about getting the latest timestamp. This only affects query statements, not update statements. For details about queries and transactions in MarkLogic Server, see [Understanding Transactions in MarkLogic Server](#) in the *Application Developer's Guide*.
 - The **Error Handler** and **URL Rewriter** fields are described in [Controlling App Server Access, Output, and Errors](#) in the *Application Developer's Guide*.
 - The properties associated with SSL support are described in [Configuring SSL on App Servers in Securing MarkLogic Server](#).
13. Scroll to the top or bottom and click **OK**.

The ODBC server is now created. Creating an ODBC server is a “hot” admin task; the changes take effect immediately. For information and setup instructions for managing user sessions and/or keeping track of login attempts, see [Section 11, “Managing User Requests and Monitoring Login Attempts” \[78\]](#).

9.2. Setting Output Options for an ODBC Server

For each ODBC Server, you can set various default output options. These output options affect how data returned from the App Server is serialized. You can also set these options at the query level to override any default options. You can set serialization options to override the App Server defaults in XQuery with the `declare option XQuery` prolog, and in XSLT using the `<xsl:output>` instruction. For details on setting the serialization options in XQuery, see [Declaring Options](#) in the *XQuery and XSLT Reference Guide*. For XSLT output details, see the [XSLT specification](#).

To specify defaults for the App Server, follow these steps:

1. Click **Groups** in the left tree menu.
2. Click the group which contains the ODBC server you want to view (for example, **Default**).
3. Click **App Servers** in the left tree menu.
4. Select the **Output Options** link in the left tree menu. The **Output Options** page displays.
5. Set any options that you want to control for this App Server.
6. Click **OK** to save your changes.

For more details about App Server output, see [Controlling App Server Access, Output, and Errors](#) in the *Application Developer’s Guide*.

9.3. Viewing ODBC Server Settings

To view the settings for a particular ODBC server, follow these steps:

1. Click **Groups** in the left tree menu.
2. Click the group which contains the ODBC server you want to view (for example, **Default**).
3. Click **App Servers** in the left tree menu.
4. Locate the ODBC server for which you want to view settings, either in the tree menu or on the summary page.
5. Click the ODBC server.
6. View the settings.

9.4. Deleting an ODBC Server

To delete the settings for an ODBC server, follow these steps:

1. Click **Groups** in the left tree menu.
2. Click the group which contains the ODBC server you want to delete (for example, **Default**).
3. Click **App Servers** in the left tree menu.
4. Locate the ODBC server you want to delete, either in the tree menu or on the summary page.
5. Click the ODBC server.
6. Click **Delete**.
7. A confirmation message displays. Confirm the delete and click **OK**.

Deleting an ODBC server is a “cold” admin task; the server restarts to reflect your changes.

9.5. Canceling a Request

To cancel a long-running request (for example, a long-running query statement or update statement), follow these steps:

1. Click **Groups** in the left tree menu.

2. Click the group which contains the server that has the request to cancel (for example, **Default**).
3. Navigate to the App Server in which the request was issued, either from the tree menu or from the summary page.
4. Click the **Status** tab.
5. At the bottom right of the **Group Status** page, click the **cancel** link on the row for the query you want to cancel.
6. Click **OK** on the **Cancel Request confirmation** page. If the request is already completed when the confirmation page occurs, the page will indicate that the request cannot be found.

The request is canceled and the App Server Status page appears again.

9.6. ODBC Request Monitoring and Cancellation

Request monitoring is supported for the ODBC App server. The following ODBC meters will be recorded for ODBC Requests:

Meter	Description
odbcRowsSent	The number of rows sent from the server over the network
odbcBytesSent	The total number of bytes sent from the server over the network

Prior to this feature, the Modules database assigned to an ODBC server served no purpose. Now, a file named `/default.api` is looked for in the Modules root - whether that be the filesystem or a Modules database - and the configuration defined in that file is heeded to.

When a query comes in over ODBC, the `odbcRowsSent` and `odbcBytesSent` meters and their values are output in the ODBC App server request log, given that Request Monitoring is enabled in `/default.api`.

Request cancellation is now enabled for ODBC server as well. The `/default.api` will take in unsigned long values as limits for `odbcRowsSent` and `odbcBytesSent`. If the number of rows or bytes sent exceed the limits defined in `/default.api`, the sending of rows over the network ends prematurely and SQL-ODBCREQLIMIT is thrown. `elapsedTime` is also supported for cancellation, and XDMP-EXTIME is thrown if the time limit is exceeded. `lockCount` and `readSize` are not relevant for ODBC.

These values are purely e-node metrics, and are not recorded on any d-nodes.

Configuration example:

With the following `/default.api` in the Modules root, a query will be canceled for one of these reasons:

- Run time exceeds 3 seconds.
- The number of bytes sent to the client exceeds 200000.
- The number of rows sent to the client exceeds 10000.

```
{
  "monitoring": {
    "general": {
      "enabled": true
    },
  },
  "limits" : {
    "elapsedTime" : 3,
    "odbcBytesSent": 200000,
    "odbcRowsSent": 10000
  }
}
```

10. Auditing Events

MarkLogic Server provides an auditing facility to audit various events such as document read access, server startup, server shutdown, document permission changes, and so on. These audit event records are logged to audit files stored under the MarkLogic Server data directory for each instance of MarkLogic Server. This section describes the auditing features.

10.1. Overview of Auditing

Auditing in MarkLogic Server enables you to specify which events should generate an audit event record. You can choose from a large list of events to audit, and can restrict audit events based on various identities (user, role, or document URI). This section describes the logging capabilities of MarkLogic Server.



NOTE

Enabling auditing events may impact system performance. The level of impact will depend upon workload, available system resources, and the events that have been enabled.

10.1.1. Audit Log Files

When auditing is enabled, MarkLogic Server writes audit events to the `AuditLog.txt` file. Each host in a cluster maintains its own audit log files. Some actions might trigger multiple audit events, and those events might be logged over multiple hosts, as events are audited on the host in which the event occurs. For more information about the audit events, see [Section 10.2, “Auditable Events” \[72\]](#). Note the following about the audit event log files:

- Writes messages to `AuditLog.txt` file for various events.
- Each event has a timestamp, event type, user, role, and other information relevant to the event (for example, document URI for document-read event). For an example of log entries, see [Section 10.2.2, “Sample Audit Logs” \[76\]](#).
- You can configure how often to rotate the audit files (similar to the log files, as described in [Section 29, “Log Files” \[259\]](#)).
- The Audit log files are stored in the same directory as the Access log files (`port_AccessLog.txt`) and the Error log files (`ErrorLog.txt`), which is in the `<marklogic-data-dir>/Logs` directory. These files are private to the host in which the audit event occurred.
- You may view the current or any archived file log at any time using standard text file viewing tools. Additionally, you can access the log files from the Log tab on the main page of the Admin Interface.

The following table shows the location of the `AuditLog.txt` files on the various platforms.

Platform	Audit File
Microsoft Windows	<code>C:\Program Files\MarkLogic\Data\Logs\AuditLog.txt</code>
Red Hat Linux	<code>/var/opt/MarkLogic/Logs/AuditLog.txt</code>
Mac OS X	<code>~Library/Application Support/MARKlogic/Data/Logs/AuditLog.txt</code>

10.1.2. Restricting Audit Events

You can configure auditing to restrict events that are audited based on the following criteria:

- You can select which events to audit.
- You can include or exclude events by user name. For included users, only events initiated by the named users are audited. For excluded users, only events initiated by users other than the named users are audited.
- You can include or exclude events by role. For included roles, only events initiated by users with the included roles are audited. For excluded roles, only events initiated by users who do not have the excluded roles are audited.
- You can include or exclude events by outcome of event (success/failure/both).
- You can include or exclude events by document URI. Documents URIs are audited if any fragment from that document is loaded into memory, and that audit event is written to the audit log on the host in which the forest that contains the document resides.

For the procedure to set up auditing, see [Section 10.3.3, “Configuring Auditing to Audit Certain Events and Set Up Certain Restrictions” \[77\]](#).

10.1.3. Audit Successful, Unsuccessful, or Both Types of Events

You can choose to audit only unsuccessful, only successful, or both types of events. If you audit many events and/or if you audit both successful and unsuccessful events, then you may end up auditing a lot of events. It is not really a problem to audit many events, but it might make your audit logs get very large very fast. For the procedure to set up auditing, see [Section 10.3.3, “Configuring Auditing to Audit Certain Events and Set Up Certain Restrictions” \[77\]](#).

10.1.4. Enabled at the Group Level

You can enable or disable auditing for each group. If auditing is enabled for a group, any configured auditable event for that group is audited. For details on the procedure to enable auditing, see [Section 10.3.1, “Enabling Auditing for a Group” \[76\]](#).

10.2. Auditable Events

There are many auditable events in MarkLogic Server. When auditing is enabled, any enabled auditable event logs are written to the `AuditLog.txt` file. In a clustered environment, audit events are written to the audit file on the host in which the event occurs. Some activities might result in audit events that are distributed over multiple hosts, because events are audited on the host in which the event occurs. For example, the document access audit events are audited on the data node where the forest containing the document is hosted, therefore if a query that updates a document is run, it could cause (depending on the audit configuration and the cluster configuration) audit events to occur on the node in which the query is evaluated (the evaluation-node) and on one or more data-nodes where the affected documents are hosted.

The following table lists the auditable events you can enable in MarkLogic Server:

Event	Description	URI Restrictions	Role/User Restrictions	Success or Failure Restrictions
amp-usage	Audits the URI of an amp when it is evaluated.	Yes, based on the URI of the amp	Yes	Success Only
audit-configuration-change	Audits the success or failure of a change to an auditing configuration.	N/A	Yes	Yes
audit-shutdown	Audits when the audit system is disabled.	N/A	Yes	Yes
audit-startup	Audits when the audit system is enabled. Note that this event does not occur when MarkLogic Server starts up, only when the audit system is enabled.	N/A	Yes	Yes

Event	Description	URI Restrictions	Role/User Restrictions	Success or Failure Restrictions
authentication-failure	Audits failed authentication attempts.	N/A	Yes	Failure Only
concurrent-request-denial	Audits when a request is denied because the concurrent request limit on the App Server was reached.	N/A	Yes	Failure Only
configuration-change	Audits the success or failure of a change to a configuration file, including the path to the configuration file that changed.	N/A	Yes	Yes
document-execute	Audits when a document in a database is executed (for example, an XQuery document) and includes the document URI in the audit record.	Yes	Yes	Success Only
document-insert	Audits when a new document is created and includes the document URI in the audit record.	Yes	Yes	Success Only
document-protect	Audits the document URI when a temporal document is protected from certain operations.	Yes	Yes	Success Only
document-read	Audits when a document is read and includes the document URI in the audit record.	Yes	Yes	Success Only
document-update	Audits when a document is updated and includes the document URI in the audit record.	Yes	Yes	Success Only
document-wipe	Audits when a temporal document is wiped (all versions deleted) and includes the document URI in the audit record.	Yes	Yes	Success Only
estimate	Audits when an <code>xmdp:estimate</code> expression is evaluated.	N/A	Yes	Success Only
eval	Audits when a path expression that accesses the database is evaluated.	N/A	Yes	Success Only
exists	Audits when an <code>xmdp:exists</code> expression is evaluated.	N/A	Yes	Success Only
external-authentication-failure	Audits when an external authorization attempt fails.	N/A	Yes	Success Only
FIPS-Disabled	Audits when FIPS mode is disabled.	N/A	N/A	Success Only
FIPS-Enabled	Audits when FIPS mode is enabled.	N/A	N/A	Success Only
HTTP-client-authentication-failure	Audits failed HTTP client authentication attempts.	N/A	Yes	Failure Only
internal-keystore	Audits all operations on the internal KMS.	No	No	No
LDAP-client-authentication-failure	Audits failed LDAP client authentication attempts.	N/A	Yes	Failure Only

Event	Description	URI Restrictions	Role/User Restrictions	Success or Failure Restrictions
lexicon-read	Audits when a value lexicon (for example, <code>cts:element-values</code>) call is used.	N/A	Yes	Success Only
login-dynamic-roles	Audits the user and role information when the user is logged in with dynamic roles.	No	No	Success Only
mlcp-copy-export-finish	Audits when an mlcp copy or export job has completed whether or not it is successful.	N/A	N/A	No
mlcp-copy-export-start	Audits when an mlcp copy or export job is about to start.	N/A	N/A	Success Only
no-permission	Audits when an operation fails because of a <code>SEC-PERMDENIED</code> exception, which happens when an operation on a document (insert, update, or execute) is attempted without the needed permissions.	Yes	Yes	Failure Only
no-privilege	Audits when a user has insufficient privileges to perform a particular function.	Yes	Yes	Failure Only
optic	Audits when an Optic call completes.	N/A	Yes	Success Only
permissions-change	Audits when permissions on a document are modified.	Yes	Yes	Yes
PKI-system [v11.1.0 and up]	Audits when any internal server operations occur that are related to PKI or KMS usage.	N/A	N/A	No
PKI-user [v11.1.0 and up]	Audits when any public API operations occur that are related to PKI or KMS usage.	N/A	N/A	No
qconsole-eval	Audits queries evaluated in the query console.	No	No	No
request-blackout-denial	Audits when a request is denied due to a request blackout period.	N/A	Yes	Failure Only (when denied)
role-change-failure	Audits when adding or removing a user role fails.	N/A	Yes	Failure Only
role-query-change-failure	Audits when <code>QBAC</code> fails to add a user role to or remove a user role from a query.	No	No	Failure Only
search	Audits when a <code>cts:search</code> expression is evaluated.	N/A	Yes	Success Only

Event	Description	URI Restrictions	Role/User Restrictions	Success or Failure Restrictions
security-access	Audits when one of these security-related functions is called: <ul style="list-style-type: none"> • <code>xdmp:can-grant-roles()</code> • <code>xdmp:has-privilege()</code> • <code>xdmp:user-roles()</code> • <code>xdmp:role-roles()</code> • <code>xdmp:privilege-roles()</code> • <code>xdmp:amp-roles()</code> • <code>xdmp:get-current-roles()</code> • <code>xdmp:user()</code> • <code>xdmp:role()</code> • <code>xdmp:amp()</code> 	N/A	Yes	Yes
server-restart	Audits when MarkLogic Server is restarted with a clean restart (for example, from the Admin Interface).	N/A	Yes	Success Only
server-shutdown	Audits when MarkLogic Server is shut down with a clean shutdown (for example, from the shutdown scripts or from the Admin Interface).	N/A	Yes	Success Only
server-startup	Audits when MarkLogic Server starts up.	N/A	N/A	Success Only
SMTP-client-authentication-failure	Audits failed SMTP client authentication attempts.	N/A	Yes	Failure Only
SPARQL	Audits when a SPARQL call completes.	N/A	Yes	Success Only
SQL	Audits when a SQL call completes.	N/A	Yes	Success Only
temporal-override	Audits when the user overrides system-managed metadata for temporal documents.	No	No	No
TLS-Failure	Audits when a TLS or SSL request fails and includes the IP address.	N/A	N/A	Failure Only
user-configuration-change	Audits when anything in a user configuration changes.	N/A	Yes	Yes
user-role-addition	Audits when a user role is added.	N/A	Yes	Yes
user-role-query-addition	Audits when QBAC adds a user role to a query.	No	No	Success Only
user-role-query-removal	Audits when QBAC removes a user role from a query.	No	No	Success Only
user-role-removal	Audits when a user role is removed.	N/A	Yes	Yes

10.2.1. Audit Log Content

The information included in an audit log depends on the type of event. All audit log entries include basic information such as the event type, user, success, and roles assigned to the user. Audit log entries may include the following space-separated fields:

Log Entry Field	Description	Example
Timestamp	Contains the date and time the auditable action occurred.	2012-03-26 10:55:53.735
Event	The name of the event that triggered the log entry. The possible auditable events are listed in Section 10.2, "Auditable Events" [72] .	event=amp-usage
Function	The function that was being executed during the event.	function=http://marklogic.com/xdmp/admin:read-config-file
Expression	The query expression that triggered this audit event.	expr=cts:element-value-query(xs:QName("info:status"), ("active", "unloading"), ("unstemmed", "lang=en"), 1)
Type	The type of task inside the MarkLogic server that generated the specific event.	type=node-update
URI	The document URI involved in the event.	uri=/queries/5523898374388210414.txt
Database	The database that was accessed during the event.	database=Security
Outcome	This indicates the success or failure of the action that triggered the audit event.	success=true
User	The user that performed the action.	user=infostudio-admin
Roles	The roles assigned to the user performing the action.	roles=cpf-restart,infostudio-user

10.2.2. Sample Audit Logs

Here are some sample `AuditLog.txt` entries with user-specific information obfuscated.

```
2018-12-05 02:23:15.302 event=SMTP-client-authentication-failure; user=daemon;
host=smtp.marklogic.com; success=false;
2018-12-05 02:42:11.515 event=HTTP-client-authentication-failure; user=xyz;
type=digest; url=http://localhost:2975/qstring.sjs?sname=http-auth-digestbasic-modules-
db; success=false;
2018-12-05 02:41:50.036 event=LDAP-client-authentication-failure; url=ldap://
dc1.mltest1.local:389; success=false;
```

10.3. Configuring Auditing for a Group

Auditing is configured at the group level using the Auditing page of the Admin Interface. For details on groups, see [Section 5, "Groups" \[34\]](#). This section describes audit configuration procedures.

10.3.1. Enabling Auditing for a Group

To enable auditing for a group, follow these steps:

1. Click **Groups** in the left tree menu.
2. Click the group for which you want to configure auditing (for example, **Default**).
3. Click **Auditing**.
4. Configure any audit events and/or audit restrictions you want.
5. Click **OK**.

10.3.2. Disabling Auditing for a Group

To disable auditing for a group, follow these steps:

1. Click **Groups** in the left tree menu.
2. Click the appropriate group (for example, **Default**).
3. Click **Auditing**.
4. Select **False** in the **Audit Enabled** option group.
5. Click **OK**.

This will immediately disable auditing for the group. Any settings you had configured will remain, but they will not be in effect until you enable auditing again.

10.3.3. Configuring Auditing to Audit Certain Events and Set Up Certain Restrictions

Follow these steps to configure audit events and audit restrictions. Your procedure will vary depending on what events and restrictions you choose to configure:

1. Click **Groups** in the left tree menu.
2. Click the group for which you want to configure auditing (for example, **Default**).
3. Click **Auditing**.
4. Under **Audit Restrictions**, enter any restrictions you want. For details on audit restrictions, see [Section 10.1.2, "Restricting Audit Events" \[71\]](#).
5. Click **OK** to save your changes.

11. Managing User Requests and Monitoring Login Attempts

MarkLogic Server provides facilities to control and manage user requests and monitoring login attempts. This section describes how to use and manage these features.

11.1. Managing Concurrent User Requests

MarkLogic Server allows you to limit the maximum number of concurrent user requests against a given App Server. This section describes this feature and provides information on configuring the concurrent request limit.

11.1.1. Limiting Concurrent Requests with User Request Limits

There is an option on each App Server (HTTP, ODBC, XDBC, and WebDAV Server) configuration to limit the number of *concurrent requests* a user can have against that App Server. A concurrent request is a request against that App Server from the same user while another request from the same user is still active. Each App Server has a `concurrent request limit` configuration parameter. The default is 0, which means there is no limit to the number of concurrent requests. The value must be an integer greater than or equal to 0.

If you set the `concurrent request limit` configuration parameter to a value other than 0, it limits the number of concurrent requests any user can run against that App Server to the specified number. For example, if you set the number to 3, then any requests made by a user named `raymond` while 3 requests from `raymond` are running will fail with an exception.

When the limit is reached, the application will throw a 403 (forbidden) error with the `XDMP-REQUESTLIMIT` exception.

11.1.2. Configuring User Concurrent Request Controls

To configure a user concurrent request limit, follow these steps in the Admin Interface:

1. Click **Groups** in the left tree menu.
2. Click the group in which the App Server you want to configure resides (for example, **Default**).
3. Click **App Servers** in the left tree menu.
4. Select the App Server in which you want to configure concurrent request limits. The configuration page appears.
5. In the **Concurrent Request Limit** field, under the **External Securities** section, enter a value corresponding to the maximum number of concurrent user requests you want to allow. For example, if you want only 3 concurrent requests, enter 3. A value of 0 means there is no concurrent request limit (unlimited).
6. Click **OK** to save the configuration change.

For new requests, the new `concurrent request limit` will be enforced.

11.2. Setting Request Blackouts on an App Server

MarkLogic Server allows you to manage when a user or group of users cannot run requests against an App Server. You can manage these blackout periods for each App Server by setting up one or more Request Blackouts for an App Server. Request blackouts can specify users, roles, and time periods for the blackouts, as well as specifying if it is a one-time blackout or a recurring blackout.

11.2.1. Configuring Request Blackouts

To configure request blackout periods, follow these steps:

1. Click **Groups** in the left tree menu.
2. Click the group name (for example, **Default**).
3. Click **App Servers** in the left tree menu.
4. Click the name of the app server in the left tree menu.
5. Click **Request Blackout** under the name of the app server. The **Request Blackout Periods** page appears.
6. Click the **Create** tab. The **Add Request Blackout Periods** page appears.
7. Fill in the form as needed for the blackout period you want to create. Clicking the radio buttons will bring up more forms to complete.
8. Click **OK** to create the blackout period.

The new blackout period will take effect immediately.

11.2.2. Deleting Request Blackouts

To delete a request blackout period, follow these steps:

1. Click **Groups** in the left tree menu.
2. Click the group name (for example, **Default**).
3. Click **App Servers** in the left tree menu.
4. Click the name of the app server in the left tree menu.
5. Click **Request Blackouts** under the App Server. The **Request Blackout Periods** page appears.
6. In the area corresponding to the blackout period you want to delete, click **Delete**.
7. Click **OK** on the confirmation page to delete the blackout period.

The blackout period is deleted immediately.

11.3. Storing and Monitoring the Last User Login Attempt

MarkLogic Server provides the ability to store the outcome of the last attempt a user made at logging in. This section describes this feature and how to use it.

11.3.1. Storing Last User Login Information in a Last-Login Database

A database named `Last-Login` is created upon installation of (or upgrade from 3.2 to) MarkLogic Server. You can use this database as the last-login database for one or more App Servers. Each time a successful or unsuccessful login is made via the App Server, the last-login database is updated with that information. Only information for the last login attempt is retained. Because this database is constantly changing with each login attempt (every request is authenticated, so each request updates the last-login database), it is a good idea to use a different database than content database for your last-login database. In general, it is probably OK to keep a single last-login database that is shared by all App Servers who use this functionality, but if you do this, keep in mind that the information will then be shared by all the App Servers; that is, that the last-login time and other statistics will be for all App Servers using the last-login database.



NOTE

A history of the successful login attempts is not retained; only the time of the last successful login is stored in the database.

11.3.2. Configuring User Login Monitoring

To set up user login monitoring for a given App Server, follow these steps:

1. Click **Groups** in the left tree menu.
2. Click the group name (for example, **Default**).
3. Click **App Servers** in the left tree menu.
4. Click the name of the app server in the left tree menu.
5. Use the **Last Login** drop-down to select a database. The `last-login` database is created for this purpose, but you can select any database that you want. If no last-login database is selected, then the last-login feature is disabled.
6. Optionally, in the **Display Last Login** field, select **true**.
7. Click **OK** to save the changes.

11.3.3. Displaying the Last Login Information

Each App Server configuration page has a `display last login` setting. The value of this setting is returned as part of the XML output of the `xdmp:user-last-login` API. You can use this information as logic in your application to determine whether to display some last-login information to the application.

The Admin Interface uses the `display last login` setting to show information about its last login attempt. When a last-login database is configured and the `display last login` setting is true, then something similar to the following is displayed at the bottom of each page of the Admin Interface:

```
last successful login: September 2, 2008 7:54:16 PM
                        last unsuccessful login: none
unsuccessful login attempts since last login: 0
```

12. Databases

This section describes how to use the Admin Interface to create and configure databases. For details on how to create and configure databases programmatically, see [Creating and Configuring Forests and Databases](#) in the *Scripting Administrative Tasks Guide*.

Later sections in this guide introduce some concepts for tuning the performance of your databases. For information on database backup and restore operations, see [Section 19, “Backing Up and Restoring a Database” \[160\]](#).

12.1. Understanding Databases

A *database* in MarkLogic Server serves as a layer of abstraction between forests and HTTP, WebDAV, or XDBC servers. A database is made up of data *forests* that are configured on hosts within the same cluster but not necessarily in the same group. It enables a set of one or more forests to appear as a single contiguous set of content for query purposes. See [Section 22.1, “Understanding Forests” \[202\]](#) for more detail on forests.

Multiple HTTP, XDBC, and WebDAV servers can be connected to the same database, allowing different applications to be deployed over a common content base. A database can also span forests that are configured on multiple hosts enabling data scalability through hardware expansion. To ensure database consistency, all forests that are attached to a database must be available in order for the database to be available.



WARNING

All system master databases — Security, Schemas, Triggers, Modules, Extensions, Last-Login and App-Services — MUST be single forest databases. For high availability, you do need to configure one or two replica forests for each system database. But there is no benefit to having multiple master forests in the database.

12.1.1. Schemas and Security Databases

The installation process creates the following *auxiliary* databases by default - *Documents*, *Last-Login*, *Schemas*, *Security*, *Modules*, and *Triggers*. Every database points to a security database and a schema database. Security configuration information is stored in the security database and schemas are stored in the schemas database. A database can point back to itself for the security and schemas databases, storing the security information and schemas in the same repository as the documents. However, security objects created through the Admin Interface are stored in the *Security* database by default. MarkLogic recommends leaving databases connected to *Security* as their security database.

12.1.2. Modules Database

The *modules* database is an *auxiliary* database that is used to store executable XQuery, JavaScript, and REST code. During installation, a database named *Modules* is created, but any database can be used as a modules database, as long as the HTTP or XDBC server is configured to use it as a modules database. Also, it is possible to use the same database to store executable modules, to store queryable documents, and/or to store triggers.

If you use a modules database, each executable document in the database must have the root (specified in the HTTP or XDBC server) as a prefix to its URI. Also, if you want to access the documents in the database via WebDAV, then it should have `automatic` directory creation enabled,

because `automatic` directory creation is required for WebDAV. For information about directories and roots, see [Section 8.3, “Directories” \[55\]](#) and [Section 8.4, “Server Root Directory” \[56\]](#).

For example, if you are using a modules database and specify a root in an HTTP or XDBC server of `http://marklogic.com/`, the following documents are executable from that server:

```
http://marklogic.com/default.xqy
http://marklogic.com/myXQueryFiles/search_db.xqy
```

but the following files are not executable (because they do not have URIs that start with the root):

```
http://mycompany.com/default.xqy
/myXQueryFiles/search_db.xqy
```

In order to execute any documents in a modules database, the documents must be loaded with `execute` permissions. You can do this either by loading the documents as a user with default privileges that include `execute` permissions, or by setting those permissions on the document after it loads. For information on using permissions, privileges, and other security features in MarkLogic Server, see [Section 23, “Security Administration” \[212\]](#) and the sections related to security in the [Application Developer’s Guide](#).

12.1.3. Triggers Database

The *triggers* database is an auxiliary database that is used to store triggers. During installation, a database named *Triggers* is created, but any database can be used as a triggers database. Also, it is possible to use the same database to store executable modules, to store queryable documents, and/or to store triggers. A triggers database is required if you are using the Content Processing Framework. For details on the Content Processing Framework, see [Content Processing Framework Guide](#).



NOTE

To avoid issues with `mlcp` when triggers are configured, it is recommended that you do not set a database to be its own Triggers Database.

12.1.4. Database Settings

Each database has settings that control various aspects of a database such as memory allocation, indexing options, and so on. You configure these settings in the Admin Interface.

Basic Administrative Settings


The administrative settings configure properties such as the database name and which security and schema databases a database uses. These settings take effect immediately after any changes are made in the Admin Interface.

Database Setting	Description
<code>database name</code>	The name of the database.
<code>security database</code>	The name of the security database which this database accesses.
<code>schema database</code>	The name of the schemas database which this database accesses.
<code>triggers database</code>	The name of the triggers database which this database accesses.
<code>data encryption</code>	Enable or disable encryption at rest for this database. For details, see Encryption at Rest in <i>Securing MarkLogic Server</i> .
<code>encryption key id</code>	Data encryption key ID. For details, see Encryption at Rest in <i>Securing MarkLogic Server</i> .

Index Settings That Affect Documents

When you change any index settings for a database, the new settings take effect based on whether reindexing is enabled (`reindexer.enable` set to `true`). For more details on text indexes, see [Text Indexing](#).

In general, adding index options slows document loading and increases the size of database files.

Database Setting	Description
language	Specifies the default language for content in this database. Any content without an <code>xml:lang</code> attribute will be indexed in the language specified here. You should have a license key if you specify a non-English language; if you specify a non-english language and do not have a license for that language, the stemming and tokenization will be generic.
stemmed searches	Controls the level of stemming applied to word searches. Stemmed searches match not only the exact word in the search, but also words that come from the same stem and mean the same thing (for example, a search for <code>be</code> will also match the term <code>is</code>). For more details on stemmed searches, see Understanding and Using Stemmed Searches in the <i>Search Developer's Guide</i> .
word searches	Whether or not to enable unstemmed word searches. Enables searches for exact matches of words.
word positions	Index word positions for faster phrase and <code>cts:near-query</code> searches.
fast phrase searches	Speeds up phrase searches by eliminating some false positive results.
fast reverse searches	Speeds up reverse query searches by indexing saved queries.
triple index	Enables the RDF triple index to support SPARQL execution over RDF triples. When this parameter is true, <code>sem:sparql()</code> can be used, but document loading is slower and the database files are larger. <div style="border: 1px solid gray; padding: 10px; margin-top: 10px;">  <p>NOTE This feature requires a valid semantics license key.</p> </div>
triple positions	Specifies whether to index positional data to speed up the performance of proximity queries that use <code>cts:triple-range-query()</code> .
fast case sensitive searches	Speeds up case sensitive searches by eliminating some false positive results.
fast diacritic sensitive searches	Speeds up diacritic-sensitive searches by eliminating some false positive results.
fast element word searches	Speeds up element-word searches by eliminating some false positive results.
element word positions	Index element word positions for faster element-based phrase and <code>cts:near-query</code> searches.
fast element phrase searches	Speeds up element phrase searches by eliminating some false positive results.
element value positions	Index element word positions for faster element-based phrase and <code>cts:near-query</code> searches that use <code>cts:element-value-query()</code> .
attribute value positions	Index attribute word positions for faster attribute-based phrase and <code>cts:near-query</code> searches that use <code>cts:element-value-query()</code> and faster <code>cts:element-query</code> searches that use a <code>cts:element-attribute-*-query</code> .
field value searches	Enables searches that use <code>cts:field-value-query</code> .
field value positions	Enables positions for searches that use <code>cts:field-value-query</code> .
three character searches	Enables wildcard searches where the search pattern contains three or more consecutive non-wildcard characters (for example, <code>abc*x</code> , <code>*abc</code> , <code>a?bcd</code>). When combined with a codepoint word lexicon, speeds the performance of any wildcard search (including searches with fewer than three consecutive non-wildcard characters). MarkLogic recommends combining the <code>three character search index</code> with a codepoint collation word lexicon. For more details about wildcard searches, see Understanding and Using Wildcard Searches in the <i>Search Developer's Guide</i> .

Database Setting	Description
three character word positions	Index word positions for three-character wildcard queries.
fast element character searches	Enables wildcard searches and speeds up element-based wildcard searches. For more details about wildcard searches, see Understanding and Using Wildcard Searches in the <i>Search Developer's Guide</i> .
trailing wildcard searches	Faster wildcard searches with the wildcard at the end of the search pattern (for example, abc*). For more details about wildcard searches, see Understanding and Using Wildcard Searches in the <i>Search Developer's Guide</i> .
trailing wildcard word positions	Index word positions for trailing wildcard searches.
fast element trailing wildcard searches	Faster wildcard searches with the wildcard at the end of the search pattern within a specific element, but slower document loads and larger database files.
word lexicon	Maintains a lexicon of all of the words in a database, with uniqueness determined by a specified collation. Additionally, works in combination with the <code>three character search index</code> to speed wildcard searches. For more details about wildcard searches, see Understanding and Using Wildcard Searches in the <i>Search Developer's Guide</i> .
two character searches	Enables wildcard searches where the search pattern contains two or more consecutive non-wildcard characters (for example, ab*). This index is not needed if you have <code>three character searches</code> and a word lexicon. For more details about wildcard searches, see Understanding and Using Wildcard Searches in the <i>Search Developer's Guide</i> .
one character searches	Enables wildcard searches where the search pattern contains a single non-wildcard characters (for example, a*). This index is not needed if you have <code>three character searches</code> and a word lexicon. For more details about wildcard searches, see Understanding and Using Wildcard Searches in the <i>Search Developer's Guide</i> .
uri lexicon	Maintains a lexicon of all of the URIs used in a database. The URI lexicon speeds up queries that constrain on URIs. It is like a range index of all of the URIs in the database. To access values from the URI lexicon, use the <code>cts:uris</code> or <code>cts:uri-match</code> APIs.
collection lexicon	Maintains a lexicon of all of the collection URIs used in a database. The collection lexicon speeds up queries that constrain on collections. It is like a range index of all of the collection URIs in the database. To access values from the collection lexicon, use the <code>cts:collections</code> or <code>cts:collection-match</code> APIs.

Rebalancer Settings

You can enable the database rebalancer to automatically distribute content evenly across forests in a database. The specifics of database rebalancing are described in [Section 16, "Database Rebalancing"](#) [118].

Database Setting	Description
assignment policy	Specifies how documents are to be distributed across the database forests. Both the rebalancing process and the document load/insert process follow this policy. For details on the document assignment policies, see Section 16.3, "Rebalancer Document Assignment Policies" [119].
rebalancer enable	When set to <code>true</code> , the database rebalancer will automatically redistribute the content across the database forests. When set to <code>false</code> , rebalancing is disabled.
rebalancer throttle	Sets the priority of system resources devoted to rebalancing. Higher numbers give rebalancing a higher priority.

Reindexing Settings

The reindexing settings enable or disable reindexing and allow you to force reindexing of older fragments.

Database Setting	Description
reindexer enable	When set to <code>true</code> , index configuration changes automatically initiate a background reindexing operation on the entire database. When set to <code>false</code> , any new index settings take effect for future documents loaded into the database; existing documents retain the old settings until they are reloaded or until you set <code>reindexer enable</code> to <code>true</code> . For information on how the reindexer effects queries, see Section 24.3, "Query Behavior with Reindex Settings Enabled and Disabled" [234].

Database Setting	Description
<code>reindexer throttle</code>	Sets the priority of system resources devoted to reindexing. Higher numbers give reindexing a higher priority.
<code>reindexer timestamp</code>	Specifies the timestamp of fragments to force a reindex/refragment operation. Click the <code>get current timestamp</code> button to enter the current system timestamp. When you set this parameter to a timestamp and <code>reindex enable</code> is set to <code>true</code> , it causes a reindex and refragment operation on all fragments in the database that have a timestamp equal to or less than the specified timestamp. Note that if you restore a database that has a timestamp set, if there are fragments in the restored content that are older than the specified content, they will start to reindex as soon as they are restored.

Document and Directory Settings

The document and directory settings affect the default settings for how documents and directories are created in the database.

Database Setting	Description
<code>directory creation</code>	<p>Specifies if directories should be automatically created when a document is created. If you are using the database to store documents accessible via a WebDAV server or as a Modules database, this setting should be set to <code>automatic</code>. The following are the settings:</p> <ul style="list-style-type: none"> <code>automatic</code>—directories are automatically created based on the URI of a document. <code>manual-enforced</code>—requires that the directory hierarchy corresponding to the URI exists before creating a document. If you create a document where the corresponding directory hierarchy does not exist, an error is raised. For example, if you try to create a document with the URI <code>http://marklogic.com/file.xml</code>, then the directory with URI <code>http://marklogic.com/</code> must exist. Otherwise, an error is raised. This setting provides the same behavior as a file system. <code>manual</code>—directories are not automatically created, but documents can still be created without corresponding directories. <p>For more information about directories, see Section 8.3, “Directories” [55]. For more information about Modules databases, see Section 12.1.2, “Modules Database” [81].</p>
<code>maintain last modified</code>	Creates and updates the last-modified property each time a document is created or updated. The default is <code>false</code> .
<code>maintain directory last modified</code>	Creates and updates the last-modified property on a directory each time a directory is created or updated. If set to <code>true</code> , update operations on documents in a directory will also update the directory last-modified timestamp, which can cause some contention when multiple documents in the directory are being updated. If your application is experiencing contention during these type of updates (for example, if you see deadlock-detected messages in the error log), set this property to <code>false</code> . The default is <code>false</code> .
<code>inherit permissions</code>	When set to <code>true</code> , documents and directories automatically inherit permissions from their parent directory (if permissions are not set explicitly when creating the document or directory). If there are any default permissions on the user who is creating the document or directory, those permissions are combined with any inherited permissions.
<code>inherit collections</code>	When set to <code>true</code> , documents and directories automatically inherit collection settings from their parent directory (if collections are not set explicitly when creating the document or directory). If there are any default collections on the user who is creating the document or directory, those permissions are combined with any inherited collections.
<code>inherit quality</code>	When set to <code>true</code> , documents and directories automatically inherit any quality settings from their parent directory (if quality is not set explicitly when creating the document or directory).

Memory and Journal Settings

The memory and journal settings are automatically configured at installation time. The memory settings configure the memory limits for the system, and the journal settings control the transactional journal, used for recovery if a database transaction fails. The default settings should be sufficient for most systems. Depending on the system workload, setting the memory settings incorrectly can adversely affect performance; if you need to change the settings and you have an active maintenance contract, you can contact MarkLogic Support for help.

This table describes the memory and journal settings:

Database Setting	Description
<code>in memory limit</code>	The maximum number of fragments in an in-memory stand. An in-memory stand contains the latest version of any new or changed fragments. Periodically, in-memory stands are written to disk as a new stand in the forest. Also, if a stand accumulates a number of fragments beyond this limit, it is automatically saved to disk by a background thread.
<code>in memory list size</code>	The size, in megabytes, of the in-memory list storage.
<code>in memory tree size</code>	The size, in megabytes, of the in-memory tree storage. The <code>in memory tree size</code> should be at least 1 or 2 megabytes larger than the largest text or small binary document you plan on loading into the database. The largest small binary file size is always constrained by the "large size threshold" database configuration setting.
<code>in memory range index size</code>	The size, in megabytes, of the in-memory range index storage.
<code>in memory reverse index size</code>	The size, in megabytes, of the in-memory reverse index storage.
<code>in memory triple index size</code>	The size, in megabytes, of the in-memory triple index storage.
<code>large size threshold</code>	Determines the size, in kilobytes, beyond which large binary documents are stored in the Large Data Directory instead of directly in a stand. Binaries smaller than or equal to the threshold are considered small binary files and stored in stands. Binaries larger the threshold are considered large binary files and stored in the Large Data Directory.
<code>locking</code>	Specifies how robust transaction locking should be. When set to <code>strict</code> , locking enforces mutual exclusion on existing documents and on new documents. When set to <code>fast</code> , locking enforces mutual exclusion on existing and new documents. Instead of locking all the forests on new documents, it uses a hash function to select one forest to lock. In general, this is faster than <code>strict</code> . However, for a short period of time after a new forest is added, some of the transactions need to be retried internally. When set to <code>off</code> , locking does not enforce mutual exclusion on existing documents or on new documents; only use this setting if you are sure all documents you are loading are new (a new bulk load, for example), otherwise you might create duplicate URIs in the database.
<code>journaling</code>	Specifies how robust transaction journaling should be. When set to <code>strict</code> , the journal protects against MarkLogic Server process failures, host operating system kernel failures, and host hardware failures. When set to <code>fast</code> , the journal protects against MarkLogic Server process failures but not against host operating system kernel failures or host hardware failures. When set to <code>off</code> , the journal does not protect against MarkLogic Server process failures, host operating system kernel failures, or host hardware failures.
<code>journal size</code>	<p>The size, in megabytes, of each journal file. The system uses journal files for recovery operations if a transaction fails to complete successfully. The default value should be sufficient for most systems; it is calculated at database configuration time based on the size of your system. If you change the other memory settings, however, the journal size should equal the sum of the <code>in memory list size</code> and the <code>in memory tree size</code>. Additionally, you should add space to the journal size if you use range indexes (particularly if you use a lot of range indexes or have extremely large range indexes), as range index data can take up journal space. Also, if your transactions span multiple forests, you may also need to add journal size, as each journal must keep the lock information for all of the documents in the transaction, not just for the documents that reside in the forest in which the journal exists.</p> <p>When you change the journal size, the next time the system creates a new journal, it will use the new size limit; existing journals will continue to use the old size limit until they are replaced with new ones (for example, when a journal fills up, when a forest is cleared, or when the system is cleanly shutdown and restarted).</p>
<code>preallocate journals</code>	This setting has no effect.
<code>preload mapped data</code>	Specifies whether memory mapped data (for example, range indexes and word lexicons) is loaded into memory when a forest is mounted to the database. Preloading the memory mapped data improves query performance, but uses more memory, especially if you have a lot of range indexes and/or lexicons. Also, it will cause a lot of disk I/O at database startup time, slowing the system performance during the time the mapped data is read into memory. If you do not preload the mapped data, it will be paged into memory dynamically when a query requests data that needs it, slowing the query response time.
<code>range index optimize</code>	Specifies how range indexes are to be optimized. When set to <code>facet-time</code> , range indexes are optimized to minimize the amount of CPU time used. When set to <code>memory-size</code> , range indexes are optimized to minimize the amount of memory used.

Other Settings

This table lists the remaining database configuration options:

Database Setting	Description
<code>position list max size</code>	The maximum size, in megabytes, of the position list portion of the index for a given term. If the position list size for a given term grows larger than the limit specified, then the position information for that term is discarded. The default value is 128, the minimum value is 1, and the maximum value is 512. For example, position queries (<code>cts:near-query</code>) for frequently occurring words that have reached this limit (words like <i>a</i> , <i>an</i> , <i>the</i> , and so on) are resolved without using the indexes. Even though those types of words are resolved without using the indexes, this limit helps improve performance by making the indexes smaller and more efficient in relation to the content actually loaded in the database.
<code>format compatibility</code>	Specifies the version compatibility that MarkLogic Server applies to the indexes for this database during request evaluation. Setting this to a value other than <code>automatic</code> specifies that all forest data has the specified on-disk format, and it disables the automatic checking for index compatibility information. The automatic detection occurs during database startup and after any database configuration changes, and can take some time and system resources for very large forests and for very large clusters. The default value of <code>automatic</code> is recommended for most installations.
<code>index detection</code>	Specifies whether to auto-detect index compatibility between the content and the current database settings. This detection occurs during database startup and after any database configuration changes, and can take some time and system resources for very large forests and for very large clusters. Setting this to <code>none</code> also causes queries to use the current database index settings, even if some settings have not completed reindexing. The default value of <code>automatic</code> is recommended for most installations.
<code>expunge locks</code>	Specifies if MarkLogic Server will automatically expunge any lock fragments created using <code>xdmp:lock-acquire</code> with specified timeouts. If you set this to <code>automatic</code> , the lock fragments will be cleaned up as they expire. With The default setting of <code>none</code> , the locks will remain in the database after the locks expire (although they will no longer be locking any documents) until they are explicitly removed with <code>xdmp:lock-release</code> .
<code>tf normalization</code>	Specifies whether to use the default term-frequency normalization (<code>scaled-log</code>), which scales the term frequency based on the size of the document, or to use the <code>unscaled-log</code> , which uses term frequency as a function of the actual term frequency in a document, regardless of the document size, or to choose an intermediate level of scaling with lower impact than the default document size-based scaling.

Merge Control Settings

The merge control settings allow you to control when merges occur, set merge parameters, and set up blackout periods where you do not want merges to occur. You can access the merge control settings by clicking the Admin Interface menu item for **Database > db_name > Merge Controls**. Use caution when adjusting the merge parameters or using merge blackouts, as merges are necessary for optimal database performance. For explanations of the merge control settings and more details on controlling merges, see [Section 15, “Understanding and Controlling Database Merges” \[108\]](#).

12.1.5. Example of Databases in MarkLogic Server

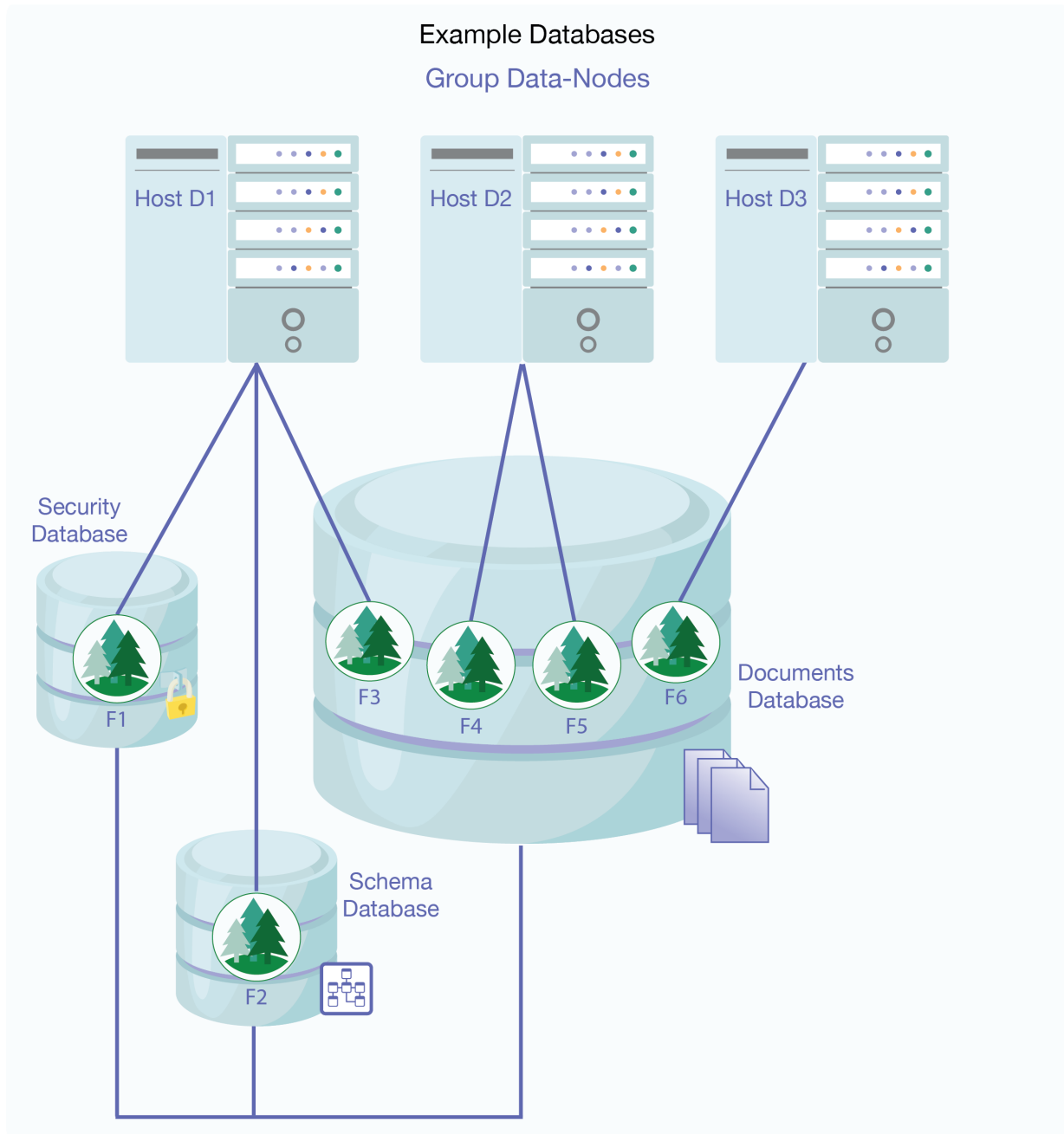
This section provides an example which demonstrates the concept of a database and the relationships between a database, a host, and a forest in MarkLogic Server.

In the diagram below, Hosts D1, D2, and D3 belong to the Data-Nodes Group.

D1 is the first host in Data-Nodes Group on which MarkLogic Server is loaded. Three databases are created by default: Security, Schemas, and Documents. In the diagram below, 3 forests, F1, F2, and F3, are configured on Host D1 and assigned to the Security, Schemas, and Documents databases respectively.

D2 is the second host to join the Data-Nodes group. Forests F4 and F5 are configured on D2 and attached to the Documents database.

D3 is the third host to join the Data-Nodes group and has forest F6 configured on it. F6 is also assigned to the Documents database.



12.2. Creating a New Database

To create a new database, follow these steps:

1. Click **Databases** in the left tree menu
2. Click the **Create** tab. The **Database** page appears
3. Enter the **Database Name**. This is the name that the system will use to refer to this database.
4. Select a security database to associate with this database. We recommend selecting **Security** as the security database.
5. Select a security database to associate with this database.
6. You may leave the rest of the parameters unchanged or set them according to your needs.
7. Click **OK**.

Your database is now created. You can now attach forests to the database. Creating a database is a “hot” admin task.

12.3. Attaching and/or Detaching Forests to/from a Database

To query content in a forest, the forest must be attached to a database. Forests can be moved from one database to another (detached from one database and attached to another). Detaching a forest from a database does not delete the forest; the forest remains on the host on which it was created with the data intact. Forests can be moved from one database to another (detached from one and attached to another). However, before you attach the forest to another database, ensure that the new database has the same configuration as the old database. If the configuration of the new database is different and the `reindex enable` setting is set to `true` on the new database, the forest will begin reindexing to match the database configuration as soon as it is attached.



NOTE

If you attach a new forest to a database that makes use of the journal archiving feature described in [Section 19.2, “Backing Up Databases with Journal Archiving” \[166\]](#), the forest will not participate in journal archiving until the next time the database is backed up. For details on how to do an immediate backup of a database, see [Section 19.4.1, “Backing Up a Database Immediately” \[169\]](#).

You can also attach and detach forests from databases using the **Forest Summary** page, as described in [Section 22.5, “Attaching and Detaching Forests Using the Forest Summary Page” \[206\]](#).

In the Admin Interface, follow these steps to attach or detach one or more forests to a database:

1. Click **Databases** in the left tree menu
2. Under **Databases**, click the name of the database.
3. Click **Forests**.
4. On the **Configure Forests In A Database** screen, check the box corresponding to the forest(s) you want to attach to the database. You can also uncheck forests you want to detach from the database.
5. Click **OK**.

The forests you attached or detached are now reflected in the database configuration. Attaching and detaching a forest to a database are “hot” admin tasks.

12.4. Viewing Database Settings

To view the settings for a particular database, follow these steps:

1. Click the **Databases** icon on the left tree menu.
2. Locate the database for which you want to view settings, either in the tree menu or in the **Database Summary** table.
3. Click the name of the database for which you want to view the settings.
4. View the settings.
5. Click **Forests**, **Triggers**, **Content Processing**, **Fragment Roots**, **Fragment Parents**, **Element-Word-Query-Throughs**, **Phrase-Throughs**, **Phrase-Arounds**, **Element Indexes** and **Attribute Indexes** to view settings specific to those aspects of the database.

12.5. Loading Documents into a Database

You can use the Admin Interface to load documents into the database. The documents will be loaded with the default permissions and added to the default collections of the user with which you logged into the Admin Interface.

To load a set of documents into a database, follow these steps:

1. Click **Databases** in the left tree menu
2. Click on the database into which you want to load the documents.
3. Click on the **Load** tab near the top right.
4. Enter the name of the directory in which the documents are located. This directory must be accessible by the host from which the Admin Interface is currently running.
5. Enter a filter for the names of the documents to be loaded (for example, `*.xml` to load all files with an xml extension). For an exact match, enter the full name of the document.
6. Click **Next** to proceed. The load confirmation screen lists all documents in the specified directory matching the specified filter.
7. Click **OK** to complete the load.

The documents are loaded into the database. The URI path of the documents are the same as your filesystem path.

12.6. Merging a Database

You can merge all of the forest data in the database using the Admin Interface. As described in [Section 15, “Understanding and Controlling Database Merges” \[108\]](#), merging the forests in a database improves performance and is periodically done automatically in the background by MarkLogic Server. The **Merge** button allows you to explicitly merge the forest data for this database.

To explicitly merge the database, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click the name of the database you want to merge.
4. Click the **Merge** button.
A confirmation message appears.
5. Confirm that you want to merge the forest data in this database and click **OK**.

Merging data in a database is a “hot” admin task; the changes take effect immediately.

12.7. Reindexing a Database

You can reindex all of the document data in the database using the Admin Interface. As described in [Section 24, “Text Indexing” \[227\]](#), text indexing accelerates the performance of a certain queries and is periodically done automatically in the background by MarkLogic Server. The reindex operation sets the [reindexer timestamp \[85\]](#) to the current system timestamp, which causes a reindex and refragment operation on all fragments in the database that have a timestamp equal to or less than the timestamp (assuming [reindexer enable \[84\]](#) is set to true). The Reindex button forces a complete reindex/refragment operation on the database.

To reindex the database, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click the **Reindex** button on the **Database** page.
A confirmation message displays.
4. Confirm that you want to reindex this database and click **OK**.

Reindexing data in a database is a “hot” admin task; the changes take effect immediately.

12.8. Clearing a Database

You can clear all of the forest content from the database using the Admin Interface. Clearing a database deletes all of the content from all of the forests in the database, but leaves the database configuration intact.

To clear all data from a database, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click the **Clear** button.
A confirmation message displays.
4. Confirm that you want to clear the forest data from this database and click **OK**.

Clearing a database is a “hot” admin task; the changes take effect immediately.

12.9. Disabling a Database

You can disable a database using the Admin Interface. You can either disable only the database or the database along with all of its forests. Disabling only the database marks the database as disabled and unmounts all the forests from the database. However, the database forests remain enabled. Disabling the database and its forests marks the database and each forest as disabled, unmounts all the forests from the database, and clears all memory caches for all the forests in the database. The database remains unavailable for any query operations while it is disabled.

Disabling a database does not delete the configuration or document data. The database and forest can later be re-enabled by clicking Enable.

To disable a database, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click the **Disable** button on the **Database** page.
A confirmation message appears.
4. Click either **Disable Database** to disable only the database, or **Disable Database and Forests** to disable the database and its forests.

12.10. Deleting a Database

A database cannot be deleted if there are any HTTP, WebDAV, or XDBC servers that refer to the database. Deleting a database detaches the forests that are attached to it, but does not delete them. The forests remain on the hosts on which they were created with the data intact.

To delete a database, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click **Delete** near the top right.
4. Assuming that there are not any HTTP, WebDAV, or XDBC servers referring to the database, a delete confirmation screen appears. Click **OK**.
5. If you want to delete the forests used by the database, follow the procedure described in [Section 22.12, “Deleting a Forest from a Host” \[210\]](#) for each forest.



NOTE

Clicking the **Clear** button clears all of the forests attached to this database, removing all of the data from the forests. Clicking the **Delete** button removes the database configuration, but does not delete the data stored in the forests.

The database is now permanently deleted. Deleting a database is a “hot” admin task.

12.11. Checking and Setting Permissions for a Document in a Database

You can use the Admin Interface to check the permissions of a document or directory in a database. You can also use the `xdmp:document-get-permissions` and `xdmp:document-set-permissions` APIs to get and set permissions. For details on document permissions, see [Securing MarkLogic Server](#).

In the Admin Interface, follow these steps to check and/or set permissions on a document or directory in a database:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Under the database, click **Permissions**.
4. Enter the URI of the document or directory and click **OK**.
5. If you want to change the permissions, choose a role and capability from the drop-down lists. If you want to add more permissions, click the **More Permissions** button.
6. To commit your changes, click **OK**. To cancel the action, press **Cancel**.

13. Word Query Database Settings

This section describes how to configure a database to include or exclude elements, add index settings, and perform other configuration changes for `cts:word-query` operations.

13.1. Understanding the Word Query Configuration

Basic search of words and phrases in MarkLogic Server is based on the query constructor `cts:word-query`. You can control the behavior of these basic searches by changing the database configuration for word query. You can exclude and/or include elements from word queries, and you can add extra indexing options compared to the options configured in the database configuration. This section describes the options available in the word query configuration.

13.1.1. Overview of Configuration Options

The following lists the main options you can set in the word query configuration to control how word queries are resolved in a database:

- By default, all elements are included in the word query configuration and the indexing options are the same as the database indexing options.
- All word query configurations are set on a per-database basis.
- The word query configuration controls the behavior of the `cts:word-query`, `cts:words`, and `cts:word-match` APIs. This includes controlling the words that get indexed, as well as controlling the words that are returned from the filter (evaluator) portion of query evaluation.
- Word query inherits the database index settings as a starting point for its index settings.
- You cannot turn off indexing options that are enabled in the database settings.
- If you check index options in word query that are enabled in the database, it will not change any behavior. However, if you subsequently disable a database index setting that is checked in the word query settings, it will remain for the word query.
- You can include and/or exclude named elements from word queries.
- For any element you include, you can optionally constrain it by a value for a specified attribute.
- For any element you include, you can optionally specify a weight. The weight is used when determining relevance scores, where a weight greater than 1.0 will boost scores and a weight lower than 1.0 will lower scores for matches within the element.

13.1.2. Understanding Which Elements are Included and Excluded

You can include and/or exclude elements from word queries. This is useful if you know you will never want to search some element content. This section describes how MarkLogic Server determines what content is included in word queries and what is not when you include and/or exclude elements from the word query configuration.



NOTE

If you want to be able to search on everything in a word query, but also want a special view of the content that includes and/or excludes some elements, consider creating a field instead of modifying the word query configuration. For details on fields, see [Section 14, “Fields Database Settings” \[98\]](#).

By default, all element content (all text node children of elements) is included in word queries. If you decide to include and/or exclude any elements from word queries, there are rules that govern which non-specified elements are indexed and which are not. The rules are based on inheriting the include

state from the parent element. For example, if the parent element is marked as an included element (and is therefore indexed and evaluated for word query), then its children, if they do not appear on the exclude list, are also included.



NOTE

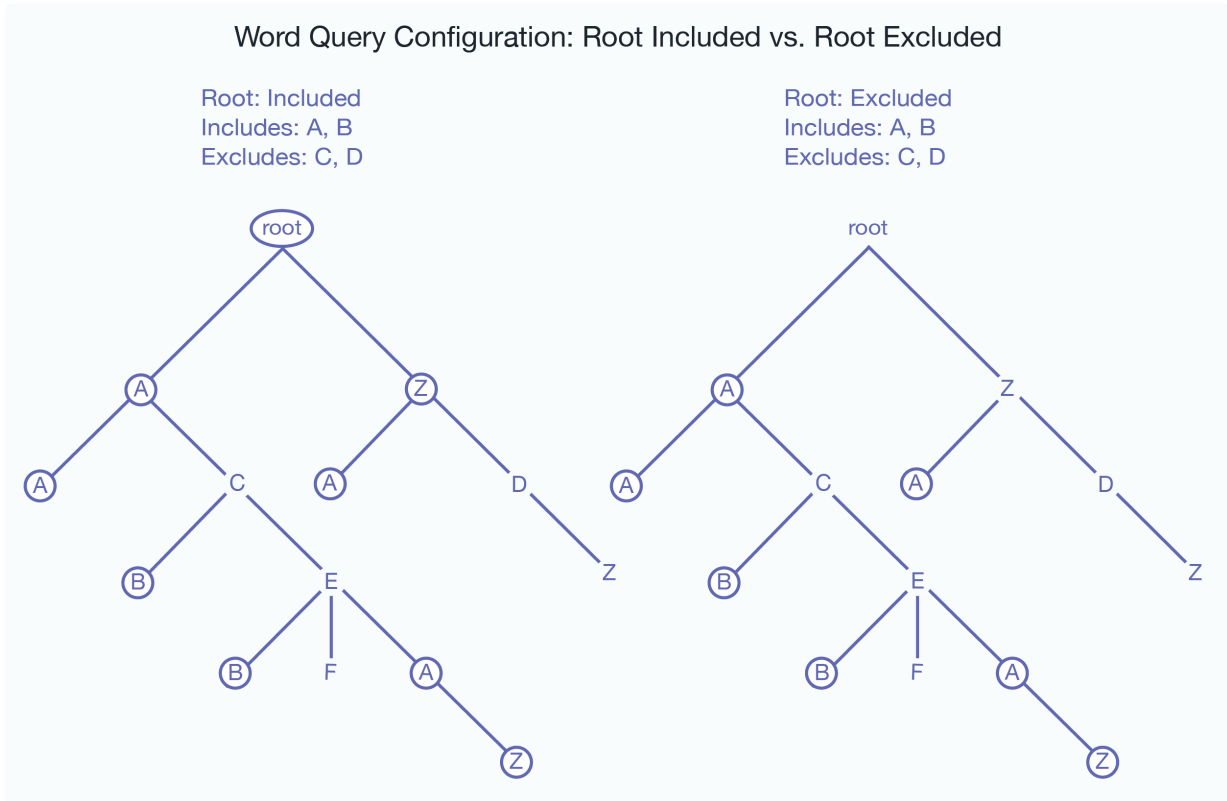
If you configure word query exclusions then MarkLogic may not use word positions, even if it is enabled. For example, MarkLogic will not use word positions for resolution of queries such as `cts:element-word-query` or `cts.jsonPropertyWordQuery` resolution in positional contexts such as a near query. This can lead to false positives. You can use `xdmp:plan` or `cts.plan` to determine whether word positions are being used.

When MarkLogic Server determines which elements to include/exclude, it walks the XML tree using the following rules:

1. Start at the root node of the document.
2. If the root node is included (either because it is explicitly included or because `include document root` is set to true), MarkLogic Server includes the immediate text node children of the document root element and then moves to its element children. If the root node is excluded, the text nodes are not included and MarkLogic Server moves down the XML tree to its element children.
3. If the parent element (the root element in this case) was included, MarkLogic Server keeps walking down the tree and including the text node children until it encounters an explicitly excluded element.
4. If the parent element (the root element in this case) was not included, MarkLogic Server keeps walking down the tree, not including the text node children, until it encounters an explicitly included element.
5. MarkLogic Server keeps walking down the tree, including or not according to the state inherited from the parent element, until it encounters the next included element (if it is in the *not included* state) or excluded element (if it is in the *included* state).
6. During this process, when an element is encountered that is neither included nor excluded, it inherits the included state (*not included* or *included*) from the parent element.
7. MarkLogic Server keeps walking down the XML tree using this logic to determine its included state, until it reaches the end of the document.

The only way to guarantee an element's text node children will be included (assuming you have any elements included and/or excluded) is to add it to the included list, and the only way to guarantee an element is not included is to add it to the excluded list.

The following figure shows what is included for two configurations, one with the root node included and one with the root node excluded. Note that the includes and excludes are the same. The lines below the element names represent the text nodes, and the yes/no indicates whether the content in the text nodes is included in word queries. The `root` represents the root node of an XML structure, with elements `A` and `B` included and elements `C` and `D` excluded. Elements that are not explicitly included or excluded (for example, `E`, `F`, and `Z`) inherit from their parents.



Notice that the `z` node, which is not explicitly included or excluded, sometimes is included and sometimes is not included, depending on the include state of its parent element.

13.1.3. Adding a Weight to Boost or Lower the Relevance of an Included Element

When you include an element, one of the options is to add a *weight* to the included element specification. When you add a weight, all text in this element (including any text in all text node descendants of the element) are weighted by the specified value, changing the relevance at query time. Specifying a weight greater than 1.0 will boost scores and a weight lower than 1.0 will lower scores for matches within the element.

When you specify a weight, the term frequency for any tokens in that element (including tokens in descendant text nodes) is multiplied by that number. This happens during document load, update, or reindexing. For example, if you specify a weight of 2.0, each term will have a term frequency of 2.0, making it as if each term appeared twice (for score calculation purposes). Similarly, if you specify a weight of 0.5, each term will have a term frequency of 0.5.



NOTE

Because the weight boosting affects term frequency, it will only affect relevance orders for scoring algorithms that include term frequency (for example, `logtf/idf` or `logtf`); scoring algorithms that do not consider weight will not be affected by these weights (for example, `score-simple`).

Adding a weight is useful to boost or lower scores on searches where the match occurs in a given element. For example, if you want matches in `TITLE` elements to contribute more towards the relevancy score than matches in other elements, you can specify a weight of 2.0 for the `TITLE`

element. Conversely, if you want matches in `TITLE` elements to contribute less to the relevancy score than matches in other elements, you can specify a weight of `0.5` for the `TITLE` element. For details on how relevance is calculated, see [Composing cts:query Expressions](#) in the *Search Developer's Guide*.

13.1.4. Specifying an Attribute Value for an Included Element

When you include an element, one of the options is to specify an attribute value. This option allows you to only include elements with a particular attribute/value pair. The attribute/value pair acts as a predicate on which to constrain the content. For example, consider the following XML snippet:

```
<chapter class="history">some text here</chapter>
<chapter class="mathematics">some more text here</chapter>
<chapter class="english">some other text here</chapter>
<chapter class="history">some different text here</chapter>
<chapter class="french">other text here</chapter>
<chapter class="linguistics">still other text here</chapter>
```

For the element `chapter`, if you specify the attribute/value pair of `class` and `history`, then only the following elements will be included:

```
<chapter class="history">some text here</chapter>
<chapter class="history">some different text here</chapter>
```

You can only specify an attribute value for an included element; you cannot specify one for an excluded element.

13.1.5. Understanding the Index Option Configuration

The word query configuration allows you to add some extra indexing options from the ones that are currently set in the database configuration. Adding any index options to the word query configuration does not add those options to the element-based index options.

To add a particular index option to word query, you check the box corresponding to the index option. Adding any index options that are not enabled in the database configuration will cause new and updated documents to use the new indexing for word query, and will trigger a reindex operation if `reindex enable` is set to `true` in the database configuration.

Options that are enabled in the database configuration appear in bold on the word query configuration. If you check the box next to an option with bold-face type, it does not change your configuration. However, if you subsequently disable that index option in the database configuration, it will remain enabled for word query as long as the box is checked.

13.2. Configuring Customized Word Query Settings

This section provides the procedure for customizing the word query settings. For details on what the meaning of the various configuration options in fields, see [Section 13.1, "Understanding the Word Query Configuration"](#) [93]. The following is the procedure for modifying the word query configuration for your database:



NOTE

When you modify the word query settings, those modifications apply to all queries that use the `cts:word-query` constructor, which is the default constructor for `cts:search`. If you want to be able to search on everything in a word query, but also want a special view of the content that includes and/or excludes some elements, consider creating a field instead of modifying the word query configuration. For details on fields, see [Section 14, "Fields Database Settings"](#) [98].

Use the Admin Interface to add a new field configuration to a database:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Under the target database, click **Word Query**.
4. in the **Include Root** field, leave the default setting of **true** selected If you want the word query to include the root element of the document,



NOTE

If you select **false** you will need to include elements in the word query configuration in order to get any results from word queries. Typically, you would leave this set to true and choose some elements to explicitly exclude and some to explicitly include (optionally adding a scoring weight and/or an attribute value constraint).

5. Under **Index Settings**, check any index settings to include in the word query. Index settings shown in bold indicate the setting is inherited from the database setting. For details, see [Section 13.1.5, "Understanding the Index Option Configuration" \[96\]](#).
6. Click **OK** to save any changes .
7. If you want to exclude any elements from word queries, click the **Excludes** tab.
8. Enter the namespace URI (if needed) and the local name for the excluded element.
9. Click **OK**.
10. Repeat [Step 7 - Step 9](#) for each element you want to exclude.
11. Click the **Includes** tab to specify elements to include in the word query.
12. On the **Included Element** page, specify a local name for the element to include. If the element is in a namespace, specify the namespace URI for the element to include.
13. [OPTIONAL] If you want to boost or lower the relevance contribution for matches within this element, in the **Weight** field, specify a weight other than the default of 1.0. Weights greater than 1.0 will boost the relevance contribution and weights lower than 1.0 will lower the contribution.
14. [OPTIONAL] If you want to only include elements that have an attribute with a specified value, complete the **Attribute Namespace Uri** (if needed), the **Attribute Localname**, and **Attribute Value** fields. Then only elements containing attributes with the specified value will be included. You must specify the exact value; no wildcard characters are used.
15. When you have specified everything for this element, click **OK**.
16. Repeat [Step 11 - Step 15](#) for each element you want to include.
17. You can delete any included or excluded fields from the tables at the bottom of the configuration page.

14. Fields Database Settings

This section describes how to use the Admin Interface to create and configure fields in the database settings. Fields are used with the `cts:field-word-query`, `cts:field-words`, and `cts:field-word-match` APIs, as well as with the field lexicon APIs, and allow you to define a named field consisting of several elements over which you can search. For details on how to create and configure fields programmatically, see [Adding a Database Field and Included Element](#) in the *Scripting Administrative Tasks Guide*. For details on lexicons on fields, see [Browsing With Lexicons](#) in the *Search Developer's Guide*.

14.1. Overview of Fields

Fields provide a convenient mechanism for querying a portion of the database based on XML element QNames or JSON property names. Unlike collections or directories, which enable you to query portions of a database based on document URIs, fields enable you to query portions of a database based on XML element and JSON property names. This offers extra convenience for the application developer, and also offers a performance boost over other methods of querying a portion of the database. Fields are extremely useful when you have content in one or more elements or JSON properties that you want to query simply and efficiently as a single unit.

Field query is similar to word query (in its default configuration, with everything included), but instead of querying everything in the database, fields query only what is configured for the specified field. Fields have their own set of indexes, independent of the database indexes. Because fields have their own indexes, and a field is typically a small subset of the whole database, querying a field is often more efficient than querying those same XML element or JSON properties directly (with `cts:word-query`, for example).

Also, because fields have their own sets of indexes, relevance for fields is calculated based on the content in the field, not based on all of the content in the database. This provides finer-grain relevance for field searches than for other searches.

You can use fields to create portions of the content that you might want to query as a single unit. Additionally, you can configure a field with indexing options over and above the ones configured in the database. For example, consider a database containing many technical articles, each article containing a brief abstract. You might want to build an application that allows greater capabilities for searching through the abstracts than for searching through the rest of the articles. Assume your main content does not have wildcard indexes, but you want to be able to search through the abstracts using wildcard searches. You can create a field on the abstract, and then add wildcard indexes to that field. Because the field represents only a relatively small percentage of the content, the relative cost of the extra indexing is small.

Indexing of JSON and XML content differs slightly. This introduces differences in the behavior of field value queries and field range queries over the two types of content. For details, see [How Field Queries Differ Between JSON and XML](#) in the *Application Developer's Guide*.

14.2. Understanding Field Configurations

Field search of words and phrases in MarkLogic Server is based on the query constructor `cts:field-word-query`. You can control the behavior of these field searches by changing the database configuration for the field you query. You can exclude and/or include elements from path and root fields, and you can add extra indexing options for some elements. This section describes the options available in the configuration.

14.2.1. Overview of Field Configuration Options

The following lists the main options you can set in the field query configuration to control how queries against the specified field are resolved:

- By default, no XML elements or JSON properties are included in the field query configuration and the indexing options are the same as the database indexing options. You must specify at least one element or property to include for the field to include anything.
- All field configurations are set on a per-database basis.
- The field configuration controls the behavior of the `cts:field-word-query`, `cts:field-value-query`, `cts:field-range-query`, `cts:field-words`, and `cts:field-word-match` APIs. This includes controlling the terms that get indexed as well as controlling the terms that are returned from the filter (evaluator) portion of query evaluation.
- Fields inherit the database index settings as a starting point for its index settings.
- You can add extra index options for each field. These added index options will not affect other queries (for example, `cts:word-query`, `cts:element-word-query`, `cts:element-attribute-word-query`, `cts:json-property-word-query`).
- If you check index options in a field that are enabled in the database, it will not change any behavior. However, if you subsequently disable a database index setting that is checked in the field setting, it will remain for the field.
- You can include and/or exclude named XML elements or JSON properties from path and root fields.
- For any XML element you include, you can optionally constrain it by a value for a specified XML element attribute.
- For any XML element or JSON property you include in a path or root field, you can optionally specify a weight. The weight is used when determining relevance scores, where a weight greater than 1.0 will boost scores and a weight lower than 1.0 will lower scores for matches within the element or property.
- Each field has its own set of indexes; it does not share the indexes with the word query indexes. Therefore, if you have a field with fewer elements than word query, there is a smaller amount of content to index and fewer I/O operations are needed to resolve the query from the indexes (index resolution phase of query processing).

There are three types of fields:

- [Root Fields \[99\]](#)
- [Path Fields \[99\]](#)
- [Section 14.2.3, “Metadata Fields” \[104\]](#)

Root and Path fields are described in [Section 14.2.2, “Root and Path Fields” \[99\]](#). Metadata fields are described in [Section 14.2.3, “Metadata Fields” \[104\]](#).

14.2.2. Root and Path Fields

You can include and/or exclude elements from a root or path field. This is useful if you know you will never want to search some element content. This section describes how MarkLogic Server determines what content is included in the field and what is not when you include and/or exclude elements from the field configuration.

This section describes the options available in the configuration.

Root Fields

Root fields include and/or exclude document elements regardless of their relative positions in the document. In a root field, you can choose whether or not to include and exclude elements starting at the document root. By default, no element content (all text node children of elements) is included in a field.

Path Fields

In a path field, the included and excluded elements are constrained to the sub-tree identified by the path. For example, if the path for the field is `/A/B/C`, only elements in node `C`, such as `A/B/C/D`, `A/B/C/D/E` and `/A/B/C/Z`, are included or excluded from the field.

A path field may include one or more paths. Multiple paths are treated as the union of the paths. Consequently, each of them will identify a root of a field-instance in a given document.

If a path includes namespace prefixes on some elements, the namespaces must be defined in the same manner used for path range indexes, as described in [Section 25.10, “Defining Namespace Prefixes Used in Path Range Indexes and Fields” \[244\]](#).

If a path for a field ends in a single node or an attribute, the include/exclude definitions are meaningless.

Each path is given a weight, which is used to boost or lower the relevance of text that is contributed by the path.

How Field Settings Determine What Is Included and Excluded

Once you define a path or root field, you can select which document elements are included and excluded. When MarkLogic Server determines which elements to include/exclude, it walks the XML tree using the following rules (note that these are the same rules used for including/excluding elements in the word query configuration):

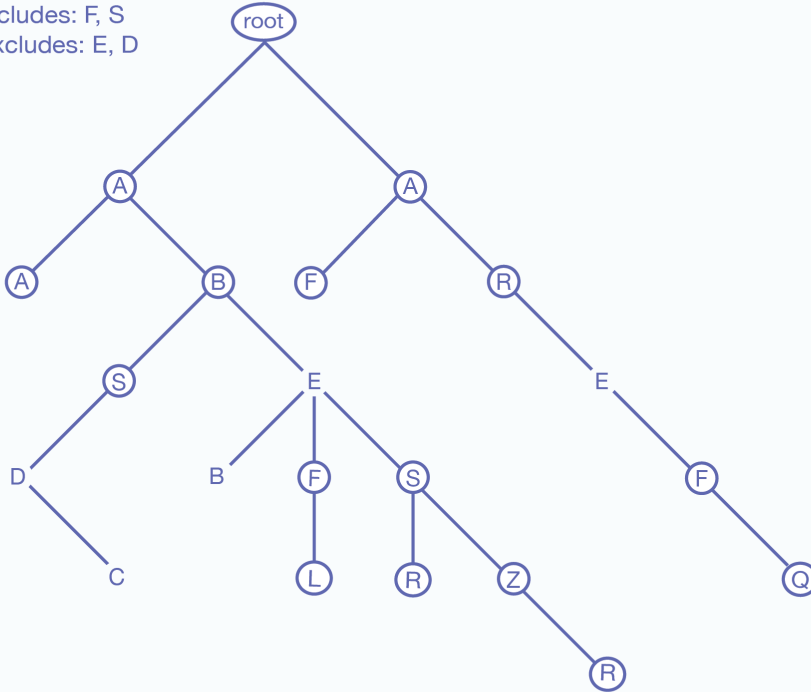
1. Start at the root node of the document.
2. If the field type is path, the explicitly included and excluded elements are constrained to the sub-tree identified by the path. All other elements are excluded.
3. If the field type is root, and if the root element is included (either because it is explicitly included or because `include document root` is set to true), MarkLogic Server includes the immediate text node children of the document root element and then moves to its element children. If the root element is excluded, the text nodes are not included and MarkLogic Server moves down the XML tree to its element children.
4. If the parent element was included, MarkLogic Server keeps walking down the tree and including the text node children until it encounters an explicitly excluded element.
5. If the parent element was not included, MarkLogic Server keeps walking down the tree, not including the text node children, until it encounters an explicitly included element.
6. During this process, when an element is encountered that is neither included nor excluded, it inherits the included state (*not included* or *included*) from the parent element. MarkLogic Server keeps walking down the tree, including or not according to the state inherited from the parent element, until it encounters the next included element (if its parent is *not included*) or excluded element (if its parent is *included*).
7. MarkLogic Server keeps walking down the XML tree using this logic to determine each element's included state, until it reaches the end of the document.

The only way to guarantee an element's text node children will be included (assuming you have any elements included and/or excluded) is to add it to the included list, and the only way to guarantee an element is not included is to add it to the excluded list.

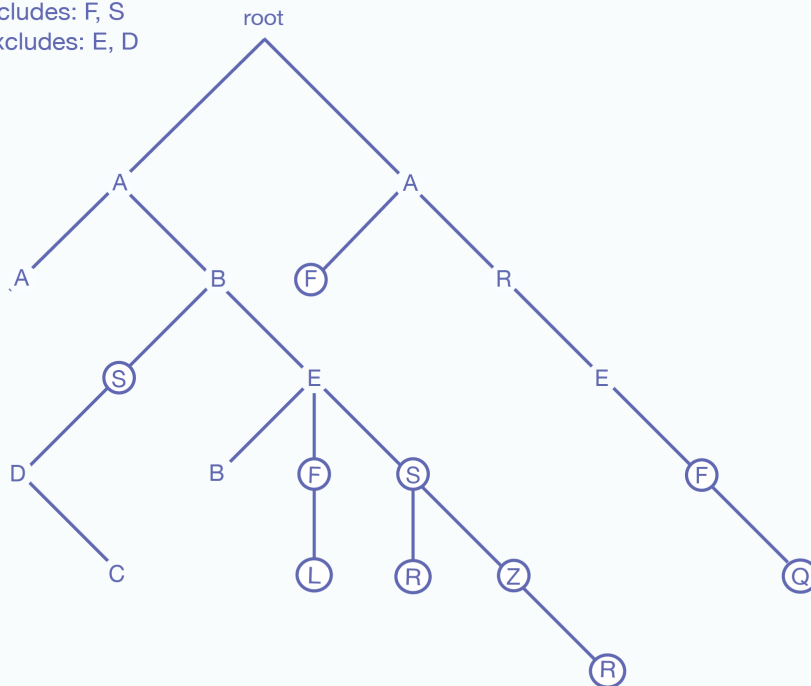
The following figure shows what is included for two possible root field configurations, one with the root node included and one with the root node excluded. Note that the includes and excludes are the same. The lines below the element names represent the text nodes, and the boxed red letters indicates that the content in the text node is included in word queries. The `root` represents the root node of an XML structure, with elements `F` and `S` included and elements `E` and `D` excluded. Elements that are not explicitly included or excluded (for example, `A`, `B`, and `C`) inherit from their parents.

Root Field Configuration: Root Included vs. Root Excluded

Root: Included
Includes: F, S
Excludes: E, D



Root: Excluded
Includes: F, S
Excludes: E, D

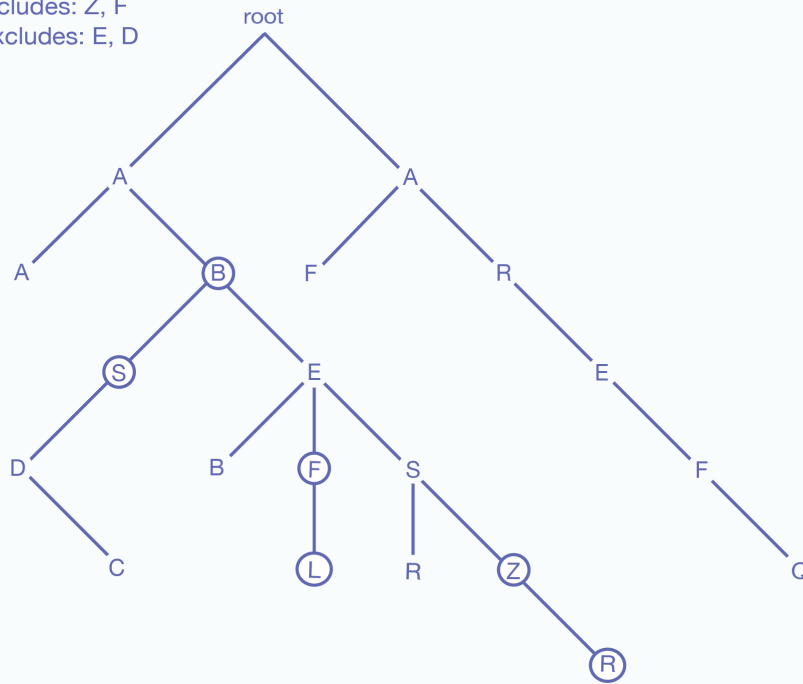


Notice that the A, B, and R nodes, which are not explicitly included or excluded, sometimes are included and sometimes are not included, depending on the include state of their parent element.

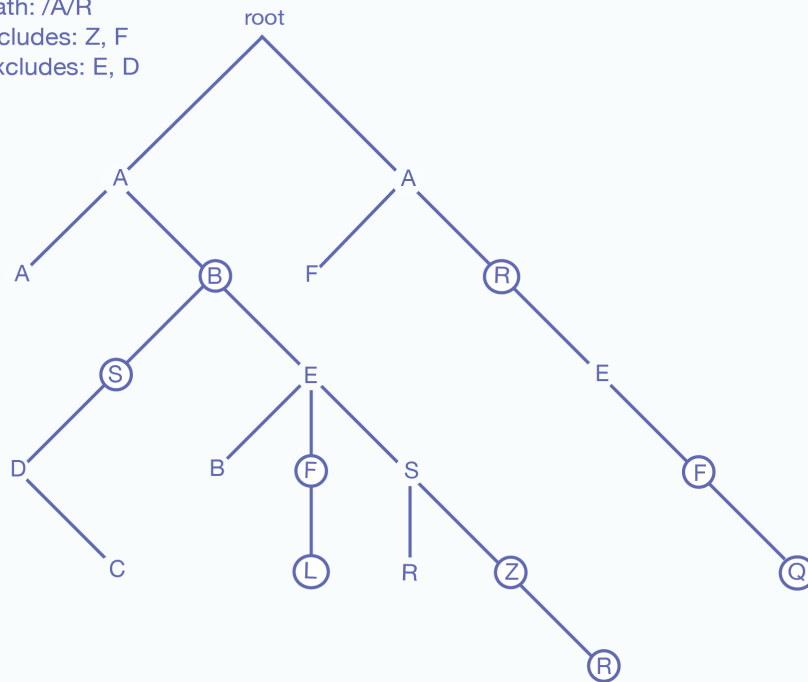
The following figure shows what is included for two possible path field configurations, one with a single path and the other with two paths. As with the previous figure for root field configurations, the includes and excludes are the same:

Field Query Configuration: One Path vs. Two Paths

Path: /A/B
Includes: Z, F
Excludes: E, D



Path: /A/B
Path: /A/R
Includes: Z, F
Excludes: E, D



Adding a Weight to Boost or Lower the Relevance of an Included Element or Property

When you include an XML element or JSON property, one of the options is to add a `weight` to the included element or property specification. When you add a weight, all text in this element (including any text in all text node descendants of the element) are weighted by the specified value, changing the

relevance at query time. Specifying a weight greater than 1.0 will boost scores and a weight lower than 1.0 will lower scores for matches within the element.

When you specify a weight, the term frequency for any tokens in that element (including tokens in descendant text nodes) is multiplied by that number. This happens during document load, update, or reindexing. For example, if you specify a weight of 2.0, each term will have a term frequency of 2.0, making it as if each term appeared twice (for score calculation purposes). Similarly, if you specify a weight of 0.5, each term will have a term frequency of 0.5.



NOTE

Because the weight boosting affects term frequency, it will only affect relevance orders for scoring algorithms that include term frequency (for example, `logtf/idf` or `logtf`); scoring algorithms that do not consider weight will not be affected by these weights (for example, `score-simple`).

Adding a weight is useful to boost or lower scores on searches where the match occurs in a given element. For example, if you want matches in `TITLE` elements to contribute more towards the relevancy score than matches in other elements, you can specify a weight of 2.0 for the `TITLE` element. Conversely, if you want matches in `TITLE` elements to contribute less to the relevancy score than matches in other elements, you can specify a weight of 0.5 for the `TITLE` element. For details on how relevance is calculated, see [Composing cts:query Expressions](#) in the *Search Developer's Guide*.

If a field has two or more elements with different weights and, if one of those elements is a child of another element, then the weight of the parent element is used and the weight of the child element is ignored. For example, you have a field, named `test`, that includes elements `A` and `B`. `A` is given a weight of 10 and `B` is given a weight of 2. The returned results of a search query that includes `cts:field-value-query("test", ("Foo"), "unfiltered")` will be computed based on a weight of 10 for the following document:

```
<A>
  <B>Foo</B>
</A>
```

Specifying an Attribute Value for an Included or Excluded Element

When you include an element, one of the options is to specify an attribute value. This option allows you to only include or exclude elements with a particular attribute/value pair. The attribute/value pair acts as a predicate on which to constrain the content. For example, consider the following XML snippet:

```
<chapter class="history">some text here</chapter>
<chapter class="mathematics">some more text here</chapter>
<chapter class="english">some other text here</chapter>
<chapter class="history">some different text here</chapter>
<chapter class="french">other text here</chapter>
<chapter class="linguistics">still other text here</chapter>
```

For the element `chapter`, if you specify the attribute/value pair of `class` and `history`, then only the following elements will be included:

```
<chapter class="history">some text here</chapter>
<chapter class="history">some different text here</chapter>
```

Similarly, you can specify an attribute value for an excluded element when you configure an excluded element.

14.2.3. Metadata Fields

Metadata fields are used by temporal documents to store valid and system timestamps and archival information, as described in the *Temporal Developer's Guide*. You can also use this capability to associate user-defined key-value metadata with non-temporal documents. Metadata fields are sometimes referred to as just “metadata” or as “key-value metadata”.

Metadata fields differ from root and path fields in that they do not define elements to be included or excluded from search. Instead, metadata fields define key/value combinations that are associated with a document, but stored outside of that document.

To search this type of metadata, you must explicitly create a field based on the metadata key you want to be able to search. For details on configuring a metadata field, see [Section 14.4.2, “Configuring a New Metadata Field” \[106\]](#).

Metadata fields can be operated on using any API function that takes a field. For example, you can do all of the following operations on a metadata field:

- Query using a `cts:field-word-query` and `cts:field-value-query` function.
- Create a word lexicon on a metadata field and use it in a `cts:field-words` and `cts:field-word-match` function.
- Create a range index on a metadata field and use it in a `cts:field-range-query`, `cts:field-values`, `cts:field-value-match`, and `cts:field-value-ranges` function.
- Make a range index reference for a metadata field range index and use it in a `cts:values`, `cts:value-match`, `cts:value-ranges`, `cts:value-co-occurrences`, `cts:value-tuples` and `cts:ordering` function.
- Configure tokenizer-overrides.
- Configure stemmed-searches.
- Configure word-searches.
- Configure field-value-searches.
- Configure fast-phrase-searches.
- Configure fast-case-sensitive-searches.
- Configure fast-diacritic-sensitive-searches.
- Configure trailing-wildcard-searches.
- Configure three-character-searches.
- Configure two-character-searches.
- Configure one-character-searches.

Metadata for temporal documents is managed by the temporal APIs, as described in [Managing Temporal Documents](#) in the *Temporal Developer's Guide*. For non-temporal documents, metadata can be inserted along with the document by the `xdmp.documentInsert` or `xdmp.documentLoad` function. You can add or modify document metadata using the `xdmp.documentPutMetadata` and `xdmp.documentSetMetadata` functions. Document metadata can be returned using the `xdmp.documentGetMetadata` and `xdmp.documentGetMetadataValue` functions.

Metadata can also be associated with a document node. Node metadata is managed by means of the `xdmp.nodeMetadata` and `xdmp.nodeMetadataValue` functions.

14.2.4. Understanding the Index Option Configuration

The field configuration allows you to add some extra indexing options from the ones that are currently set in the database configuration. Adding any index options to the field configuration does not add those options to the element-based index options at the database level.

To add or remove a particular index option to a field, you check or uncheck the box corresponding to the index option. Adding any index options that are not enabled in the database configuration will cause new and updated documents to use the new indexing for the field, and will trigger a reindex operation if `reindex enable` is set to true in the database configuration.

Options that are enabled in the database configuration appear in bold in the field configuration. The field settings in the database configuration and the database field configuration are ORed together. For example, if you uncheck the box next to an option with bold-face type in the field configuration, it does not change the equivalent option in the database configuration. To disable a field setting for the database, both the database and field configurations for that option must be consistent.

14.3. Field Word Lexicons and Field Value Lexicons

As with word lexicons, you can create a word lexicon for each field. A *field word lexicon* is a list of all of the unique words in the database that occur in the field. The list is ordered in the specified collation. You can create multiple field lexicons on the same field with different collations. The field word lexicons are accessed with the `cts:field-words` and `cts:field-word-match` APIs.

As with element or attribute lexicons, you can create a value lexicon on a field. A *field value lexicon* is a list of all of the unique values in the database that occur in the field. To create a field value lexicon, define a field range index.

For more details about lexicons, see [Browsing With Lexicons](#) in the *Search Developer's Guide*.

14.4. Configuring Fields

This section provides procedures to create and modify field configurations in a database. For details on what the meaning of the various configuration options in fields, see [Section 14.2, "Understanding Field Configurations"](#) [98].

14.4.1. Configuring a New Path or Root Field

Use the Admin Interface to add a new field configuration to a database:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Under the database in which you want to create the field, click the **Fields** link.
4. Click the **Create** tab. The **Create Field in Database** page appears.
5. Enter a name for the field.
6. By default, the field type is **paths**. If creating a path field, enter the path expression. If you want to boost or lower the relevance contribution for matches within this path, specify a weight other than the default of 1.0. Weights greater than 1.0 will boost the relevance contribution and weights lower than 1.0 will lower the contribution. If you are defining multiple paths, click **more field paths**



NOTE

In most cases, a path field will give you everything you need. However, to create a root field, skip to [Step 9](#).

7. [OPTIONAL] Create any Field Range Indexes or Tokenizer overrides. You can also go back and add these later.
8. If you want the field to include any extra index options from the database, or if you want to remove some index options from the field, check or uncheck the appropriate settings under Index Settings. Index settings shown in bold indicate the setting is inherited from the database setting. You can uncheck an inherited index setting to not inherit the setting from the database-level configuration. For details, see [Section 14.2.4, "Understanding the Index Option Configuration"](#) [104].
9. Alternately, if creating a root field, set **Field Type** to **root**. Note that in most cases, a path field will give you everything you need, and you are not likely to need to create a root field.
10. In the **Include Root** field, if you want the root field to include the root element of the document, even if it is not explicitly included, click `true` to include document root. Typically, you leave this set to the default of `false`, unless your field will include most of the elements in the database.
11. Scroll to the bottom of the screen and click **OK**. An **Included Elements** and **Excluded Elements** section are added to the bottom of the screen.

12. If you want to add a word lexicon for the field, enter the collation URI next in the add text box. The URI for the UCA Default Collation, <http://marklogic.com/collation/>, is useful for many applications. For details on collations, see [Language Support in MarkLogic Server](#) in the *Search Developer's Guide*. Click the **OK** button to add the field word lexicon (if you want to create one). If you want to create other field word lexicons with different collations, repeat this step specifying a different collation URI for the new lexicon.
13. Click the **Includes** tab to specify elements to include in the field.
14. On the **Included Element** page, specify a local name for the element to include. If the element is in a namespace, specify the namespace URI for the element to include.
15. [OPTIONAL] If you want to boost or lower the relevance contribution for matches within this element, specify a weight other than the default of 1.0. Weights greater than 1.0 will boost the relevance contribution and weights lower than 1.0 will lower the contribution.
16. [OPTIONAL] If you want to only include elements that have an attribute with a specified value, enter the attribute namespace URI (if needed), the attribute local name, and a value for the attribute. Then only elements containing attributes with the specified value will be included. You must specify the exact value; no wildcard characters are used.
17. When you have specified everything for this element, click **OK**.
18. Repeat [Step 13 - Step 17](#) for each element you want to include.
19. If you want to exclude any elements from the field, click the **Excludes** tab.
20. Enter the namespace URI (if needed) and the local name for the excluded element.
21. [OPTIONAL] If you want to only exclude elements that have an attribute with a specified value, enter the attribute namespace URI (if needed), the attribute local name, and a value for the attribute. Then only elements containing attributes with the specified value will be excluded. You must specify the exact value; no wildcard characters are used.
22. Click **OK**.
23. Repeat [Step 19 - Step 22](#) for each element you want to exclude.
24. You can delete any included or excluded fields from the tables at the bottom of the field configuration page by clicking **Delete**.

14.4.2. Configuring a New Metadata Field

Use the Admin Interface to add a new metadata field configuration to a database:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Under the database in which you want to create the field, click **Fields**.
4. Click the **Create** tab. The **Create Field in Database** page appears.
5. Enter a name for the field.
6. For field type, select **metadata**.
7. [OPTIONAL] Create any Field Range Indexes. You can also go back and add these later.
8. If you want to add a word lexicon for the field, enter the collation URI next in the add text box. The URI for the UCA Default Collation, <http://marklogic.com/collation/>, is useful for many applications. For details on collations, see the [Language Support in MarkLogic Server](#) in the *Search Developer's Guide*. Click **OK** to add the field word lexicon (if you want to create one). If you want to create other field word lexicons with different collations, repeat this step specifying a different collation URI for the new lexicon.
9. [OPTIONAL] Create any Field Range Indexes or Tokenizer overrides. You can also go back and add these later.
10. If you want the field to include any extra index options from the database, or if you want to remove some index options from the field, check or uncheck items in the **Index Settings** section. Index settings shown in bold indicate the setting is inherited from the database setting. You can uncheck an inherited index setting to not inherit the setting from the database-level configuration. For details, see [Section 14.2.4, "Understanding the Index Option Configuration" \[104\]](#).

**NOTE**

The field value positions, trailing wildcard word positions, and three character word positions options can be set, but they will have no effect on queries.

14.4.3. Modifying an Existing Field

To modify an existing field, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click the **Fields** link in the left tree menu. The **Summary** tab appears.
4. Click the name of the field you want to edit. The **Configure** tab opens.
5. If you want to change any of the settings, make any desired modifications and click **OK**.
6. The remainder of the procedure is the same as in [Configuring a New Path or Root Field](#), starting with [Step 12 \[106\]](#) to create a field word lexicon, and, in the case of path and root fields, continuing on to add or delete included and excluded elements.

14.4.4. Creating a Range Index on a Field

You can create a range index on a field for faster searches on the field data. You must first create a field before creating a range index on the field. The usual trade-offs between query speed and ingestion speed and server resources described in [Section 25.1, “Understanding Range Indexes” \[237\]](#) apply to field range index.

To create a range index on a field, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click **Field Range Indexes** in the left tree menu.
4. Click the **Add** tab.
5. Select a value from the **Scalar Type** field.
6. Enter the name of an existing field.
7. In the **Range Value Positions** field, select **true** if you want the index to store position data. (Metadata fields are positionless, so position settings have no impact.)
8. In the **Invalid Values** list, select **reject** to prevent the ingestion of documents with fields that do not match the type specified for the range index. Select **ignore** to allow the ingestion of non-matching documents.
9. Click **OK**.

The index is created. If the `reindexer enable` setting is `true` for that database, then reindexing will begin immediately. The new index is not available for use in range and lexicon queries until the reindexing operation is complete.

15. Understanding and Controlling Database Merges

This section describes database merges and how you can control them.

15.1. Overview of Merges: Merges are Good

This section provides an overview of merges.

15.1.1. Dynamic and Self-Tuning

Merges are a way of self-tuning the performance of the system, and MarkLogic Server continuously assesses the state of each database to see if it would benefit from self-tuning through a merge. In most cases, the default merge settings and the dynamic nature of merges will keep the database tuned optimally at all times. Because merges can be resource intensive (both disk I/O and CPU), however, some DBAs might need to control when merges occur and/or when they do not occur. You can do that by setting your merge policy as appropriate for your environment, as described in [Section 15.2, “Setting Merge Policy” \[109\]](#).

Dynamic and self-tuning, merges are a “good thing”; they not only reclaim disk space, but improve the query and search performance of the system. Databases are made up of one or more forests, and forests are made up of one or more *stands*. The more stands there are in a forest, the more time it takes to resolve a query. Merges reduce the number of stands in each forest in a database, thereby improving the time it takes to resolve queries.

15.1.2. What Happens During a Merge

A database consists of one or more forests, and each forest consists of one or more stands. Each stand consists of one or more fragments. When a document is updated, new versions of all of the fragments associated with the document update are created in a new stand. Any old versions of the fragment remain in the old stand with a system timestamp that lets MarkLogic Server know that they are old versions of the fragments. Similarly, when a document is deleted, its fragments remain in the old stand with a system timestamp that lets MarkLogic Server know that they are old versions of the fragments.

Merges occur to move any unchanged fragments from an old stand into a new stand, deleting any old versions of fragments (including deleted fragments), thereby freeing up disk space and compacting the usable fragments so they are all together on disk. Additionally, merges combine index data for all of the fragments in a stand, thereby optimizing the indexes. Merges are a normal part of database operation, and they ensure that the system continues to perform at its best as updates and deletes occur.

To summarize, as part of merging, the following occurs:

- Multiple stands are combined into one for improved performance.
- Disk space is reclaimed.
- Indexes and lexicons are combined and re-optimized based on their new size.

The result is a database that is smaller and can resolve queries much faster than before the merge.

15.1.3. Dangers of Disabling Merges

MarkLogic Server is designed to periodically merge. It is dangerous to leave merges disabled on a database when there are any updates occurring to the system. While disabling merges might eliminate some contention for resources during periods where merges and other requests are simultaneously occurring on the system, the performance of MarkLogic Server will degrade over time if merges are not allowed to proceed when changes (inserts, updates, deletes) are made to the database.

Furthermore, disabling or eliminating merging may eventually lead to a condition in which the server is unable to make changes to the database. For example, when an in-memory stand fills up, it is written to an on-disk stand. MarkLogic Server has a fixed limit for the maximum number of stands (64), and

eventually, that limit will occur and you will no longer be able to update your system. Therefore, there is no control available to disable merges. If you feel you need to disable merges and you have an active maintenance contract, you can contact MarkLogic Technical Support for help.

In most cases where merges are causing disruptions to your system, you should be able to adjust the merge policy parameters to settings that will work in your environment. If you feel you need to disable merges and you have an active maintenance contract, you can contact MarkLogic Technical Support for help. Monitor the system and make sure the number of stands per forest does not grow too high. For details on setting merge controls, see [Section 15.2.2, “Configuring the Merge Policy” \[110\]](#) and [Section 15.8, “Configuring Merge Policy Rules” \[114\]](#).

In some cases, especially in environments with many forests and constantly changing content across many of the forests, an alternative to disabling merges is to set one or more forests to be delete-only. For details, see [Section 22.3, “Making a Forest Delete-Only” \[204\]](#).

15.1.4. Merges Will Change Scores

When a database merges, it deletes old fragments that exist in the database, therefore changing (making it smaller) the total number of fragments in the database. Because the number of fragments in the database is used in determining the score for a `cts:search` operation, merges will have an impact on search scores, which in turn might impact the order of search results (which are ordered by relevance score).

The amount of impact that merges have on scores is dependent on how many old versions of fragments there are waiting to be merged, the content of the old fragments, and the overall size of the database. For large databases with relatively little amount of change, the difference in the scores will be very small. For smaller databases with large amount of change, the differences in scores can be significant before and after a merge completes.

15.2. Setting Merge Policy

This section describes the tools you can use to control merges.

In some cases, especially in environments with many forests and constantly changing content across many of the forests, another tool for setting merge policy is to set one or more forests to be delete-only (`updates allowed` set to `false`). For details, see [Section 22.3, “Making a Forest Delete-Only” \[204\]](#).

15.2.1. Overview of the Merge Policy Controls

If you determine that you need to manage your merges, there are several types of controls to help you manage the conditions in which merges occur:

The following controls determine the conditions under which MarkLogic Server deems a merge is desirable:

- `merge min size`
- `merge min ratio`

The following controls determine the conditions under which a merge will be allowed:

- `merge max size`
- `merge blackout periods`

The following control determines if multiple versions of fragments are preserved when a merge is performed:

- `merge timestamp`

The following controls explicitly initiate a merge (see [Manually Initiating a Merge](#)):

- `xdmp:merge()`
- The **Merge** button in the Admin Interface.

The Admin Interface has controls for canceling a merge (see [Canceling a Merge](#)).

For more information on how set up your system to better control merges, see [Configuring Merge Policy Rules](#).

15.2.2. Configuring the Merge Policy

The merge policy determines when automatic merges occur on a database, as well as other administrative functions.

To configure the merge policy, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click **Merge Policy** in the left hand menu. The **Merge Policy** page appears.
4. Enter the appropriate values and click **OK**. The following table describes the available settings:

Database Setting	Description
Merge Priority	<p>Specifies the CPU scheduler priority at which merges should run. The settings are:</p> <ul style="list-style-type: none"> • <code>normal</code> specifies the same CPU scheduler priority as for requests. • <code>lower</code> specifies a lower CPU scheduler priority than for requests. <p>Merges always run with normal priority on forests with more than 16 stands.</p>
Merge Max Size	<p>The maximum size, in megabytes, of a stand that will result from a merge. If a stand grows beyond the specified size, it will not be merged. If two stands would be larger than the specified size if merged, they will not be merged together. If you set this to smaller sizes, large merges (which may require more disk and CPU resources) will be prevented. The default is 48 GB (49152 MB), which is recommended because it provides a good balance between keeping the number of stands low and preventing very large merges from using large amounts of disk space. Set this to 0 to allow any sized stand to merge. Use care when setting this to a non-zero value lower than the default value, as this can prevent merges which are ultimately required for the system to maintain performance levels and to allow optimized updates to the system.</p>
Merge Min Size	<p>The minimum number of fragments that a stand can contain. Two or more stands with fewer than this number of fragments are automatically merged.</p>
Merge Min Ratio	<p>A positive integer indicating the minimum ratio between the number of fragments in a stand and the number of fragments in all of the other smaller stands (that is stands with fewer fragments) in the forest. Stands with a fragment count below this ratio relative to all smaller stands are automatically merged with the smaller stands. For an example, see Section 15.8.2, "If You Want to Reduce the Number of "Large" Merges" [115].</p>
Merge Timestamp	<p>The timestamp stored on merged stands. This is used for point-in-time queries, and determines when space occupied by deleted fragments and old versions of fragments may be reclaimed by the database. If a fragment is deleted or updated at a time after the merge timestamp, then the old version of the fragment is retained for use in point-in-time queries. Set this to 0 (the default) to let the system reclaim the maximum amount of disk space during merge activities. A setting of 0 will remove all deleted and updated fragments when a merge occurs. Set this to 1 before loading or updating any content to create a complete archive of the changes to the database over time. Set this to the current timestamp to preserve all versions of content from this point on. Set this to a negative number to specify a window of timestamp values, relative to the last merge, at ten million ticks per second. The timestamp is a number maintained by MarkLogic Server that increments every time a change occurs in any of the databases in a system (including configuration changes from any host in a cluster). To set to the current timestamp, click the <code>current timestamp</code> button; the timestamp is displayed in red until you press OK to activate the timestamp for future merges. For details on point-in-time queries, see the Application Developer's Guide.</p> <p>Click Get Current Timestamp to return the current merge timestamp.</p>
Retain Until Backup	<p>Specify whether the deleted fragments are retained since the last full or incremental backup. When enabled, <code>retain until backup</code> supersedes <code>merge timestamp</code>. Deleted fragments are not merged until backups are finished, regardless of the <code>merge timestamp</code> setting. Enabling <code>retain until backup</code> is same to setting the <code>merge timestamp</code> to the timestamp of the last backup. For more information, see Section 19.3.2, "Using Journal Archiving with Incremental Backups" [169].</p>

Database Setting	Description
Merge Blackout Periods	Specify times when merges are disabled. To specify a merge blackout period, click the Create tab and specify when you want the blackout to occur. You can make it a recurring blackout period, or specify a one-time blackout period. Use caution when setting large blackout periods when there are significant updates occurring on the system; merges are a normal part of the self-tuning mechanism of the database, and disabling them completely or for long periods of time can cause performance degradation.

15.3. Blackout Periods for Merges

Although merges are a normal part of system behavior, there are times when it is inconvenient for a merge to start. Merge blackout periods allow you to specify times when a merge should not begin. This section describes merge blackouts.

15.3.1. Understanding Merge Blackouts

A merge blackout is a predetermined time period in which automatic merges are disabled. A Merge that starts before a merge blackout period will continue until either it completes or until it is canceled, even if the merge continues into a blackout period. If you want to stop any merges at the beginning of a blackout period, you must cancel them manually as described in [Section 15.7.2, “Canceling a Merge” \[114\]](#). Because merges that start just before a blackout period will continue into the blackout period, if you want to be sure no merges occur during a time period you should make the blackout period start earlier. This is especially true for merges that might run a long time.

If the system determines that a merge is required and it is during a blackout period, the merge will not begin until the blackout period is past.

15.3.2. Configuring Merge Blackout Periods

To configure merge blackout periods, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click the **Merge Policy** menu item under your database. The **Merge Policy** page appears.
4. Click the **Create** tab.
5. Fill in the form as needed for the blackout period you want to create. Clicking the radio buttons will bring up more forms to complete.
6. Click **OK** to create the blackout period.

The new blackout period takes effect immediately.



WARNING

We recommend configuring a merge blackout with a small **limit merges to:** size: In a cluster configured for high availability, forests that recover from node restart will remain in `wait replication` until the next merge. Until that time, the acting primary forest cannot afford to fail over as there is no longer an acting replica with the mount state of `sync replicating`.

15.3.3. Deleting Merge Blackout Periods

To delete a merge blackout period, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click the **Merge Policy** menu item under your database. The **Merge Policy** page appears.

4. In the area corresponding to the blackout period you want to delete, click the **Delete** button.
5. Click **OK** on the confirmation page to delete the blackout period.

The blackout period is deleted immediately.

15.4. Merges and Point-in-Time Queries

When a merge occurs, it deletes all fragments from the stands being merged that have a system timestamp older than the configured `merge timestamp` (unless the `merge timestamp` is set to 0, in which case it will delete all fragments older than the current timestamp). This can keep multiple versions of some fragments in the database. You can query the older fragments using point-in-time queries. For details, see the [Point-in-Time Queries](#) in the *Application Developer's Guide*.

15.5. Setting a Negative Merge Timestamp to Preserve Fragments for a Rolling Window of Time

If you are doing update operations and you want the ability to roll back to the point in time when you started, you can set the `merge timestamp` to a negative number to preserve fragments for the specified number of ticks. The ticks are calculated at 10,000,000 ticks per second.

For example, if you want to preserve deleted fragments for 24 hours (relative to the last merge), then you can set the `merge timestamp` to `-864,000,000,000` (10,000,000 ticks/second times 60 seconds/minute times 60 minutes/hour times 24 hours/day). You can then use `xdmp:forest-rollback` on all of the forests in the database to roll back up to a day (or whatever time period you have set your negative merge timestamp).

If you do set a negative value for the `merge timestamp` parameter, keep in mind that you will keep deleted fragments for that period of time, so your database will be that much larger during that period. This could be significant, especially if you end up reloading several times during that period.

The following table shows the negative `merge timestamp` for specified periods of time.

Time Period to Preserve Fragments	Calculation	<code>merge timestamp</code> Value
5 minutes	$10000000 * 60 * 5$	-3000000000
1 hour	$10000000 * 60 * 60$	-36000000000
24 hours	$10000000 * 60 * 60 * 24$	-864000000000

15.6. Monitoring a Merge

There are two main places to look for monitoring information about merges.

15.6.1. Messages in the ErrorLog.txt File

MarkLogic Server logs INFO level messages to the `ErrorLog.txt` file whenever a merge begins, completes, or is canceled. Additionally, there are other log messages that are logged at more detail logging levels during a merge. The following are some sample log messages for a typical merge:

```
2006-04-20 13:43:11.151 Info: Merging /var/opt/MarkLogic/Forests/bill/00000004
and /var/opt/MarkLogic/Forests/bill/00000005 to /var/opt/MarkLogic/Forests/bill/00000006
2006-04-20 13:43:15.726 Debug: OnDiskStand /var/opt/MarkLogic/Forests/bill/00000006,
disk=47MB, memory=20MB
2006-04-20 13:43:15.726 Info: Merged 81 MB in 4 s at 20 MB/s to /var/opt/MarkLogic/
Forests/bill/00000006
2006-04-20 13:43:15.806 Debug: ~OnDiskStand /var/opt/MarkLogic/Forests/bill/00000004
2006-04-20 13:43:15.806 Debug: ~OnDiskStand /var/opt/MarkLogic/Forests/bill/00000005
2006-04-20 13:43:15.859 Info: Deleted /var/opt/MarkLogic/Forests/bill/000000042006-04-20
13:43:15.894 Info: Deleted /var/opt/MarkLogic/Forests/bill/00000005
```

If you cancel a merge, you will see an message similar to the following in the `ErrorLog.txt` file:

```
2006-05-08 17:45:44.027 Error: PooledThread::run: XDMP-CANCELED: Canceled
merge of stands: 13419435601900621379, 6182944041533805976 to: C:\Program
Files\MarkLogic\Data\Forests\bill\0000009a
```

By examining the `ErrorLog.txt` file, you can determine when a merge started, when it completed, which stands were merged together, what stand they were merged into, the size of the merge, and other useful information.



NOTE

There must be sufficient disk space on the file system in which the forest data is stored for a merge to complete successfully; if a merge runs out of disk space, it will fail with an error message. Also, there must be sufficient disk space on the file system in which the log files reside to log any activity on the system. If there is no space left on the log file device, MarkLogic Server will abort. Additionally, if there is no disk space available to add messages to the log files, MarkLogic Server will fail to start.

15.6.2. Database Status Page

The Database Status page lists the merge state, which indicates if a merge is going on, shows the size of the merge, and estimates how long it will take the merge to complete. Additionally, the Database Status page includes a link to cancel the current merge (for details, see [Section 15.7.2, “Canceling a Merge” \[114\]](#)).

During a merge, the merge rates are reported. The rate reported in the Merging status is the merge rate of all merges on the forest, averaged over the last few seconds. The Merge Reads and Writes reported in the Rates status are the merge rates for the current merge, averaged over the entire duration of that merge.

To access the Database Status page:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click the **Status** tab.

15.7. Explicit Merge Commands

This section describes how to manually initiate and cancel a merge.

15.7.1. Manually Initiating a Merge

You can manually initiate a merge, either by explicitly issuing the `xdmp:merge` command as described in [Merging the Forests in a Database](#) in the *Scripting Administrative Tasks Guide*, or by clicking the **Merge** button on the database configuration page of the Admin Interface. Either of these actions will immediately begin a merge on the database (if using `xdmp:merge`, on the database to which the App Server that responds to the request is connected, or if using the Admin Interface, the database being configured). Manually initiated merges continue even when merges are disabled for a database.

When you issue an `xdmp:merge` command or click the **Merge** button, it will begin a merge even if one would not occur automatically. If no options are specified to `xdmp:merge`, default values are used (not the configured values for the database).

**NOTE**

If you have updates occurring on the system while a merge is in progress, the new fragments will not be merged during the active merge operation; they will be merged during a subsequent merge.

Manually initiating a merge is useful when you have your merge controls set such that very large merges do not occur (for example, `merge min ratio` set to 1), but you want to run the large merges during a period of low activity on your system. It can also be useful for expunging deleted fragments that have not yet reached the threshold for automatic merges. Note that if a `merge timestamp` is set on the database, even a forced merge will not merge out deleted fragments up to the merge timestamp. In normal situations, deleted fragments are retained for a short period of time. If you want to forcibly merge those, you need to explicitly set the `merge-timestamp` option to the current timestamp in your `xdmp:merge` call.

The `xdmp:merge` API also allows you to specify options to the merge to control the maximum merge size, the forests which are merged, whether to merge to a single stand, as well as other options. For details, see `xdmp:merge` in the MarkLogic documents for Server-Side XQuery APIs.

15.7.2. Canceling a Merge

You can cancel a merge in the Database Status page of the Admin Interface. If you access the status page for a database during a merge, on the part of the status page for the stand(s) being merged, there is a cancel button (usually on the bottom right of the status page).

When you cancel a merge, the new stand that has not completed its merge is discarded, leaving the unmerged stands as they were before the merge began. Note that if you cancel an automatic merge, it might start up a new merge as soon as it is canceled (if the merge controls are set such that a merge is triggered). To avoid this situation, you can change some of the merge control parameters before you cancel an automatic merge.

To cancel a merge:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click the **Status** tab.
4. At the bottom right of the **Database Status** page, click the **Cancel** button on the row for the stand being merged.
5. Click **OK** on the **Cancel Merge** confirmation page.

The merge is canceled and the **Database Status** page appears again.

15.8. Configuring Merge Policy Rules

By changing some of the merge policy parameters, you can effectively control certain aspects of your merges. [Section 15.2.2, “Configuring the Merge Policy” \[110\]](#) describes what each parameter does. This section describes some scenarios with suggestions for how to tune the merge control parameters to satisfy the conditions.

15.8.1. Determining the Baseline for Your Merges

The merge characteristics of your system depend on many factors, including the size of your forests, the amount of update activity on the system, and the way your data is fragmented. If you feel you need to change the configuration of your merges, the first step is to determine the merge characteristics for your database. This requires running your system under normal loads, then analyzing the log files to determine the following data about your merges:

- average size of the merges
- average frequency of the merges
- average time it takes for the merges to complete

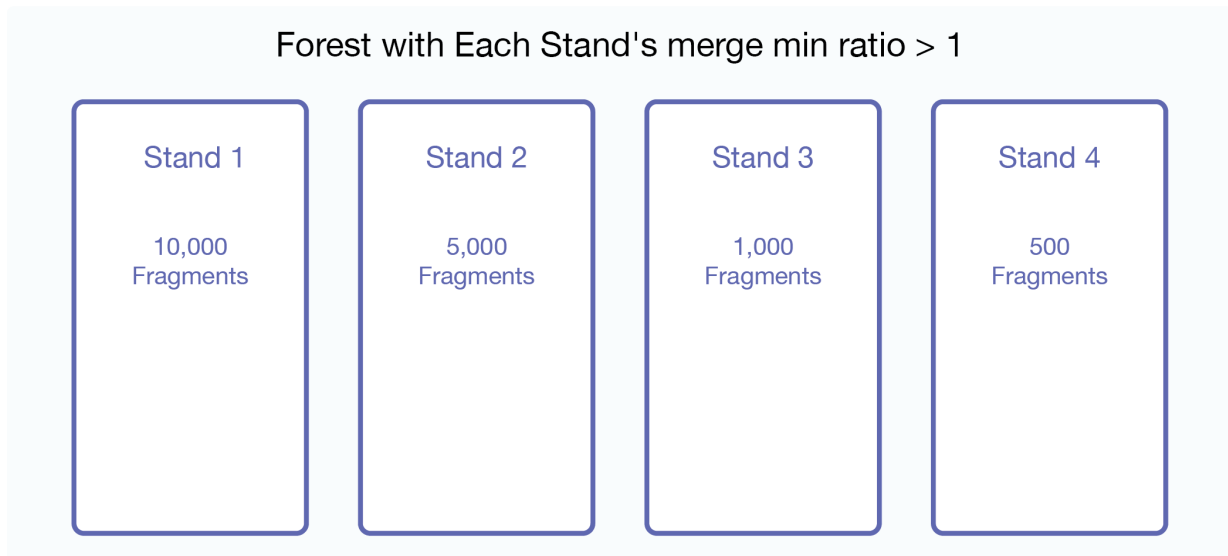
If it turns out that your merges are never taking more than a few minutes to complete, then there is probably no need to change any of your settings.

15.8.2. If You Want to Reduce the Number of "Large" Merges

In most cases, MarkLogic Server will perform relatively small merges just often enough to keep the system properly optimized. Small merges are generally not very disruptive and reasonably fast. In some cases, however, you might find that your merges are too large and are taking too much time. Exactly how large constitutes a "Large" merge is difficult to measure, but if you determine that your merges are too large, then you might want to try and configure your settings to avoid a really large merge.

One way to avoid large merges is to set the `merge max size` value. If you do set this value, however, you should only set it to a value as a temporary way to control your maximum merge size, as it can lead to a state where the database really needs to perform a large merge but cannot. Such a situation can lead to a poorly optimized system. One way to think about large merges is to compare them to sleeping for people; a person can go without much sleep for relatively short periods of time (a day or two or maybe even three for some people), but eventually, the person needs sleep or else he begins to function extremely poorly. Similarly, if a database is growing, it will eventually need to perform a large merge. Also, be careful not to set `merge max size` to such a small value that you end up with a very large number of stands. Always use care when setting the `merge max size` value, as you might end up with a large number of stands in your database, which can cause it to perform poorly and, when it reaches the maximum number of stands (64), will cause it to go offline.

Another way to accomplish a goal of reducing the number of large merges is to lower the value for `merge min ratio` to 1. A value of 1 for `merge min ratio` will not stop large merges from happening, but will make large merges only occur when the number of fragments in your largest stand is equal to the number of fragments in all of the other stands combined. Therefore, the only time merges will be more than 1/2 the size of your forest is when the fragment count of the sum of all but the largest stand is equal to or greater than the fragment count of the largest stand. To illustrate this, consider a forest with the following scenario:



If the `merge min ratio` is set to 1, then a stand can merge if the following ratio is less than 1:

$$\frac{\text{\# of fragments in a stand}}{\text{total \# of fragments in all other smaller stands in the forest}} < 1$$

Substituting in the values from the example for Stand 1 yields

$$10000 / (5000 + 1000 + 500) = 10000 / 6500 = 1.54$$

which is greater than 1. Therefore, Stand 1 is not merged.

Next, putting in the values for Stand 2 yields

$$5000 / (1000 + 500) = 5000 / 1500 = 3.33$$

which is greater than 1. Therefore, Stand 2 is not merged.

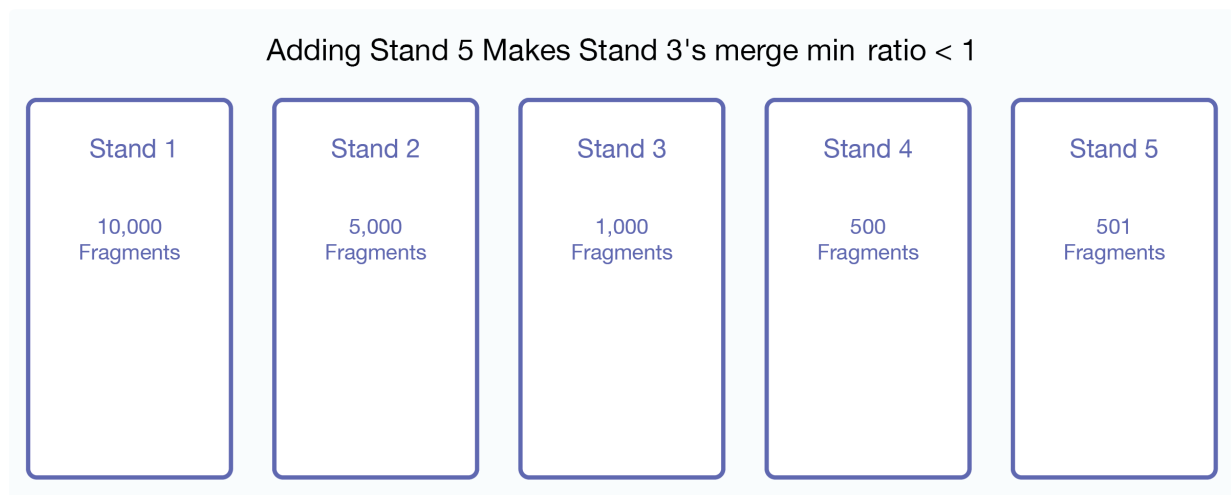
Next, putting in the values for Stand 3 yields

$$1000 / 500 = 2.0$$

which is greater than 1. Therefore, Stand 3 is not merged.

Therefore, if the forest remains in a steady state (that is, no new content is added), then a `merge_min_ratio` of 1 will cause this forest to not be merged.

Now, consider that a load is happening during this time and a stand that has 501 fragments is saved into the forest. The result is 5 stands as follows:

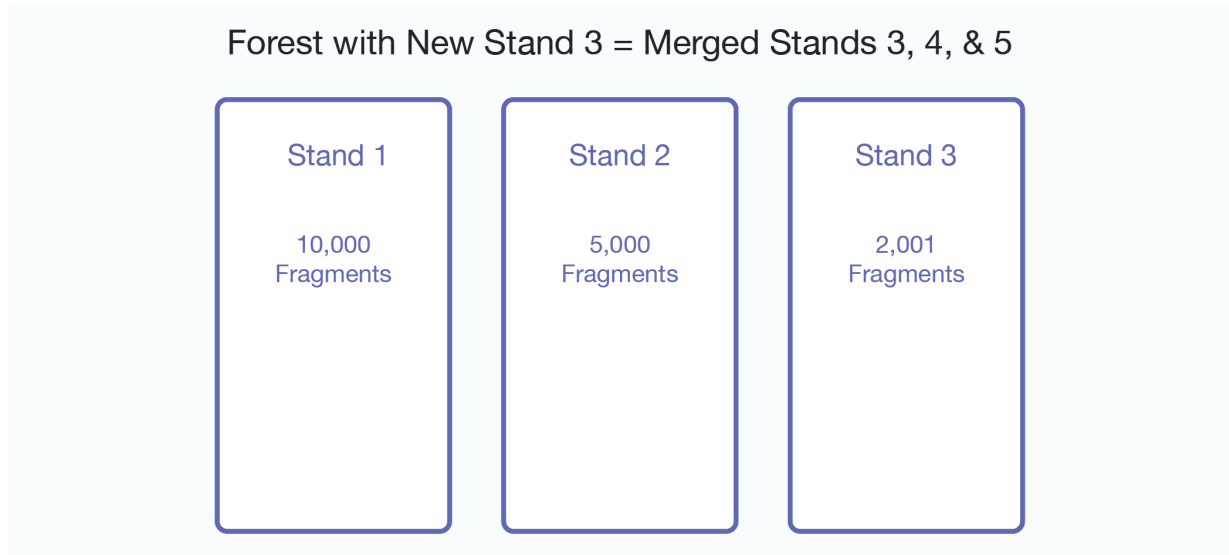


Now, substituting in the values for Stand 3 yields

$$1000 / (500 + 501) = 1000 / 1001 = 0.99$$

which is less than 1. Therefore, Stand 3 is merged.

Note that Stands 4 and 5 are smaller than Stand 3, so the sum of the fragments in those stands appear in the denominator of the `merge_min_ratio`. Therefore, Stands 3, 4, and 5 are merged. Therefore, a `merge_min_ratio` of 1 will cause this forest to be merged down to 3 stands, where Stands 1 and 2 remain unmerged and Stands 3, 4, and 5 are merged together into a new stand. The stands then look like this:



Note that, in a real world scenario with relatively large forests, this scenario (where the sum of the smaller stands fragment counts have as many fragments as the largest stand) will not happen very often, but will happen occasionally. For example, if another 3,000 fragments continued to accumulate in this forest, then Stand 1 would merge with the other stands.

15.8.3. Other Solutions

In some cases, changing the merge parameters might not be the best solution for your system. For example, if your merges are taking a very long time due to slow disk drives or other system contention, addressing those issues might do more to help your merge times than any amount of tuning can do. Also, if your merges are extremely large, it could be that the forests are larger than optimal. There is no fixed maximum size for a forest, but experience in the field has shown that when forests grow over 512 GB, query performance tends to start to decrease while merge times tend to start to increase. If your forests are larger than 512 GB, consider breaking them into multiple forests.

16. Database Rebalancing

As your needs for data in a database expand and contract, the more evenly the content is distributed among the database forests, the better its performance and the more efficient its use of storage resources. This section describes the database rebalancing mechanism that enables MarkLogic Server to evenly distribute content among the database forests.

16.1. Overview of the Database Rebalancer

A database rebalancer consists of two parts: an *assignment policy* for data insert and rebalancing and a *rebalancer* for data movement. The rebalancer can be configured with one of several assignment policies, which define what is considered “balanced” for a database. You choose the appropriate policy for a database. The rebalancer runs on each forest and consults the database's assignment policy to determine which documents do not “belong to” this forest and then pushes them to the correct forests.



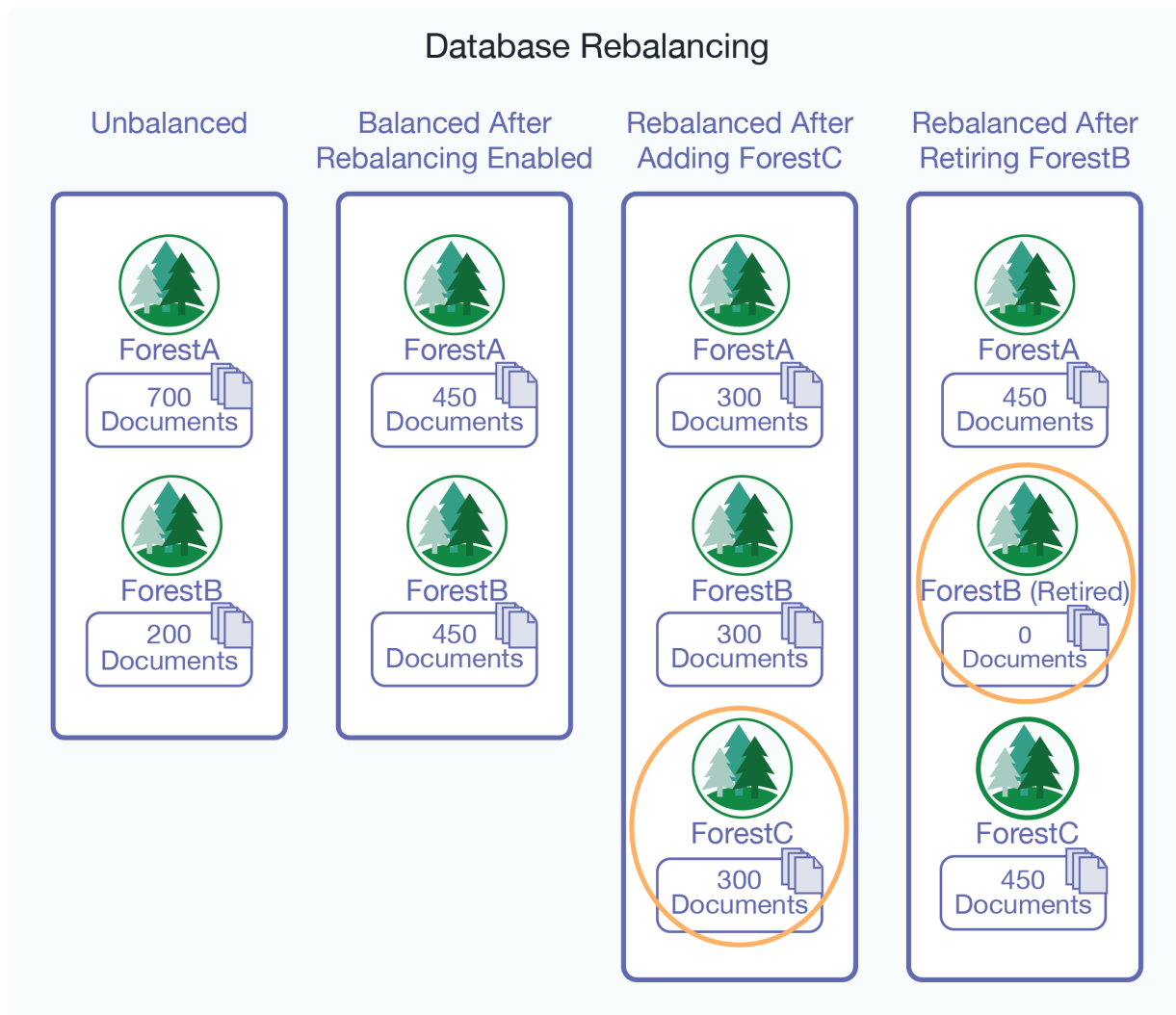
NOTE

Document loads and inserts into the database follow the same document assignment policy used by the rebalancer, regardless of whether the rebalancer is enabled or disabled.

When you add a new forest to a database configured with a rebalancer, the database will automatically redistribute the documents among the new forest and existing forests. You can also *retire* a forest in a database to remove all of the documents from that forest and redistribute them among all of the remaining forests in the database.

In addition to enabling and disabling on the database level, the rebalancer can also be enabled or disabled at the forest level. For the rebalancer to run on a forest, it must be enabled on both the database and the forest.

The following illustration shows how 900 documents might be distributed between database forests before rebalancing, after rebalancing, after adding a new forest to the database, and after retiring a forest from the database.



16.2. Rebalancer Trigger Events

In addition to the rebalancer periodically rebalancing the database, the following events trigger the rebalancer process:

- Any configuration changes to the database, such as adding a new forest or retiring an existing forest.
- Upon completion of a restore operation on the database.
- Upon completion of a backup operation on the database.

16.3. Rebalancer Document Assignment Policies

A database is given an *assignment policy* that defines the logic used by the forests when reassigning documents to the other forests participating in the rebalancer process. Though they run in separate threads, both the rebalancer process and the document load/insert process follow the same assignment policy set on the database for the rebalancer.

16.3.1. Bucket Assignment Policy

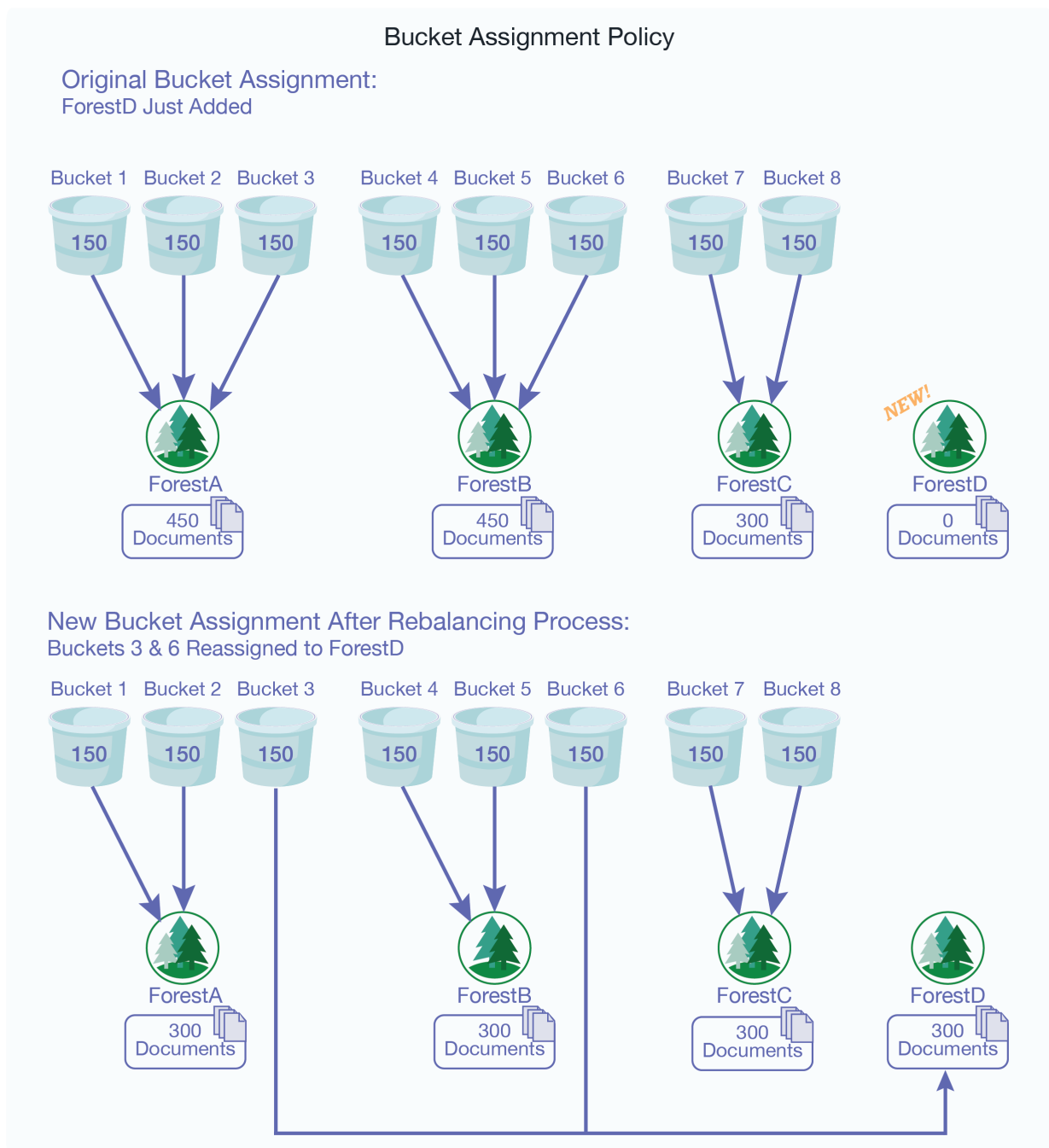
The bucket policy uses the URI of a document to decide which forest the document should be assigned to. The URI is first "mapped" to a bucket then the bucket is "mapped" to a forest. The mapping from a bucket to a forest is kept in memory for fast access. The number of buckets is always 16K, regardless of the number of forests in the database.



NOTE

How document URIs are mapped to buckets and buckets are mapped to forests are non-configurable implementation details.

Though there are 16K buckets used by the bucket assignment policy, for the purposes of the example illustrated below, assume there are eight buckets that distribute the 1200 documents across three forests: ForestA, ForestB, and ForestC and that the document URIs allow for even distribution of them among the buckets. ForestD is then added to the database and the rebalancer moves 1/3 of the documents from Forests A and B to ForestD by reassigning Bucket 3 from ForestA to ForestD and Bucket 6 from ForestB to ForestD.



The bucket assignment policy is, in most situations, the most efficient document assignment policy because it is deterministic and it moves the least amount of data of the deterministic assignment policies.

16.3.2. Segment Assignment Policy

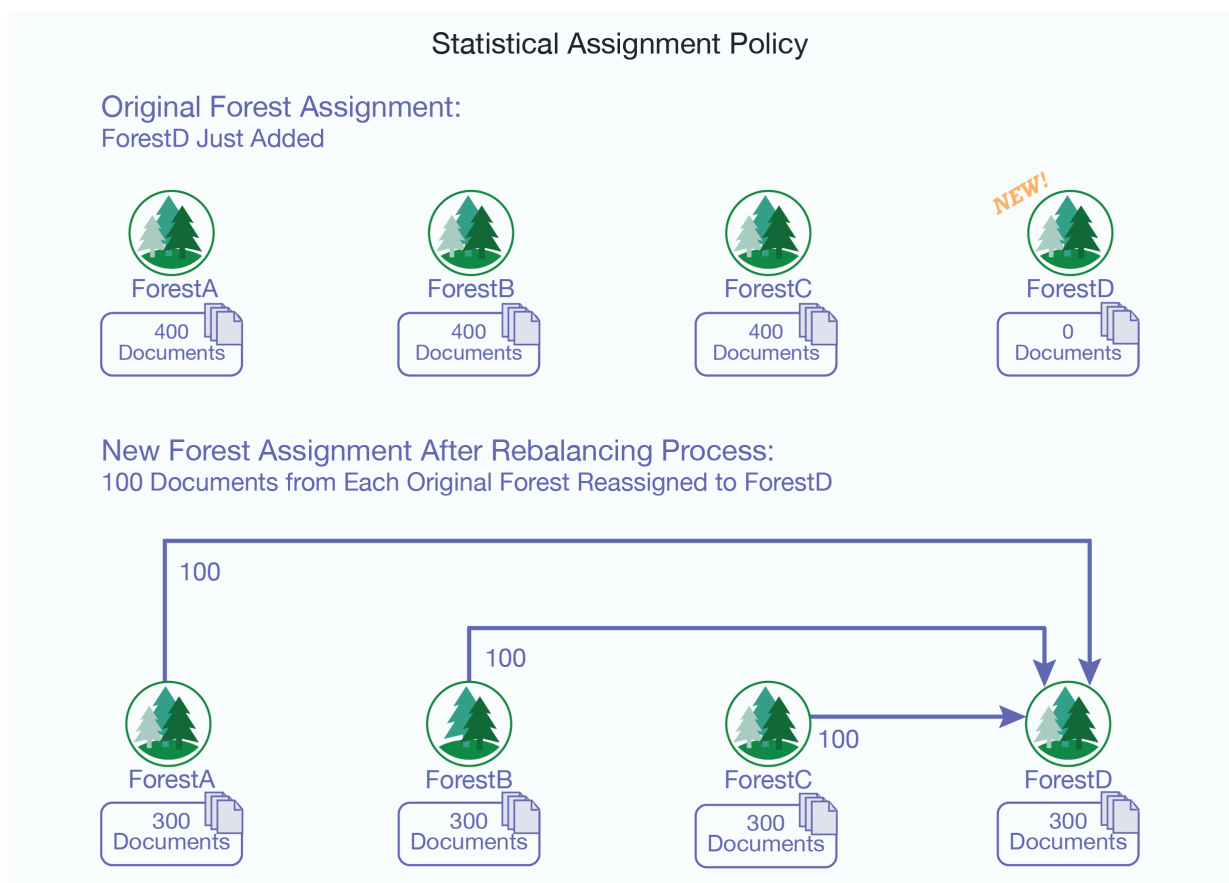
Unlike the legacy policy, described in [Section 16.3.6, "Legacy Assignment Policy" \[125\]](#) that ensures that documents are evenly distributed across forests in the database, the segment policy ensures that fragments are evenly distributed across the forests. The segment policy assigns fragments to forests based on their document URIs to allow for fast locking.

The segment policy is the most efficient rebalancing policy when you are adding or reducing the number of forests by 30% or more. For example, if the number of forests doubles, the half of the fragments in the existing forests are assigned to the newly added forests. Conversely, if the number of forests is reduced by half, all of the fragments in a retired forests are assigned to the remaining forests.

16.3.3. Statistical Assignment Policy

The statistical policy does not map a URI to a forest. Instead, each forest keeps track of how many documents it has and broadcasts that information to the other forests through heartbeats. The rebalancer then moves documents from the forests that have the most documents to the forests that have the least number of documents. When a new forest is added, the statistical policy moves the least number of documents to get to a balanced state. All forests don't have to have the exact same amount of documents for a database to be considered "balanced."

For example, as shown in the figure below, a new forest, ForestD, is added to the database that already has three forests: ForestA, ForestB, and ForestC, each contains 400 documents. Each of the existing forests move 100 documents to the new forest, ForestD.



**NOTE**

The number of documents in above example is used for the purposes of illustrating the behavior of the rebalancer when the statistical policy is set. In practice, it is inefficient to move such a small number of documents between forests. Typically, you will not see any significant rebalancing of documents between forests until the number of documents in the database exceeds 100,000.

If your database is balanced (the document count on each forest is roughly the same), setting the assignment policy to statistical will not trigger major data movement and any new inserts from then on will be automatically balanced across the forests.

16.3.4. Range Assignment Policy

The range policy is designed for use with Tiered Storage Range Partitions described in [Section 17.3, “Range Partitions” \[134\]](#). It uses a range index value to decide which forest a document should be assigned to. When setting the range policy, you specify a range index for use as the *partition key* and configure each forest attached to the database with a *range* that defines a lower and upper end.

**NOTE**

Avoid using the range policy to manage documents that might have more than one value for a range index, as the behavior in such a circumstance is undefined.

There may be multiple forests that cover the same range, but two forests cannot have partially overlapped ranges. For example, it is valid for both ForestA and ForestB to cover (1 to 10) but not valid for ForestA to cover (1 to 6) while ForestB covers (4 to 10). It is also not valid for ForestA to cover (1 to 10) while ForestB covers (4 to 9). Among those forests that cover the same range, documents are assigned to the forests based on their document count, following a similar mapping process as the statistical policy described in [Section 16.3.3, “Statistical Assignment Policy” \[121\]](#).

**NOTE**

In order to accommodate range “gaps” and documents that do not contain an element used as the partition key, you should always configure a *default forest*, as described below.

If a document has been processed by the Content Processing Framework (CPF), the property documents associated with the document may have a partition key value that is different from that in the document. When using the range policy, you may want to use the `xdmp:document-add-properties` or `xdmp:document-set-properties` function to put the same partition key value as specified in the document into the property documents to ensure that they are moved to the same forest as the original document. For example, the partition key is `creation-date` and the `example.xml` document has a `creation-date` of `2010-01-02`, but its associated property documents contain no `creation-date` element. You could then use the `xdmp:document-add-properties` function as follows to add a matching `creation-date` element to the `example.xml` property documents.

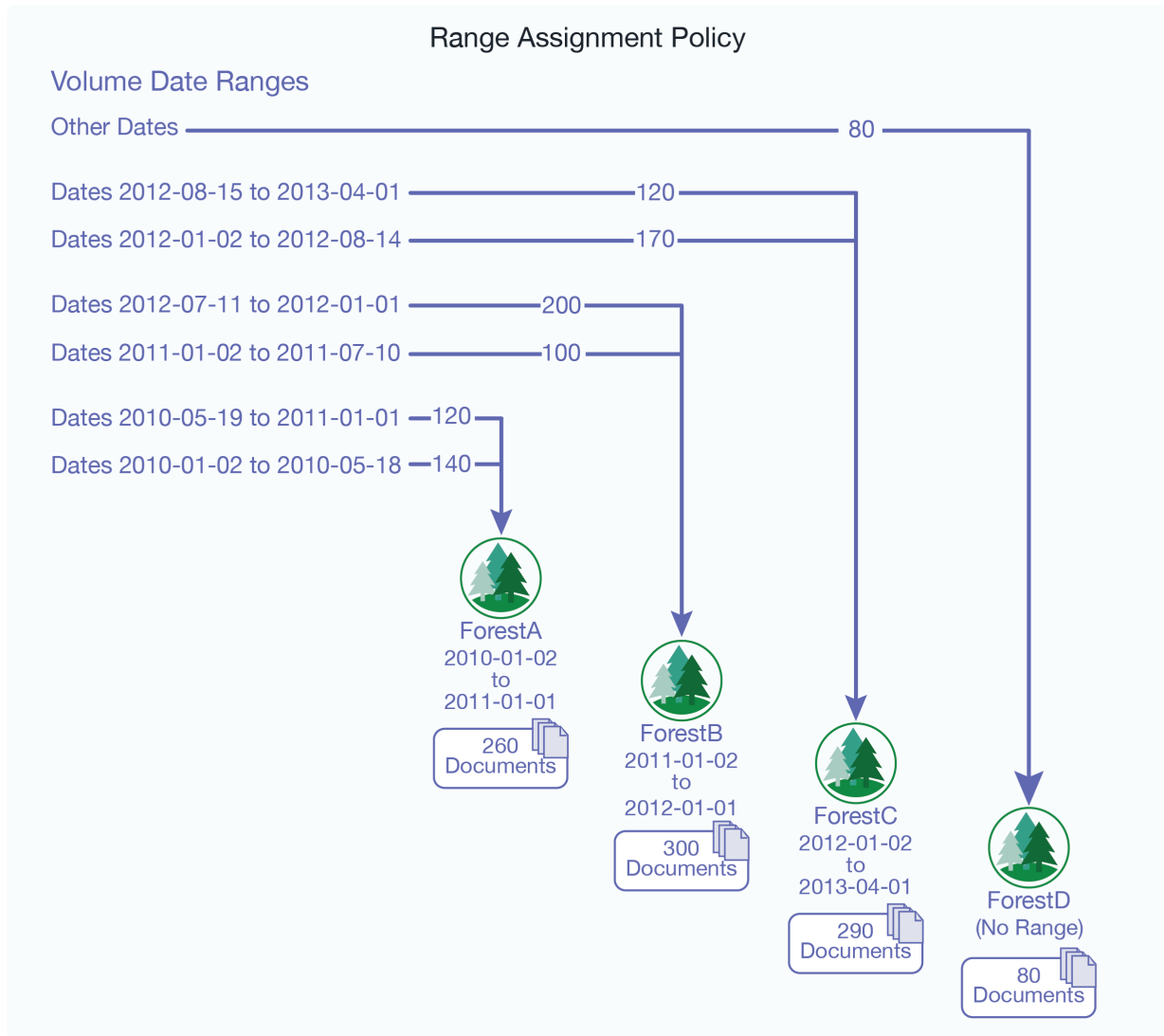
```
xdmp:document-add-properties(  
  "example.xml",  
  (<creation-date>2010-01-02</creation-date>))
```

A forest with no range value behaves as the default forest, which means that documents that do not fit into any of the ranges set on the other forests are moved to the default forest. You cannot retire a forest unless there is another forest for the documents to move to, which means that there must either be another forest with the same range as the retired forest or that there is a default forest (no range set) attached to the database. If a database contains no default forest, an attempt to retire a forest containing documents with partition key values that do not match the ranges in the other forests will not be successful.

**NOTE**

You should always define a default forest when configuring the range assignment policy.

For example, as shown in the figure below, you have documents that are organized into 6 volumes and each document contains a `<creation-date>` element that indicates when that document was created. You can create an element range index, named `creation-date`, of type `date` and identify `creation-date` as the partition key for the range policy. If you have four forests, you can set the lower bound of the range on the ForestA to `2010-01-02` and the upper bound to `2011-01-01`; on ForestB, the lower bound to `2011-01-02` and the upper bound to `2012-01-01`, and on ForestC, the lower bound to `2012-01-02` and the upper bound to `2013-04-01`. The fourth forest, ForestD, is designated as the default forest by not specifying a range. Any documents that have dates that fall outside of the date ranges set for the other forests and directed to the default forest.



16.3.5. Query Assignment Policy

The query assignment policy, like the range assignment policy, is designed for use with Tiered Storage Query Partitions described in [Section 17.4, "Query Partitions" \[135\]](#). The query assignment policy works in a similar manner as the range assignment policy. However, rather than using lower and upper bound values to determine which documents are in a partition, the query assignment policy uses a query to determine which documents are in a partition. Users have the flexibility to use multiple keys and use different conditions for different types of documents.

With range assignment policy, the boundaries are fixed. However, you might want to rebalance the documents based on the difference between the entry time and the current time. When a range query compares a dateTime with duration, it becomes an age query.

For example, this query will match documents where "LastModified" is within past year:

```
cts:element-range-query(
  xs:QName("LastModified"),
  ">=",
  xs:yearMonthDuration("P1Y"))
```

When creating a query partition, you assign it a partition number. Unlike range partitions, queries set for partitions using the query assignment policy can have "overlaps," but, in the event of an overlap, the partition with lower number is selected before partitions with higher numbers.



NOTE

As is the case with range assignment policy, you should always define a default partition when configuring the query assignment policy.

Here is an example of a query assignment policy setup. MD and AD are elements in the documents:

Partition Name	Tier1	Tier2	Tier3	Tier4
Partition Number	1	2	3	4
Query	(Termination eq yes) OR (Source eq "Hiring" AND MD > 30 days) OR (Source eq "CFO" AND MD > 30 days)	(Source eq "Hiring" AND MD <= 30 days AND MD > 1 year) OR (Source eq "CFO" AND MD <= 30 days AND MD > 60 days) OR (Source eq "Benefits" AND AD > 1 year)	(Source eq "Hiring" AND MD <= 1 year AND MD > 3 years) OR (Source eq "CFO" AND MD <= 60 days) OR (Source eq "Benefits" AND AD <= 1 year)	(Source eq "Hiring" AND MD <= 3 years)
Default	Yes	No	No	No

There is only one `cts:query` per partition.

When the query assignment policy is used, these rules are used for document insert:

- The partition number is used for priority. If there is more than one query that match the document, the partition with the lower partition number is used.
- If none of the queries matches the document, the default partition is used.
- If there is no default partition, the forests without a partition number are used.
- Otherwise, it is an error.

Among the forests in a partition, the documents are assigned to the forests using the statistical assignment policy.

The query requires the proper indexes to be configured in the database. The complexity of the query affects the performance of insert and rebalancing. Therefore slow queries such as those with wildcard matching are not recommended.

See [Section 17.7.2, "Setting the Query Assignment Policy for the Query Partition" \[143\]](#) for details on how to set the query assignment policy.

16.3.6. Legacy Assignment Policy

After upgrading to MarkLogic 7.0 or a later version, existing databases will be configured with the rebalancer disabled and the legacy assignment policy. This is to preserve the expected behavior when new documents are loaded into the database.



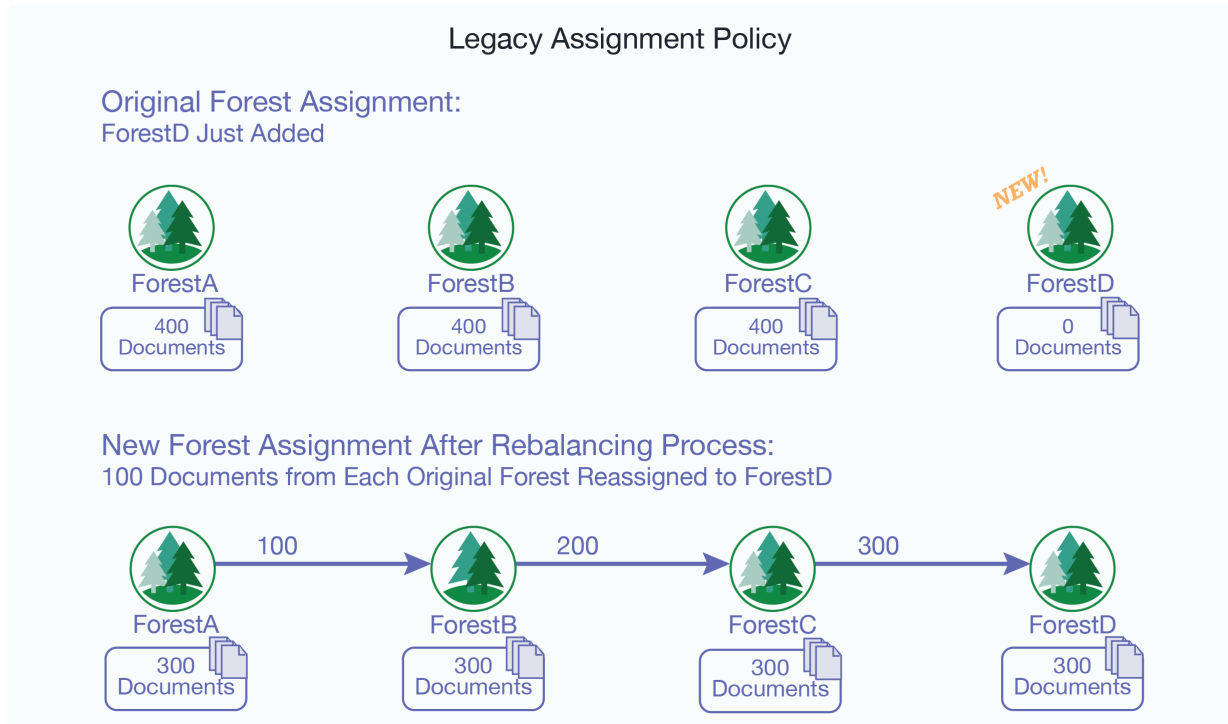
NOTE

Under most circumstances you would not use the legacy policy when the database rebalancer is enabled. The segment policy, described in [Section 16.3.2, "Segment Assignment Policy" \[121\]](#), is generally preferred over the legacy policy.

The legacy policy uses the URI of a document to decide which forest the document should be assigned to. The mapping from a URI to a forest uses the same algorithm as the one used on older releases of MarkLogic Server.

For example, as shown in the figure below, a new forest, ForestD, is added to the database that already has three forests: ForestA, ForestB, and ForestC, each contains 400 documents because the document URIs allow for even distribution of them among the forests. The data is rebalanced as follows:

- ForestA moves 100 documents to ForestB
- ForestB moves 200 documents to ForestC
- ForestC moves 300 documents to ForestD



The legacy policy is the least efficient rebalancer policy, as it requires the greatest amount of document movement to rebalance the documents among the forests. For this reason, you should only use the legacy policy on legacy databases with the rebalancer disabled.

16.3.7. Summary of Assignment Policies

The following table summarizes the characteristics of the rebalancer assignment policies:

Policy	Data Movement	Deterministic?	Backward Compatible?
Bucket	Less	Yes (URI based)	No
Segment	Most	Yes (URI based)	No
Statistical	Least	No	No
Range	Less	Yes (Partition key based)	No
Query	Less	Yes (Partition key based)	No
Legacy	Most	Yes (URI based)	Yes

16.4. How the Rebalancer Moves Documents

There are many similarities between the rebalancing process and the reindexing process. Rebalancing is configured at the database level and individual rebalancing processes run separately on each forest.

The main task of the rebalancer is to consult the assignment policy associated with the database to get a list of documents (URIs) that do not “belong to” this forest and then push them out to the right forests. The deletion of documents from the rebalancing forest and the insertion of them into the right forests happens in the same transaction. All fragments with the same URI are handled by the same transaction. Each transaction moves a batch of documents.

When rebalancing is enabled, you can configure the rebalancer throttle for a database. The rebalancer throttle works the same as the reindexer throttle in that it establishes the priority of system resources devoted to rebalancing. When the rebalancer throttle is set to 5 (the default), the rebalancer works aggressively, starting the next batch of rebalancing soon after finishing the previous batch. When set to 4, it waits longer between batches, when set to 3 it waits even longer, and so on until when it is set to 1, it waits the longest. The higher numbers give rebalancing a higher priority and uses the most system resources.

16.4.1. How Data is Moved When a Forest is Attached to the Database

Attaching an empty forest to a database is the same as adding a new forest. If the forest contains existing documents, they will participate in the rebalancing with the documents that are in the other forests that are already attached to the database.

16.4.2. How Data is Moved When a Forest is Retired from the Database

If a rebalancer-enabled forest is retired, the rebalancer empties the forest by “balancing out” all of the documents to the other forests attached to the database. The rebalancers on other forests re-calculate document routing as if the retired forest no longer exists. For new inserts, the retired forest is excluded from consideration by the document assignment policy.



NOTE

Retire is a separate operation from detach or delete. A read-only forest cannot be retired. To preserve all of the documents in the database, you must first retire a forest to rebalance the documents on the remaining forests in the database before detaching that forest.

16.5. Configuring the Rebalancer on a Database

You can configure and monitor the rebalancing process through the Admin Interface or the Admin APIs.

To configure the rebalancer on a database, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. On the **Configure** tab, scroll down to the **Rebalancer Enable** field and click **true**.
4. From the **Assignment Policy** pull-down menu, select the assignment policy. For details on the available rebalancer assignment policies, see [Section 16.3, “Rebalancer Document Assignment Policies” \[119\]](#).
5. From the **Rebalancer Throttle** pull-down menu, select the rebalancer throttle setting. For details on the rebalancer throttle, see [Section 16.4, “How the Rebalancer Moves Documents” \[126\]](#).
6. Click **OK**.

16.6. Configuring the Rebalancer on a Forest

In addition to enabling and disabling on the database level, as described in [Section 16.5, “Configuring the Rebalancer on a Database” \[127\]](#), the rebalancer can also be enabled or disabled on each individual forest. For the rebalancer to run on a forest, it must be enabled on both the database and the forest.

**NOTE**

The rebalancer is enabled on each new forest by default.

To configure the rebalancer on a forest, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. In the left-tree menu under the database name, select **Forests**.
4. On the **Configure** tab, click the name of the forest for which you want to enable or disable the rebalancer.
5. In the **Configure** tab, scroll down to **Rebalancer Enable**, and set it to **true** to enable the rebalancer or **false** to disable the rebalancer.
6. If you have configured the forest's database with the range assignment policy, you can set the range for this forest in the **Lower Bound** and upper bound fields. Do not set a range if this forest is to serve as a default forest.
7. Click **OK**.

16.7. Retiring a Forest from the Database

You can “retire” a forest from a database in order to move all of its documents to the other forests and rebalance them among those forests, as described in [Section 16.4.2, “How Data is Moved When a Forest is Retired from the Database” \[127\]](#). If you want to preserve forest documents in a database, you must first retire the forest before detaching it from the database.

To retire a forest from a database, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. In the left-tree menu under the database name, click **Forests**.
4. On the **Configure** tab, check the **Retired** box for the forest you want to retire from the database. If you want to preserve forest documents in a database, leave the **Attached** box checked.
5. Click **OK**. The documents in the retired forest will be evenly redistributed to the other forests in the database.
6. After the rebalancer has emptied the retired forest, if the forest is no longer needed, you can detach the forest from the database, as described in [Section 12.3, “Attaching and/or Detaching Forests to/from a Database” \[89\]](#).

16.8. Checking the Rebalancer Status

When the rebalancer is enabled on the database, you can check the state of the rebalancer, along with an estimated completion time, on the **Status** tab of the **Database** page.

When the rebalancer is disabled on the database, you can click **Show rebalance** at the top of the **Status** tab to view the number of fragments that are pending rebalancing.

16.9. How the Rebalancer Interacts with Other Database and Forest Settings

This section describes how the database rebalancer interacts with other database and forest settings.

16.9.1. Database Replication

If you have configured a database for database replication and that database is enabled for rebalancing with the segment, legacy or bucket policy, the order of the forests in the database configuration is

important, and it should be the same on the master and replica databases. If the order of the master and replica forests is different, you will see a message similar to the following in the log:

```
Warning: forest order mismatch: local forest XXX is at position A while foreign master forest YYY (cluster=ZZZ) is at position B.
```

Should you see this error, you can execute the `admin:database-reorder-forests` function on the replica database to reorder the forests to match the same order as on the master. If you do not reorder the forests so the master and replica match, then rebalancing will occur if replication is deconfigured.

16.9.2. Restoring a Database from a Backup

If you have a database enabled for database rebalancing with the segment, legacy or bucket policy, the order of forests on the database may differ from the order of forests when the database was backed up. You can execute `xdmp:database-restore-validate` function to return a backup-plan containing a `database` element that shows the order of the forests when the backup was done. If the order of the forests do not match, then you should execute the `admin:database-reorder-forests` function to reorder the forests on your database before restoring it from the backup.



NOTE

When using the segment, legacy or bucket policy, if the order of forests on the database being restored differs from the order of forests when the database was backed up, the restore operation may trigger major data movement between the forests on the restored database.

16.9.3. Tiered Storage

The range assignment policy described in [Section 16.3.4, “Range Assignment Policy” \[122\]](#) is designed to support tiered storage. For details on tiered storage, see [Section 17, “Tiered Storage” \[131\]](#).

16.9.4. Fast Locking

Fast locking works with the segment, legacy, and bucket policy. However, a database cannot use the statistical policy or the range policy with fast locking. With the statistical policy, two transactions that insert the same URI do not know which forest the other one will pick, so the server must use strict locking. With the range policy, there may be two transactions that insert the same URI but with different values for the range index, so the server must use strict locking.

16.9.5. Delete-Only and Read-Only Forests

Delete-only (DO) and read-only (RO) forests affect how documents are assigned. The following table summarizes the interaction between this feature and DO/RO forests:

Policy	New Insert	RW -> DO/RO	DO/RO -> RW
Legacy	DOs/ROs are excluded from assignment.	Recalculate routing for every URI; lots of movement.	Recalculate routing for every URI; lots of movement.
Segment	DOs/ROs are excluded from assignment.	Recalculate routing for every URI; lots of movement.	Recalculate routing for every URI; lots of movement.
Bucket	DOs/ROs are still included in the routing table calculation, but a URI that belongs to a DO/RO is re-assigned in a deterministic way.	No movement.	Only move documents that are reassigned (to non DO/RO) during insert.
Statistical	DOs/ROs are excluded from assignment; RWs get balanced load.	No movement since all RWs are already balanced.	Some movement until all RWs are balanced.

Policy	New Insert	RW -> DO/RO	DO/RO -> RW
Range and Query	DOs/ROs are excluded from assignment. Within each partition, RWs get balanced load.	No movement within a partition because RWs are already balanced.	Some movement within a partition until all RWs are balanced.

Note that the second and the third columns cover what the rebalancers on RWs do when a forest is changed from RW to DO/RO or DO/RO to RW.

The rebalancer on a RO forest is always off. The rebalancer on a DO forest is off unless it is "retired".

A flash-backup forest is generally handled as a RO forest except that on new inserts, if the assignment logic cannot find a forest to insert the documents but there is at least one flash-backup forest, a Retry (instead of Exception) is thrown.

16.10. Rebalancer Settings after Upgrading from an Earlier Release

For a brand new database, the rebalancer is enabled by default and the assignment policy is bucket. The bucket policy moves less data than the legacy policy when adding or deleting a forest and it is still deterministic.

After upgrading from an earlier release of MarkLogic Server, the rebalancer is disabled on existing databases and the policy is set to legacy.

At the forest level, in both cases, the rebalancer is enabled by default.

17. Tiered Storage

MarkLogic Server allows you to manage your data at different *tiers* of storage and computation environments, with the top-most tier providing the fastest access to your most critical data and the lowest tier providing the slowest access to your least critical data. Infrastructures, such as Hadoop and public clouds, make it economically feasible to scale storage to accommodate massive amounts of data in the lower tiers. Segregating data among different storage tiers allows you to optimize trade-offs among cost, performance, availability, and flexibility.

Tiered storage is supported by the XQuery, JavaScript, and REST APIs. This section describes the tiered storage operations using the REST API, which supports all of the operations you will want to integrate into your storage-management scripts.



NOTE

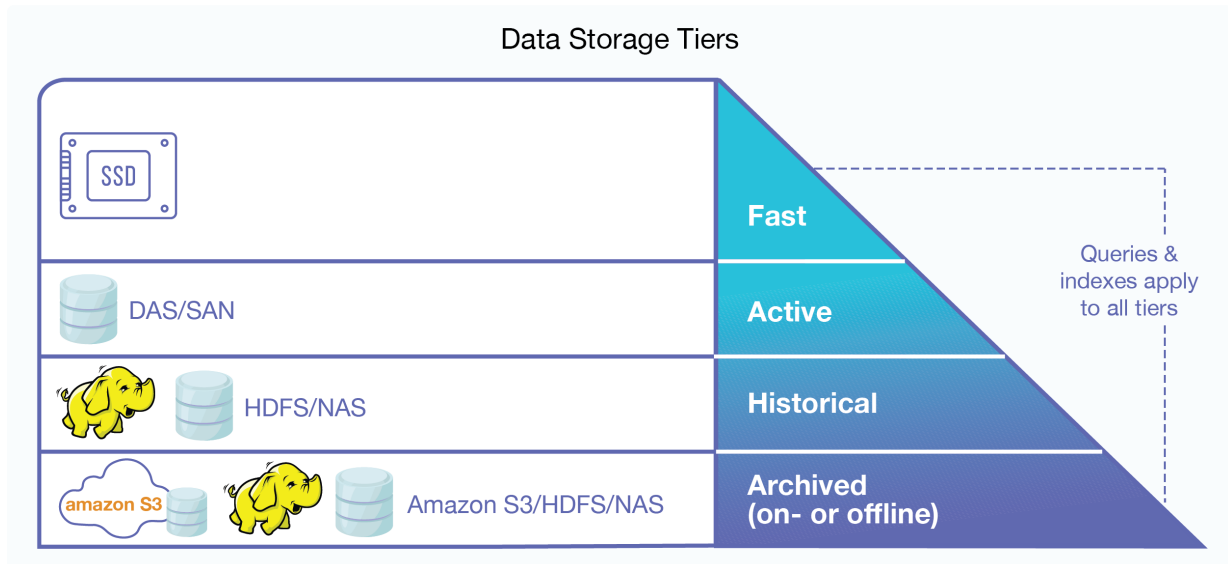
To use Tiered Storage, a license that includes Tiered Storage is required.

17.1. Terms Used in This Section

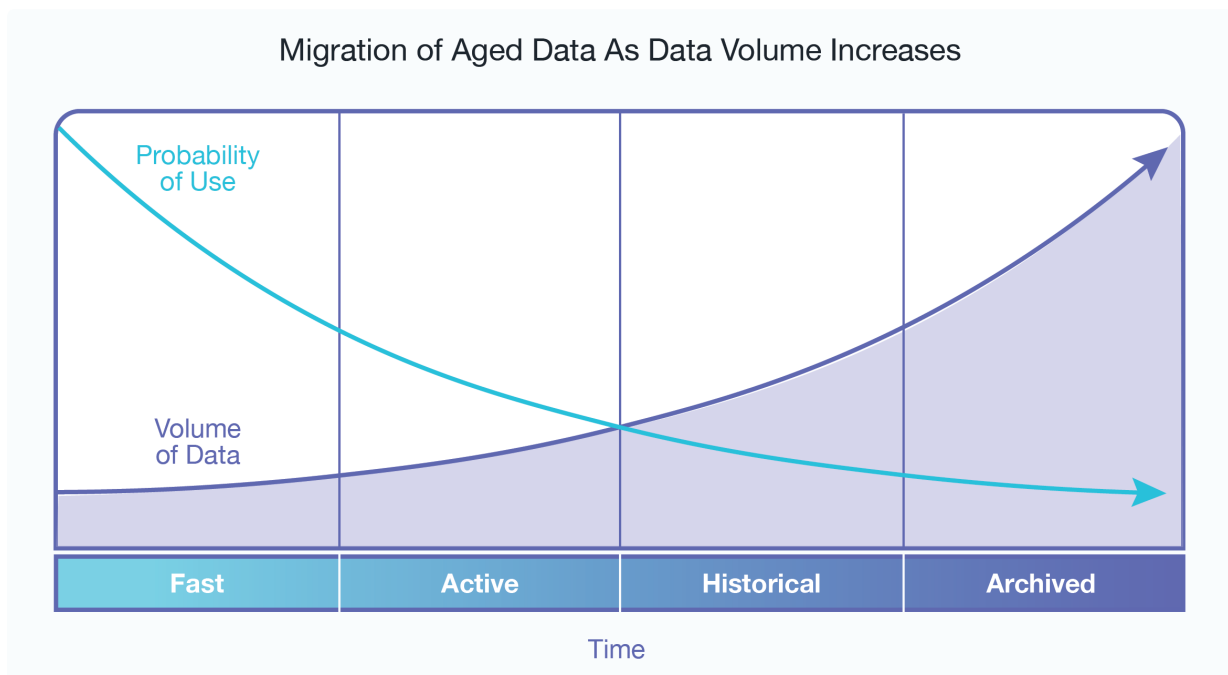
- A *Partition* is a set of forests sharing the same name prefix and same partition definition. Typically forests in a partition share the same type of storage and configuration such as updates allowed, availability, and enabled status. Partitions are based on forest naming conventions. A forest's partition name prefix and the rest of the forest name are separated by a dash (-). For example, a forest named 2011-0001 belongs to the 2011 partition.
- A *Range Partition* is a partition that is associated with a range of values. Documents with a partition key value that fall within the range specified for a partition are stored in that range partition.
- A *Query Partition* is a partition that is associated with a query. Documents that are returned by the query specified for a query partition are stored in that query partition.
- A *Partition Key* defines an element or attribute on which a range index, collection lexicon, or field is set and defines the context for the range set on the range partitions in the database. The partition key is a database-level setting.
- A *Default Partition* is a partition with no defined range or query. Documents that have no partition key or a partition key value that does not fall into any of the partition ranges or queries are stored in the default partition.
- A *Super-database* is a database containing other databases (sub-databases) so that they can be queried as if they were a single logical database.
- A *Sub-database* is a database contained in a super-database.
- *Active Data* is data that requires low-latency queries and updates. The “activeness” of a particular document is typically determined by its recency and thus changes over time.
- *Historical Data* is less critical for the lowest-latency queries than “active” data, but still requires online access for queries. Historical data is not typically updated.
- *Archived Data* is data that has aged beyond its useful life in the online storage tiers and is typically taken offline.
- An *Online* partition or forest is available for queries and updates.
- An *Offline* partition or forest is not available for queries, but is tracked by the cluster. The benefit of taking data offline is to spare the RAM, CPU, and network resources for the online data.
- The *Availability* of a partition or forest refers to its online/offline status.

17.2. Overview of Tiered Storage

The MarkLogic tiered storage APIs enable you to actively and easily move your data between different tiers of storage. For example, visualize how data might be tiered in different storage devices in a pyramid-like manner, as illustrated below:



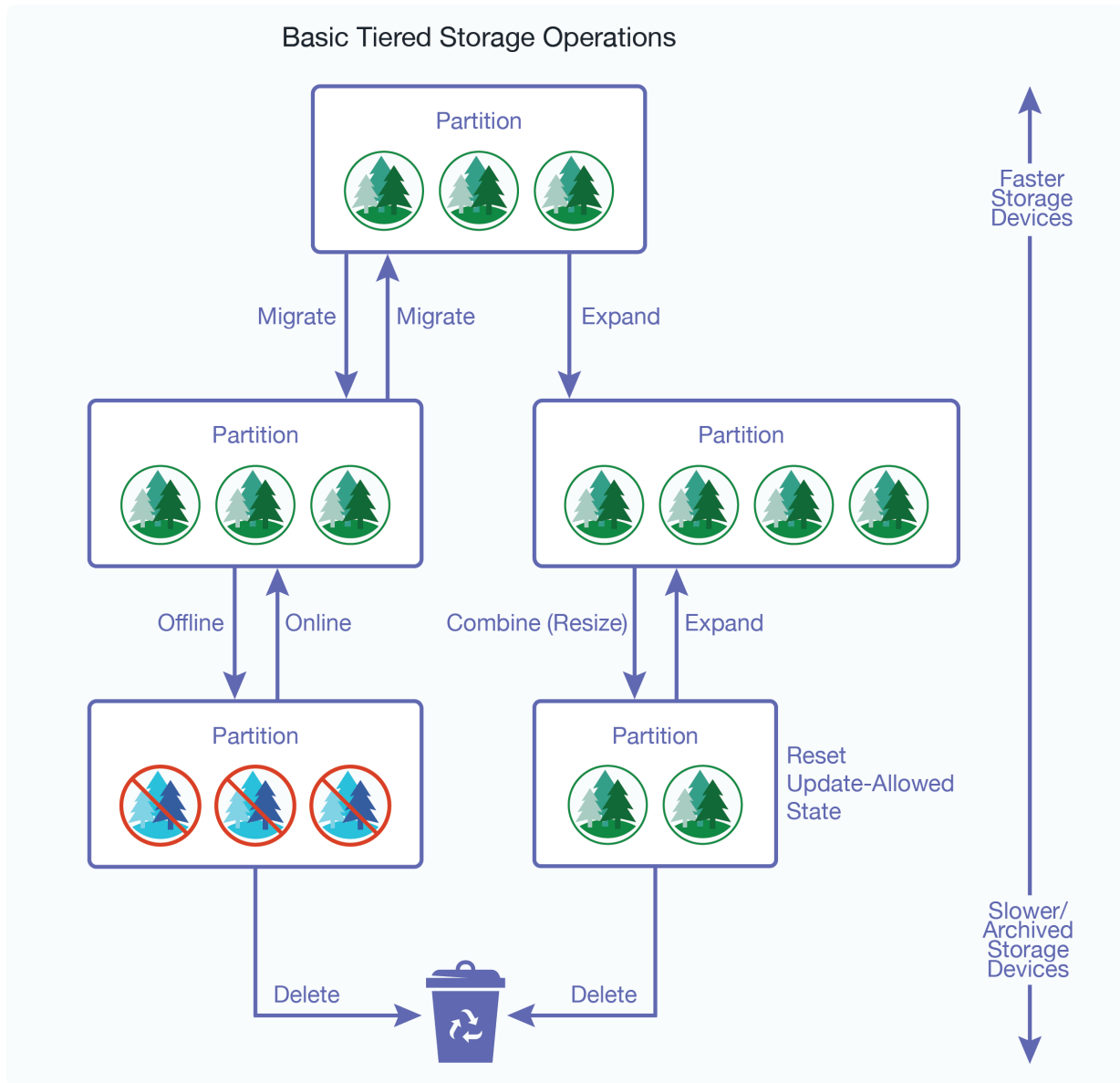
As data ages and becomes less updated and queried, it can be migrated to less expensive and more densely packed storage devices to make room for newer, more frequently accessed and updated data, as illustrated in the graph below:



The illustration below shows the basic tiered storage operations:

- Migrate a partition to a different database, host, and/or directory, which may be mounted on another storage device.
- Resize the partition to expand or contract the number of forests it contains.
- Combine a number of forests into a single forest.

- Reset the update-allowed state of a partition. For example, make the partition read-only, so it can be stored more compactly on a device that is not required to reserve space for forest merges.
- Take a partition offline to archive the partition. The partition data is unavailable to query, update, backup, restore and replicate operations.
- Take a partition online to make the partition data available again.
- Delete a partition when its data has outlived its useful life.



WARNING
 Forest migrate, forest combine, partition migrate, and partition resize may result in potential data loss when used during XA transactions.

There are two types of partitions:

- [Section 17.3, "Range Partitions" \[134\]](#)

- [Section 17.4, “Query Partitions” \[135\]](#)

17.3. Range Partitions

A range partition consists of a group of database forests that share the same name prefix and the same *range assignment policy* described in [Section 16.3.4, “Range Assignment Policy” \[122\]](#).



NOTE

When deploying forests in a cluster, you should align forests and forest replicas across hosts for parallelization and high availability, as described in the [Scalability, Availability, and Failover Guide](#).

The range of a partition defines the scope of element or attribute values for the documents to be stored in the partition. This element or attribute is called the *partition key*. The partition key is based on a range index, collection lexicon, or field set on the database. The partition key is set on the database and the partition range is set on the partition, so you can have several partitions in a database with different ranges.

For example, you have a database, named `WorkingVolumes`, that contains nine forests that are grouped into three partitions. Among the range indexes in the `WorkingVolumes` database is an element range index for the `update-date` element with a type of `date`. The `WorkingVolumes` database has its partition key set on the `update-date` range index. Each forest in the `WorkingVolumes` database contains a lower bound and upper bound range value of type `date` that defines which documents are to be stored in which forests, as shown in the following table:

Partition Name	Forest Name (<i>prefix-name</i>)	Partition Range Lower Bound	Partition Range Upper Bound	Lower Bound Included
Vol1	Vol1-0001 Vol1-0002	2010-01-01	2011-01-01	false
Vol2	Vol2-0001 Vol2-0002 Vol2-0003	2011-01-01	2012-01-01	false
Vol3	Vol3-0001 Vol3-0002 Vol3-0003 Vol3-0004	2012-01-01	2013-01-01	false

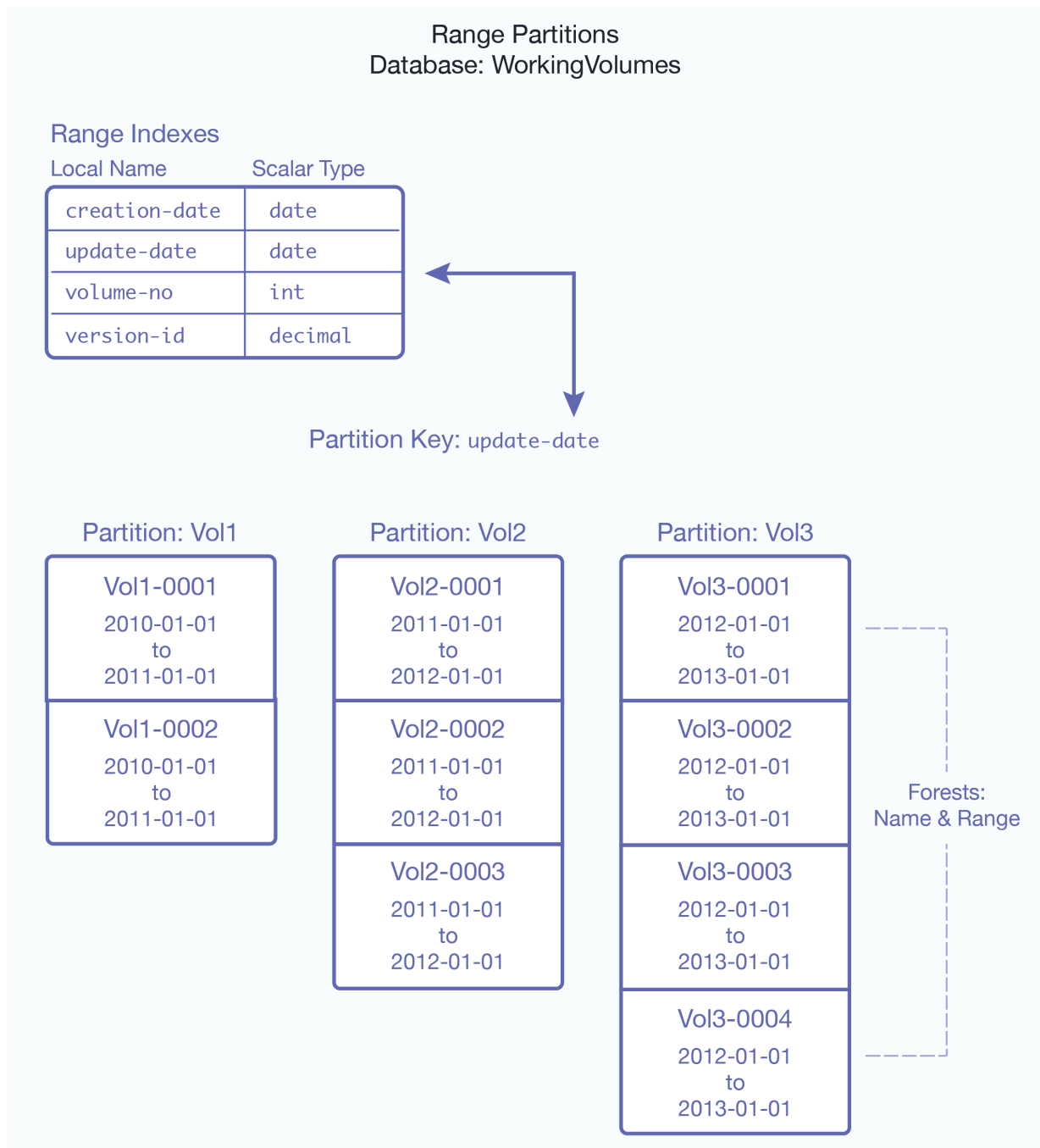


NOTE

When `Lower Bound Included` is set to `false` on a database, the lower bound of the partition ranges are ignored. With this setting, documents with a partition key value that match the lower bound value are excluded from the partition and documents that match the upper bound value are included.

In this example, a document with an `update-date` element value of `2011-05-22` would be stored in one of the forests in the `Vol2` partition. Should the `update-date` element value in the document get updated to `2012-01-02` or later, the document will be automatically moved to the `Vol3` partition. How the documents are redistributed among the partitions is handled by the database rebalancer, as described in [Section 16.3.4, “Range Assignment Policy” \[122\]](#).

Below is an illustration of the `WorkingVolumes` database, showing its range indexes, partition key, and its partitions and forests:



17.4. Query Partitions

A query partition consists of a group of database forests that share the same name prefix and the same *query assignment policy* described in [Section 16.3.5, “Query Assignment Policy” \[124\]](#).



NOTE

Query partitions query documents in an unfiltered manner. For details about unfiltered queries, see the [Fast Pagination and Unfiltered Searches](#) in the *Query Performance and Tuning Guide*.

Each query partition is associated with a query that determines which documents are stored in that partition. When creating a query partition, you assign it a partition number. Unlike range partitions, queries set for partitions using the query assignment policy can have “overlaps,” so that a document may be matched by the query set for more than one partition. In the event of an overlap, the partition with the lower number is selected over partitions with higher numbers.



NOTE

As is the case with range assignment policy, you should define a default partition when configuring the query assignment policy. If you do not define a default partition, the database forests that are not associated with a query partition are used.

For example, you have three query partitions, a default partition and two partitions associated with the following types of queries:

Query Partition 1: (Default -- no query)

Query Partition 2:

Requirement	Query Type
the author includes "Twain"	word
there is a paperback edition	value
the price of the paperback edition is less than \$9.00	range

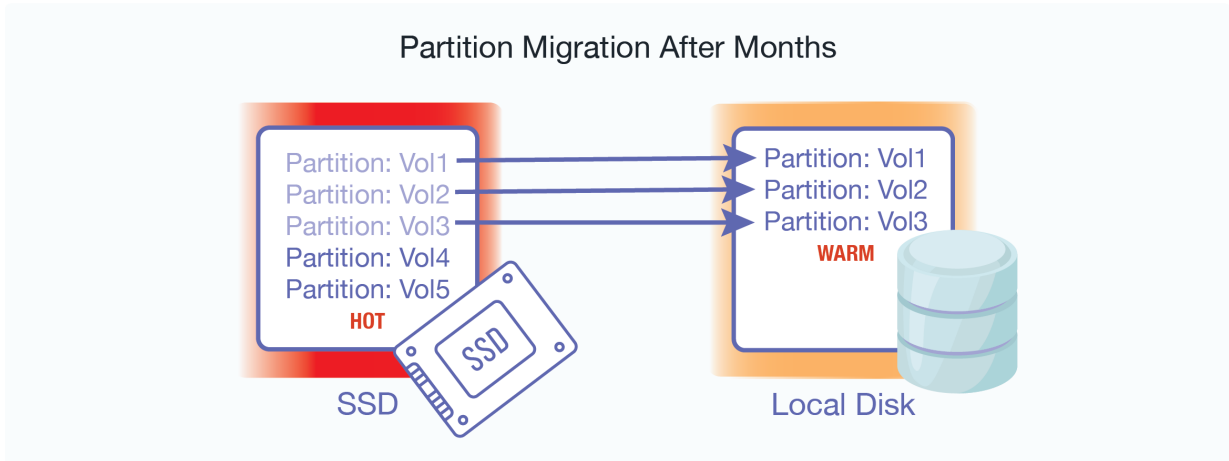
Query Partition 3:

Requirement	Query Type
the title includes "Adventures"	word
the characters include "Huck"	word
the class is "fiction"	word

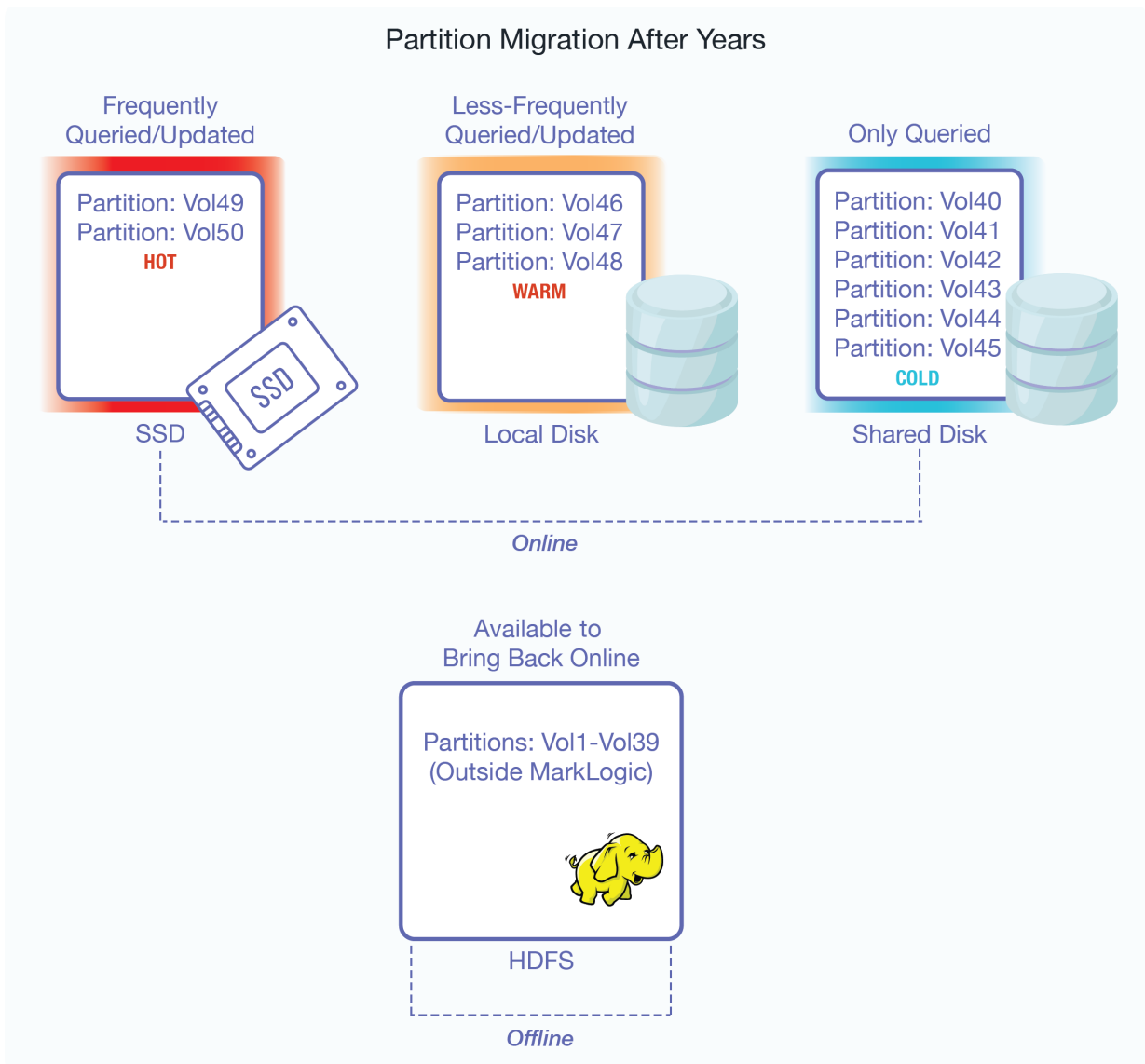
In this example, the document “Adventures of Huckleberry Finn” matches both queries, but is stored in Query Partition 2 because it is the partition with the lower number. On the other hand, the document “Moby Dick” does not match either query, so it is stored in Partition 1, the Default Query Partition.

17.5. Partition Migration

Both range and query partitions can be migrated between different types of storage. For example, you have the range partitions created in [Section 17.3, “Range Partitions” \[134\]](#) and, after a few months, the volumes of documents grow to 5 and there is no longer enough space on the fast SSD device to hold all of them. Instead, the oldest and least queried volumes (Vol1-Vol3) are migrated to a local disk drive, which represents a slower storage tier:

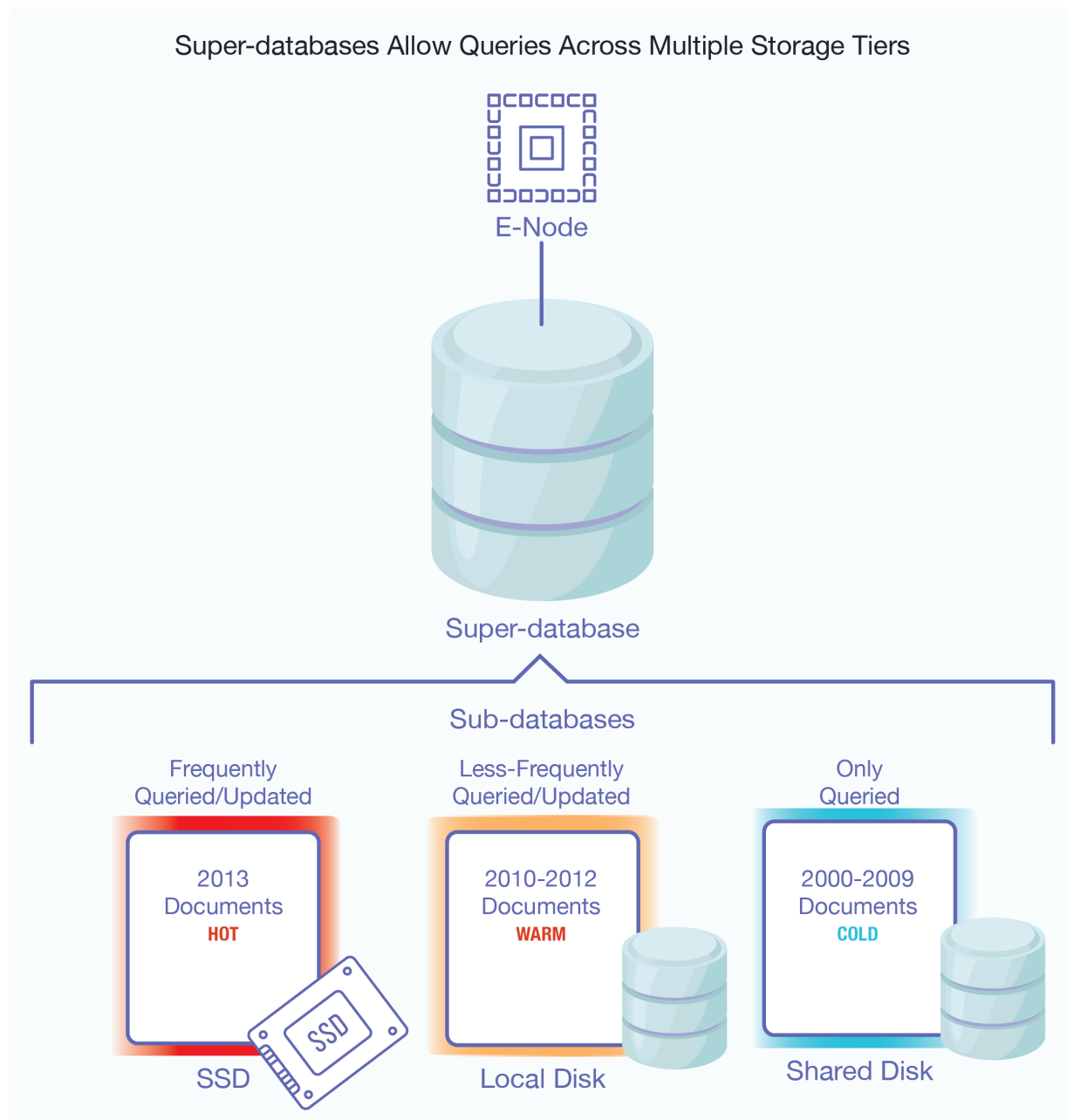


After years of data growth, the volumes of documents grow to 50. After migrating between storage tiers, the partitions are eventually distributed among the storage tiers:



Multiple databases, even those that serve on different storage tiers, can be grouped into a *super-database* in order to allow a single query to be done across multiple tiers of data. Databases that

belong to a super-database are referred to as *sub-databases*. A single sub-database can belong to multiple super-databases:



For details on super-databases and sub-databases, see [Section 17, “Tiered Storage” \[131\]](#).

17.6. Configuring a Database with Range Partitions

If a database is to participate in a tiered storage scheme using range partitions, it must have the following settings:

- Rebalancer enable set to `true`
- Rebalancer Assignment Policy set to `range`
- Locking set to `strict`
- A range index established for the partition key, as described in [Section 25, “Range Indexes and Lexicons” \[237\]](#)
- A partition key, as described in [Section 17.6.1, “Defining a Range Partition Key” \[139\]](#)

- Range partitions, as described in [Section 17.6.2, “Creating Range Partitions” \[140\]](#)



WARNING

All of the forests in a database configured for tiered storage using range partitions must be part of a partition.

For details on how to configure the database rebalancer with the range assignment policy, see the sections [Section 16.3.4, “Range Assignment Policy” \[122\]](#), [Section 16.5, “Configuring the Rebalancer on a Database” \[127\]](#), and [Section 16.6, “Configuring the Rebalancer on a Forest” \[127\]](#).

17.6.1. Defining a Range Partition Key

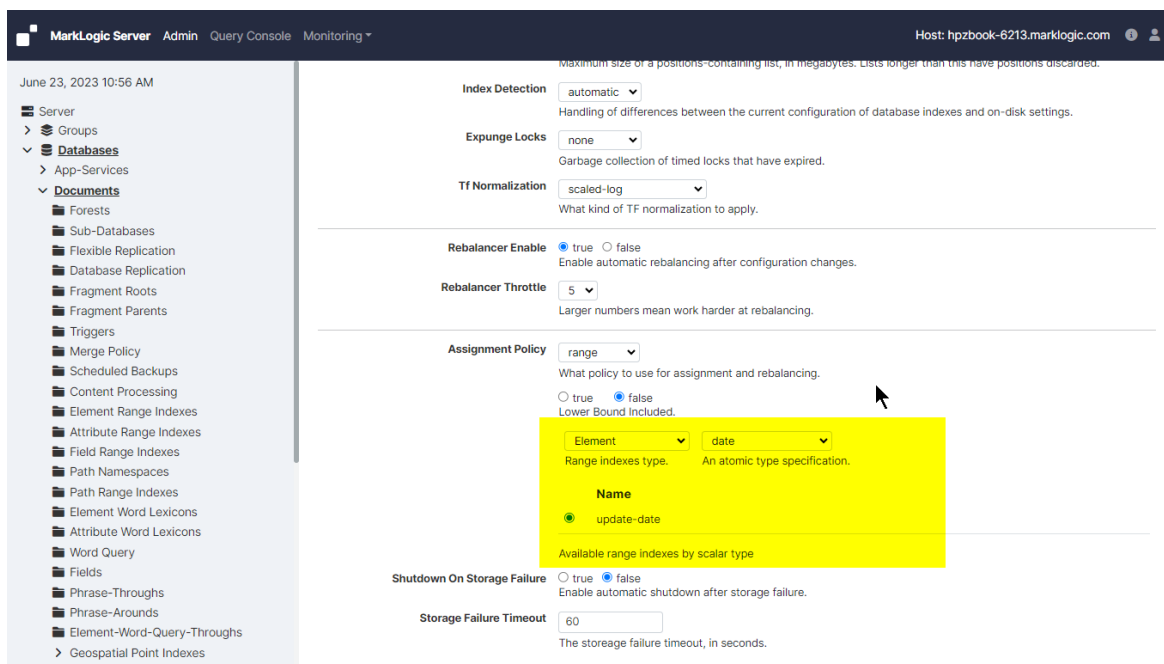
The partition key describes a common element or attribute in the stored documents. The value of this element or attribute in the document determines the partition in which the document is stored. A partition key is based on a range index, collection lexicon, or field of the same name set for the database. The range index, collection lexicon, or field used by the partition key must be created before the partition key is created.

For example, assume your documents all have an `update-date` element with a date value. The following procedure describes how to create a partition key for the `update-date` element:

1. Create an element range index, named `update-date`, on the database of type `date`. The details on how to create an element range index are described in [Section 25.6, “Defining Element Range Indexes” \[242\]](#).
2. Click **Databases** in the left tree menu. A list of databases appears.
3. Click your target database.
4. On the **Configure** tab, scroll down and change the **Assignment Policy** field to **range**. Additional fields appear under the assignment policy.
5. Choose whether the lower bound should be included in the assignment policy.
 - a. Set **Lower Bound Included** to **true** if you want to include documents with a partition key value that matches the lower bound value, and exclude documents that match the upper bound value.
 - b. Set **Lower Bound Included** to **false** if you want to exclude documents with a partition key value that matches the lower bound value, and include documents that match the upper bound value.

For example, if the range is `2011-01-01` (lower) to `2012-01-01` (upper) and **Lower Bound Included** is set to **false**, documents with an `update-date` value of `2011-01-01` will not be included in the partition, but documents with an `update-date` value of `2011-01-02` and `2012-01-01` will be included.

6. Set the **Range indexes type** and **atomic type specification** (scalar type) of the range index, field, or collection lexicon you want to use as your partition key. In this example, the range indexes type is **Element** and the atomic type specification (scalar type) is **date**.
7. The range indexes, fields, or collection lexicons available are listed under the **Name** column. Select the value to use as your partition key (`update-date` is selected in this example).



17.6.2. Creating Range Partitions

Range partitions are based on forest naming conventions. A forest’s partition name prefix and the rest of the forest name are separated by a dash (-). For example, a forest named `June-0001` belongs to the `June` partition.



NOTE

It is a best practice to create a default partition (a partition without a range) before creating partitions with ranges. Doing this will allow you to load documents into the default partition before you have finished creating the other partitions. As new partitions with ranges are created, the documents will be automatically moved from the default partition to the partitions with matching ranges.



WARNING

All of the forests in a database configured for tiered storage must be part of a partition.

The two ways to create a range partition are covered in this section.

Creating a Range Partition with New Forests

You can use the `POST: /manage/v2/databases/{id|name}/partitions` REST resource address to create a new range partition with empty forests. When creating a range partition, you specify the partition range and the number of forests to be created for the partition. You can also specify that the range partition be created for multiple hosts in a cluster, in which case the specified number of forests will be created on each host.

For example, the following creates a range partition, named `2011`, in the `Documents` database on hosts, `MyHost1` and `MyHost2`, with a range of `2011-01-01 - 2012-01-01` and four empty forests,

named 2011-0001, 2011-0002, 2011-0003, and 2011-0004, on MyHost1 and four empty forests, named 2011-0005, 2011-0006, 2011-0007, and 2011-0008, on MyHost2:

```
$ cat create-partition.xml
<partition xmlns="http://marklogic.com/manage">
  <partition-name>2011</partition-name>
  <upper-bound>2012-01-01</upper-bound>
  <lower-bound>2011-01-01</lower-bound>
  <forests-per-host>4</forests-per-host>
  <hosts>
    <host>MyHost1</host>
    <host>MyHost2</host>
  </hosts>
</partition>
```

```
$ curl --anyauth --user user:password -X POST \
-d @create-partition.xml -H 'Content-type: application/xml' \
http://MyHost:8002/manage/v2/databases/Documents/partitions
```

You can also include an options element to create replica forests for shared-disk or local-disk failover. For details, see [Section 17.10, “Partitions with Forest-Level Failover” \[154\]](#).

Creating a Range Partition from Existing Forests

You can create a range partition from existing forests simply by renaming the forests so that they adhere to a range partition naming convention. For example, you have four forests, named 1-2011, 2-2011, 3-2011, and 4-2011. You can make these four forests into a range partition, named 2011, by renaming 1-2011 to 2011-1, and so on. You should also specify a common range for each renamed forest, or leave the range fields blank to identify the forests as belonging to a default range partition. Default range partitions store the documents that have partition key values that do not fit into any of the ranges set for the other range partitions.

For example, to rename the 1-2011 forest to 2011-1 and set the range to 2011-01-01 - 2012-01-01, follow these steps:

1. Click **Forests** in the left tree menu. A list of forests appears.
2. Click your target forest.
3. In the **Forest Name** field, change the name from 1-2011 to 2011-1:
4. In the **Range** section, set the lower bound value to 2011-01-01 and the upper bound value to 2012-01-01:
5. Click **OK**.



NOTE

You can also accomplish this operation using the XQuery, JavaScript, and REST APIs. For example, in XQuery using the `admin:forest-rename` and `admin:forest-set-range-policy-range` functions.

17.7. Configuring a Database with Query Partitions

If a database is to participate in a tiered storage scheme using query partitions, it must have these settings:

- Rebalancer enable set to `true`
- Rebalancer Assignment Policy set to `query`
- Locking set to `strict`

- Indexes established for the elements or properties to be queried
- Query partitions, as described in [Section 17.7.1, “Creating Query Partitions” \[142\]](#)

**NOTE**

Unlike range partitions, it is not necessary for all of the forests in a database configured for tiered storage to be part of a query partition.

For details on the database rebalancer with the query assignment policy, see the sections [Section 16.3.5, “Query Assignment Policy” \[124\]](#), [Section 16.5, “Configuring the Rebalancer on a Database” \[127\]](#), and [Section 16.6, “Configuring the Rebalancer on a Forest” \[127\]](#).

To configure a database to use the query assignment policy, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. On the configuration tab for the database, set the **Assignment Policy to query**. The **Default Partition** field appears.
4. [OPTIONAL] Enter the partition number for a default query partition in the **Default Partition** field. If you do not define a default query partition, then your database must have forests that are not part of a query partition. These forests will serve the same role as a default partition.

17.7.1. Creating Query Partitions

When creating a query partition, you specify the query partition name, number, and the number of forests to be created for the query partition. You can also specify that the query partition be created for multiple hosts in a cluster, in which case the specified number of forests will be created on each host.

Query partitions are based on forest naming conventions. A forest’s partition name prefix and the rest of the forest name are separated by a dash (-). For example, a forest named `tier1-0001` belongs to the `tier1` partition. Unlike range partitions, it is not necessary for all of the forests in a database configured for tiered storage to be part of a query partition.

**NOTE**

It is a best practice to create a default query partition (a partition without a query). Doing this will allow you to load documents into the default partition before you have finished creating the other partitions. As new partitions with queries are created, the documents will be automatically moved from the default partition to the query partitions with matching queries.

For details on how to configure the database rebalancer with the query assignment policy, see the sections [Section 16.3.5, “Query Assignment Policy” \[124\]](#), [Section 16.5, “Configuring the Rebalancer on a Database” \[127\]](#), and [Section 16.6, “Configuring the Rebalancer on a Forest” \[127\]](#).

Query partitions do unfiltered searches, which means that the results are not filtered for validation. For details about unfiltered queries, see the [Fast Pagination and Unfiltered Searches](#) in the *Query Performance and Tuning Guide*.

For example, the following command creates query partition number 1, named `tier1`, with two forests in the `Documents` database on the host, `MyHost1`:

```
curl -X POST --anyauth --user admin:admin \
-H "Content-type: application/json" \
-d '{
  "partition-name": "tier1",
  "partition-number": "1",
  "forests-per-host": 2,
  "host": [ "MyHost1" ],
  "option": [ "failover=none" ]
}' \
http://MyHost1:8002/manage/v2/databases/Documents/partitions
```

17.7.2. Setting the Query Assignment Policy for the Query Partition

After creating a query partition, you can use the `POST:/manage/v2/databases/{id|name}/partition-queries` REST resource address to assign to it a query assignment policy, as described in [Section 16.3.5, “Query Assignment Policy” \[124\]](#).



NOTE

Any indexes required for the query must be created before creating the query partition.

A query assignment policy in XML takes the form:

```
<partition-query-properties xmlns="http://marklogic.com/manage/partition-query/
properties">
  <partition-number>1</partition-number>
  <query>
    ....cts:query.....
  </query>
</partition-query-properties>
```

A query assignment policy in JSON takes the form:

```
{
  "partition-number": "1",
  "query": {
    ....cts:query.....
  }
}
```

The search portion is a `cts:query` expression, as described in [Composing cts:query Expressions](#) in the *Search Developer's Guide*. There can be only one `cts:query` per partition.

The query requires the proper index to be configured in the database. The complexity of the query affects the performance of insert and rebalancing. Therefore slow query like wildcard matching is not recommended.

For example to direct all documents that have either the word “Manager” or “Engineer” in them to the `tier1` query partition created above, you would do the following:

```
$ cat query1.xml
<partition-query-properties xmlns="http://marklogic.com/manage/partition-query/
properties">
  <partition-number>1</partition-number>
  <query>
    <cts:or-query xmlns:cts="http://marklogic.com/cts">
      <cts:word-query>
        <cts:text xml:lang="en">Manager</cts:text>
      </cts:word-query>
      <cts:word-query>
        <cts:text xml:lang="en">Engineer</cts:text>
      </cts:word-query>
    </cts:or-query>
  </query>
</partition-query-properties>
```

```
curl -X POST --anyauth -u admin:admin \
-H "Content-Type:application/xml" -d @query1.xml \
http://gordon-1:8002/manage/v2/databases/Schemas/partition-queries
```

The following query assignment policy will match documents where "LastModified" is within the last year:

```
<partition-query-properties xmlns="http://marklogic.com/manage/partition-query/
properties">
  <partition-number>1</partition-number>
  <query>
    <cts:element-range-query operator="&gt;=" xmlns:cts="http://marklogic.com/cts">
      <cts:element>LastModified</cts:element>
      <cts:value type="xs:yearMonthDuration">PLY</cts:value>
    </cts:element-range-query>
  </query>
</partition-query-properties>
```

The same query assignment policy in JSON:

```
{
  "partition-number": 1,
  "query": {
    "element-range-query": {
      "operator": ">=",
      "element": "LastModified",
      "value": {
        "type": "xs:yearMonthDuration",
        "val": "PLY"
      }
    }
  }
}
```

For queries against a `dateTime` index, when `$value` is an `xs:dayTimeDuration` or `xs:yearMonthDuration`, the query is executed as an age query. `$value` is subtracted from `fn:current-dateTime()` to create a `xs:dateTime` used in the query. If there is more than one item in `$value`, they must all be the same type.

For example, given a `dateTime` index on element `startDateTime`, queries `cts:element-range-query(xs:QName ("startDateTime"), ">", xs:dayTimeDuration("P1D"))` and `cts:element-range-query(xs:QName ("startDateTime"), ">", fn:current-dateTime() - xs:dayTimeDuration("P1D"))` are the same: both match values within the last day.

17.7.3. Isolating a Query Partition

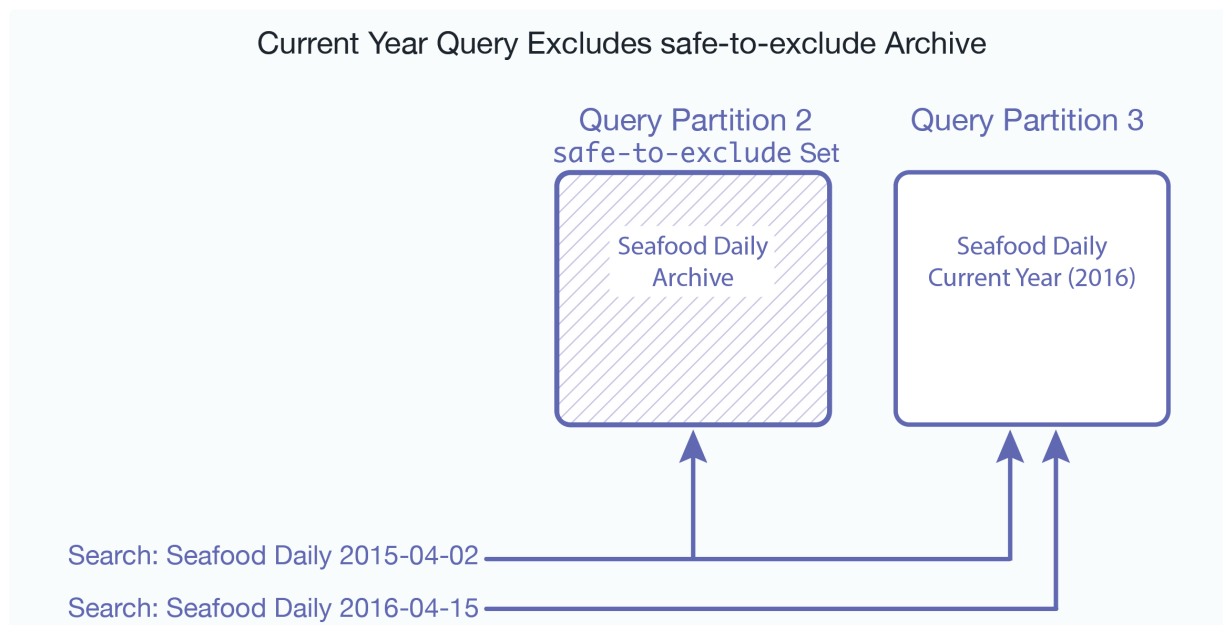
By default, when a search query is given to MarkLogic, all query partitions are searched, regardless of the query assignment policy set on the partition. To avoid this overhead, you can use the

`tieredstorage:partition-set-exclusion-enabled` function to set `safe-to-exclude` on the query partition so that it will not be searched if the search query does not match the query assignment policy set for that partition.

When documents are distributed in query partitions based on time and date, you may want the option to always search a particular tier (typically the tier holding the most recent documents) because it may be the case that some of the documents in that tier are about to be migrated to a different tier but have not yet been moved. So if a search only matches the query set in a “lower” tier, the non-matching “higher” tier will also be searched to locate the matching documents that have not yet moved to the lower tier.

For example, you have two query partitions that hold the documents, “Seafood Daily,” as shown below. The query assignment policy for each compares the date of the document with the current date and sorts the documents so that one partition contains the issues from the current year and the other archives the issues from previous years. The query partition serving as the archive is set to `safe-to-exclude` and the query partition containing this year’s issues is not set with this option.

The current year is 2016 and a search query is given that matches the query for Archive Partition will also result in a search on Current Year Partition. However, a search query that matches the Current Year Partition will exclude the Archive Partition.



17.7.4. Look Up Partitions Queries

To look up partitions queries, follow these steps:

1. Use the Management REST API via `/manage/v2/databases/{id|name}/partition-queries`.
2. Look in the associated schemas database for the partition queries with `tieredstorage:partition-queries`.

17.8. Overview of the Tiered Storage REST API

Tiered storage is supported by the XQuery, JavaScript, and REST APIs. All of the operations you will want to integrate into your storage-management scripts to automate repetitive storage management operations are available through the REST API. However, some of the initial, one-time, setup operations, such as those related to setting the range policy and partition key on the database, are only supported by the Admin Interface and the XQuery API.

**NOTE**

The Tiered Storage REST API supports both JSON and XML formats. The XML format is used for all of the examples in this section.

17.8.1. Asynchronous Operations

The partition resize and migrate, as well as the forest migrate and combine operations are processed asynchronously. This is because these operations may move a lot of data and take more time than generally considered reasonable for control to return to your script. Such asynchronous operations are tracked reusing ticket endpoints. This asynchronous process is initiated by `GET: /manage/v2/tickets/{tid}?view=process-status`, as outlined in the following steps:

The generated ticket is returned in this form:

```
/manage/v2/tickets/{id}?view=process-status.
```

You can view the status of the operation by visiting the URL. For example, if the returned ticket is

```
/manage/v2/tickets/8681809991198462214?view=process-status
```

and your host is `MyHost`, then you can view the status of your operation using this URL:

```
http://MyHost:8002/manage/v2/tickets/8681809991198462214?view=process-status.
```

**NOTE**

Historical ticket information can always be accessed by viewing the ticket default view.

17.8.2. Privileges

The following privileges are required for the resource addresses described in this section:

- GET operations require the `manage-user` privilege.
- PUT, POST, and DELETE operations require the `manage-admin` privilege.

17.8.3. /manage/v2/databases/{id|name}/partitions

Method	Description	Parameters	XQuery Equivalent
GET	Gets a list of partitions on the database	format? (json xml)	<code>tieredstorage:database-partitions</code>
POST	Add a range or query partition to the database	format? (json xml)	<code>tieredstorage:range-partition-create</code> <code>tieredstorage:query-partition-create</code>

For examples, see these topics:

- [Section 17.9.1, “Viewing Partitions” \[149\]](#)
- [Section 17.6.2, “Creating Range Partitions” \[140\]](#)
- [Section 17.7.1, “Creating Query Partitions” \[142\]](#)

17.8.4. /manage/v2/databases/{id|name}/partitions/{name}

Method	Description	Parameters	XQuery Equivalent
GET	Gets a summary of the partition, including links to containing database, links to member forests, and link to configuration	format? (json xml)	<code>tieredstorage:partition-forests</code>
DELETE	Deletes the partition	delete-data? (true false)	<code>tieredstorage:partition-delete</code>
PUT	Invokes one of the following operations on the partition: <ul style="list-style-type: none"> • resize (asynchronous) • transfer (synchronous) • migrate (asynchronous) 	format? (json xml)	<code>tieredstorage:partition-resize</code> <code>tieredstorage:partition-transfer</code> <code>tieredstorage:partition-migrate</code>

For examples, see these topics:

- [Section 17.9.9, “Deleting Partitions” \[154\]](#)
- [Section 17.9.3, “Resizing Partitions” \[151\]](#)
- [Section 17.9.4, “Transferring Partitions between Databases” \[152\]](#)
- [Section 17.9.2, “Migrating Forests and Partitions” \[150\]](#)

17.8.5. /manage/v2/databases/{id|name}/partitions/{name}/properties

Method	Description	Parameters	XQuery Equivalent
GET	Gets the partition properties (enabled, updates-allowed)	format? (json xml)	
PUT	Modifies the partition properties (updates-allowed, online offline)	format? (json xml)	<code>tieredstorage:partition-set-availability</code> <code>tieredstorage:partition-set-updates-allowed</code>

For examples, see these topics:

- [Section 17.9.7, “Taking Forests and Partitions Online and Offline” \[153\]](#)
- [Section 17.9.8, “Setting the updates-allowed State on Partitions” \[153\]](#)

17.8.6. /manage/v2/databases/{id|name}/partition-queries

Method	Description	Parameters	XQuery Equivalent
GET	Gets the query assignment policies for the query partitions set for the specified database.	format? (json xml)	<code>tieredstorage:partition-queries</code>
POST	Sets the query assignment policy for a query partition.		<code>tieredstorage:partition-set-query</code>

For examples, see this topic:

- [Section 17.7.2, “Setting the Query Assignment Policy for the Query Partition” \[143\]](#)

17.8.7. /manage/v2/databases/{id|name}/partition-queries/{partition-number}

Method	Description	Parameters	XQuery Equivalent
GET	Gets the query assignment policy of the query partition with the specified number.	format? (json xml)	tieredstorage:partition-get-query
DELETE	Deletes the query assignment policy for the query partition with the specified number.		tieredstorage:partition-delete-query

17.8.8. /manage/v2/databases/{id|name}/partition-queries/{partition-number}/properties

Method	Description	Parameters	XQuery Equivalent
GET	Gets the properties of the query for the query partition with the specified number.	format? (json xml)	tieredstorage:partition-get-query
PUT	Update the query assignment policy in the query partition with the specified number.	format? (json xml)	tieredstorage:partition-set-query

17.8.9. /manage/v2/forests

Method	Description	Parameters	XQuery Equivalent
GET	Gets a summary and list of forests.	format? (json xml) view database-id group-id host-id fullrefs	admin:get-forest-ids xdmp:forests
POST	Creates new forest(s)	format? (json xml)	admin:forest-create
PUT	Invokes one of the following operations on the forest: <ul style="list-style-type: none"> forest-combine forest-migrate <p>These operations are asynchronous</p>	format? (json xml)	tieredstorage:forest-combine tieredstorage:forest-migrate

For examples, see these topics:

- [Section 17.9.2, “Migrating Forests and Partitions” \[150\]](#)
- [Section 17.9.5, “Combining Forests” \[152\]](#)

17.8.10. /manage/v2/forests/{id|name}

Method	Description	Parameters	XQuery Equivalent
GET	Gets a summary of the forest.	format? (json xml) view	admin:forest-get-*

Method	Description	Parameters	XQuery Equivalent
POST	Initiates a state change on the forest.	state (clear merge restart attach detach retire employ)	xdmp:forest-clear xdmp:merge xdmp:forest-restart admin:database-attach-forest admin:database-detach-forest admin:database-retire-forest admin:database-employ-forest
DELETE	Deletes the forest.	level (config-only full)	admin:forest-delete

For an example, see this topic:

- [Section 17.9.6, “Retiring Forests” \[152\]](#)

17.8.11. /manage/v2/forests/{id|name}/properties

Method	Description	Parameters	XQuery Equivalent
GET	Gets the properties on the forest	format? (json xml)	admin:forest-get-enabled admin:forest-get-rebalancer-enable admin:forest-get-updates-allowed admin:database-get-attached-forests admin:forest-get-failover-enable admin:forest-get-availability
PUT	Initiates a properties change on the forest. The properties are: enable disable forest enable disable rebalancer modify updates-allowed specify failover hosts or replica forests availability	format? (json xml)	admin:forest-set-enabled admin:forest-set-rebalancer-enable admin:forest-set-updates-allowed admin:database-attach-forest admin:database-detach-forest admin:forest-set-failover-enable admin:forest-set-availability

17.9. Common Forest and Partition Operations

This section describes common partition operations. Some of these operations occur asynchronously. They immediately return a ticket number that you can use to check the status of the operation. For example, if the following data is returned,

```
<link><kindref>process-status</kindref><uriref>/manage/v2/tickets/4678516920057381194?view=process-status</uriref></link>
```

then you can check the status of the operation by entering a resource address like `http://MyHost:8002/manage/v2/tickets/4678516920057381194?view=process-status`.

For details on asynchronous processes, see [Section 17.8.1, “Asynchronous Operations” \[146\]](#).

17.9.1. Viewing Partitions

You can return all of the information on a partition.

For example, to return the details of the 2011 range partition on the `Documents` database, use the following command:

```
curl -X GET --anyauth --user admin:admin --header \
"Content-Type:application/xml" \
http://MyHost:8002/manage/v2/databases/Documents/partitions/2011
```

17.9.2. Migrating Forests and Partitions

Forests and partitions can be migrated from one storage device to another. For example, a range partition on an SSD has aged to the point where it is less frequently queried and can be moved to a slower, less expensive, storage device to make room for a more frequently queried range partition.

For example, the 2011 range partition on the `Documents` database is mounted on a local disk on the host, `MyHost`. To migrate the 2011 range partition to the `/warm-storage` data directory mounted on a shared disk on the host, `OurHost`, use these scripts:

```
$ cat migrate-partition.xml
<migrate xmlns="http://marklogic.com/manage">
  <hosts>
    <host>OurHost</host>
  </hosts>
  <data-directory>/warm-storage</data-directory>
  <options>
    <option>failover=none</option>
    <option>local-to-shared</option>
  </options>
</migrate>
```

```
$ curl --anyauth --user user:password -X PUT \
-d @migrate-partition.xml -H 'Content-type: application/xml' \
http://MyHost:8002/manage/v2/databases/Documents/partitions/2011
```



NOTE

If you do not specify a `data-directory`, the default data directory is used.

The tiered storage migration operations allow you to migrate a forest or partition between different types of storage. The following table lists the four migration options. The migration option you select determines the sequence of steps taken by tiered storage during the migration operation.

Migration Option	Description
local-to-local (default)	Indicates that the migration is to move data from local storage to local storage. This is the default if no migration option is specified and the type of storage cannot be derived from the data directory path.
local-to-shared	Indicates that the migration is to move data from local storage to shared storage. This type of migration supports changing hosts.
shared-to-local	Indicates that the migration is to move data from shared storage to local storage. This type of migration supports changing hosts.
shared-to-shared	Indicates that the migration is to move data from shared storage to shared storage. This type of migration supports changing hosts.

You can use the `PUT:/manage/v2/forests` resource address to migrate individual forests. For example, the forests `2011-0001` and `2011-0002`, are mounted on a local disk on the host, `MyHost`. To migrate these forests to the `/warm-storage` data directory mounted on a shared disk on the host, `OurHost`, use these scripts:

```
$ cat migrate-forests.xml
<forest-migrate xmlns="http://marklogic.com/manage">
  <forests>
    <forest>2011-0001</forest>
    <forest>2011-0002</forest>
  </forests>
  <host>MyHost</host>
  <data-directory>/warm-storage</data-directory>
  <options>
    <option>local-to-shared</option>
  </options>
</forest-migrate>
```

```
$ curl --anyauth --user user:password -X PUT \
-d @migrate-forests.xml -H 'Content-type: application/xml' \
http://MyHost:8002/manage/v2/forests
```

**NOTE**

If failover is configured on your forests, do a full backup of database after a forest or partition migrate operation to ensure that you can recover your data should something go wrong. You may also need to increase the timeout setting on the migrate operation, as it will take longer when failover is configured.

17.9.3. Resizing Partitions

You can increase or decrease the number of forests in a partition. Once the resize operation has completed, the documents in the partition forests will be rebalanced for even distribution.

For example, to resize the 2011 range partition up to five forests, use this code:

```
$ cat resize-partition.xml
<resize xmlns="http://marklogic.com/manage">
  <forests-per-host>5</forests-per-host>
  <hosts>
    <host>MyHost</host>
  </hosts>
</resize>
```

```
$ curl --anyauth --user user:password -X PUT \
-d @resize-partition.xml -H 'Content-type: application/xml' \
http://MyHost:8002/manage/v2/databases/Documents/partitions/2011
```

In addition to resizing your partition, you can migrate your partition to another host by specifying a different host in the payload. Additionally, you can move the partition to a different storage tier (such as local-to-shared) by specifying one of the migration options described in [Section 17.9.2, “Migrating Forests and Partitions” \[150\]](#).

**NOTE**

If you resize partitions for databases configured for database replication, first resize the replica partitions before resizing the master partitions.

17.9.4. Transferring Partitions between Databases

You can move a partition from one database to another. For example, to transfer the 2011 range partition from the DB1 database to the DB2 database, use this code:

```
$ cat transfer-partition.xml
<transfer xmlns="http://marklogic.com/manage">
  <destination-database>DB2</destination-database>
</transfer>

$ curl --anyauth --user user:password -X PUT \
-d @transfer-partition.xml -H 'Content-type: application/xml' \
http://MyHost:8002/manage/v2/databases/DB1/partitions/2011
```

17.9.5. Combining Forests

You can use the PUT: /manage/v2/forests resource address to combine multiple forests into a single forest. For example, to combine the forests, 2011-0001 and 2011-0002, into a single forest named 2011, follow these steps:

```
$ cat combine-forests.xml
<forest-combine xmlns="http://marklogic.com/manage">
  <forests>
    <forest>2011-0001</forest>
    <forest>2011-0002</forest>
  </forests>
  <forest-name>2011</forest-name>
  <hosts>
    <host>MyHost</host>
  </hosts>
</forest-combine>

$ curl --anyauth --user user:password -X PUT \
-d @combine-forests.xml -H 'Content-type: application/xml' \
http://MyHost:8002/manage/v2/forests
```

You can both combine forests and migrate the combined forest to another host in a single operation by specifying a different host value. You can also move the forests to a different storage tier (such as local-to-shared) by specifying one of the migration options described in [Section 17.9.2, “Migrating Forests and Partitions” \[150\]](#).



NOTE

If you want to combine forests that are attached to databases configured for database replication, first combine the foreign replica forests with the `snapshot` option before combining the master forests.

If failover is configured on your forests, do a full backup of database after a forest combine operation to ensure that you can recover your data should something go wrong. You may also need to increase the timeout setting on the combine operation, as it will take longer when failover is configured.

17.9.6. Retiring Forests

You can “retire” a forest from a database in order to move all of its documents to the other forests and rebalance them among those forests, as described in [Section 16.4.2, “How Data is Moved When a Forest is Retired from the Database” \[127\]](#).

This code shows how to retire the forest, 2011, from the Documents database,

```
curl -i -X POST --digest --user user:password -H \
"Content-Type:application/x-www-form-urlencoded" \
--data "state=retire&database=Documents" \
http://MyHost:8002/manage/v2/forests/2011
```

17.9.7. Taking Forests and Partitions Online and Offline

You can take a forest or partition offline and store it in an archive, so that it is available to later bring back online, if necessary. The benefit of taking data offline is to spare the RAM, CPU, and network resources for the online data.

An offline forest or partition is excluded from query, update, backup, restore and replicate operations performed by the database to which it is attached. An offline forest or partition can be attached, detached, or deleted. Operations, such as rename, forest-level backup and restore, migrate, and combine are not supported on an offline forest or partition. If a forest is configured with failover, the replica forest inherits the online/offline setting of its master forest, so disabling an offline master forest does not trigger a failover.


For example, to take the 2011 range partition in the DB2 database offline, do the following:

```
$ cat partition-offline.xml
<partition-properties xmlns="http://marklogic.com/manage">
  <availability>offline</availability>
</partition-properties>
```

```
$ curl --anyauth --user user:password -X PUT \
-d @partition-offline.xml -H 'Content-type: application/xml' \
http://MyHost:8002/manage/v2/databases/DB2/partitions/2011/properties
```

17.9.8. Setting the updates-allowed State on Partitions

You can change the updates-allowed state of a partition to make its forests. The possible states are shown in the table below.

State	Description
all	Read, insert, update, and delete operations are allowed on the partition.
delete-only	Read and delete operations are allowed on the partition, but insert and update operations are not allowed.
read-only	Read operations are allowed on the partition, but insert, update, and delete operations are not allowed. A transaction attempting to make changes to fragments in the partition will throw an exception. <div data-bbox="422 1440 1385 1581" style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;">  <p>NOTE Resizing a read-only partition to fewer forests preserves its original forests.</p> </div>
flash-backup	Puts the partition in read-only mode without throwing exceptions on insert, update, or delete transactions, allowing the transactions to retry.

For example, to set the updates-allowed state in the 2011 range partition in the Documents database to read-only, do the following:

```
$ cat read-only-partition.xml
<partition-properties xmlns="http://marklogic.com/manage">
  <updates-allowed>read-only</updates-allowed>
</partition-properties>
```

```
$ curl --anyauth --user user:password -X PUT \
-d @read-only-partition.xml -H 'Content-type: application/xml' \
http://MyHost:8002/manage/v2/databases/Documents/partitions/2011/properties
```

17.9.9. Deleting Partitions

You can delete a partition, along with all its forests. For example, to delete the 2011 range partition from the `Documents` database, enter this command:

```
$ curl --anyauth --user user:password -X DELETE \
-H 'Content-type: application/xml' \
http://MyHost:8002/manage/v2/databases/Documents/partitions/2011
```

17.10. Partitions with Forest-Level Failover

The partition create, migrate and resize operations allow you to specify an `options` element to create replica forests for shared-disk or local-disk failover, as described in the [Configuring Local-Disk Failover for a Forest](#) and [Configuring Shared-Disk Failover for a Forest](#) in the *Scalability, Availability, and Failover Guide*.

To create replica forests for forest-level failover, you must create the partition on at least two hosts. For each master forest created on one host a replica forest will be created on another host. For example, to create a single replica forest for each forest in the 2011 range partition and configure the forests for local-disk failover between `MyHost1`, `MyHost2`, and `MyHost3`, use these scripts:

```
$ cat create-partition.xml
<partition xmlns="http://marklogic.com/manage">
  <partition-name>2011</partition-name>
  <upper-bound>2012-01-01</upper-bound>
  <lower-bound>2011-01-01</lower-bound>
  <forests-per-host>4</forests-per-host>
  <data-directory>/forests</data-directory>
  <hosts>
    <host>MyHost1</host>
    <host>MyHost2</host>
    <host>MyHost3</host>
  </hosts>
  <data-directory></data-directory>
  <large-data-directory></large-data-directory>
  <fast-data-directory></fast-data-directory>
  <options>
    <option>replicas=1</option>
    <option>failover=local</option>
  </options>
</partition>
```

```
$ curl --anyauth --user user:password -X POST \
-d @create-partition.xml -H 'Content-type: application/xml' \
http://MyHost:8002/manage/v2/databases/Documents/partitions
```

Keep in mind the following details when configuring partitions or forests with forest-level failover:

- If failover is configured on your forests, do a full backup of database after doing a partition or forest migrate or a forest combine to ensure that you can recover your data should something go wrong. You may also need to increase the timeout setting on the migrate or combine operation, as these operations will take longer when failover is configured.
- It is not recommended to configure local-disk failover for forests attached to a database with journaling set to `off`.
- You cannot configure a partition with shared-disk or local-disk failover on Amazon Simple Storage Service (S3), unless its fast data directory, as designated by `<fast-data-directory>`, is not on S3.
- If your deployment of MarkLogic is on Amazon Elastic Compute Cloud (EC2) or is distributed across multiple data centers, be sure to specify an equal number of hosts on different zones when creating, migrating, or resizing your partition with forest-level failover. For example, two hosts on `us-east-1a`, two hosts on `us-east-1b`, and two hosts on `us-east-1c`. In this example, tiered storage will

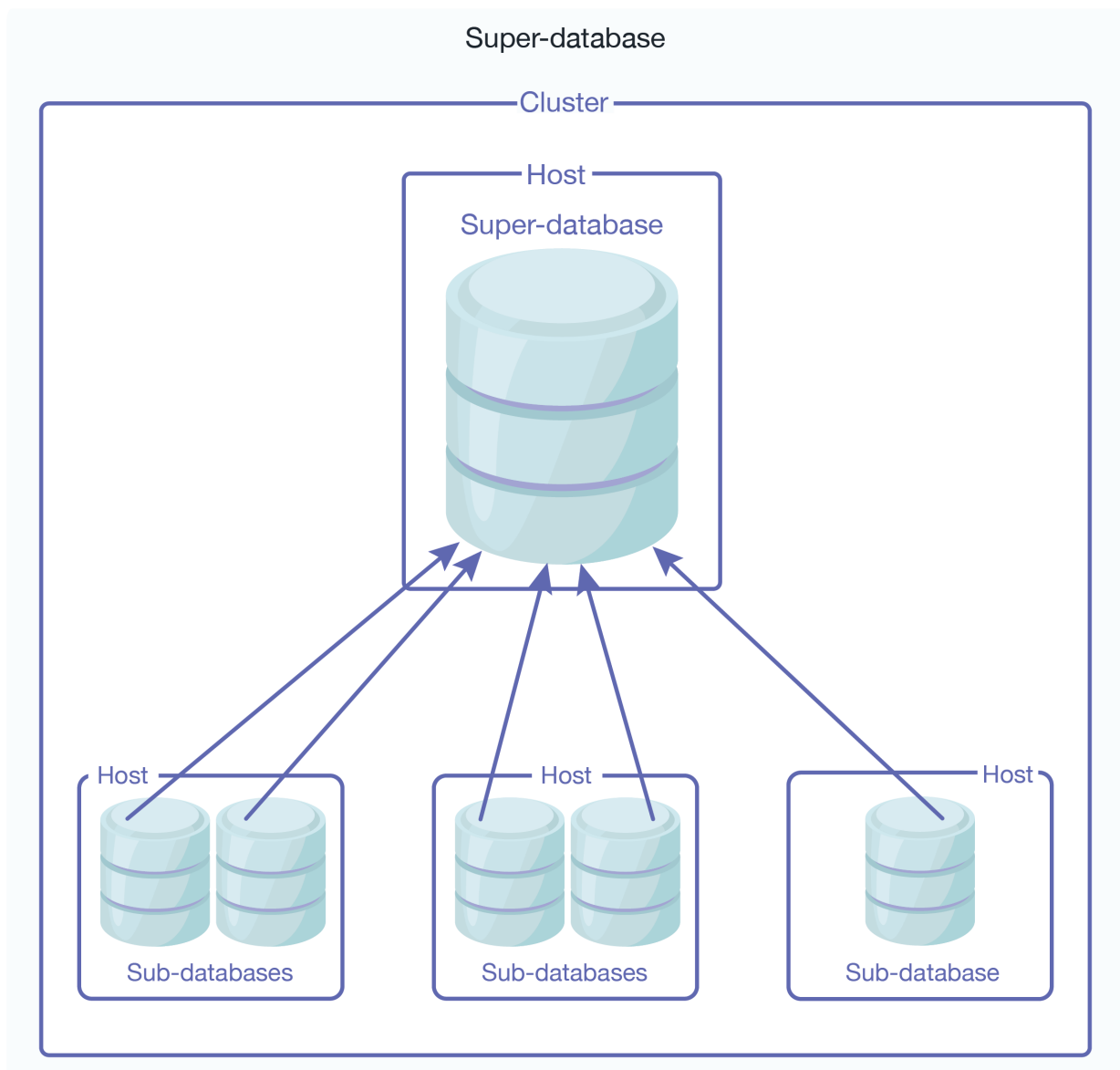
ensure that master and their replica forests are created on hosts in different zones. This ensures that the partition will remain accessible should a forest, host, or entire zone go down.

18. Super Databases and Clusters

MarkLogic Server allows you to group multiple databases into a *super-database* in order to allow a single query to be done across multiple databases. Databases contained in a super-database are called *sub-databases*. Sub-databases can be distributed on different storage tiers and on different clusters (collectively called *super-clusters*). A sub-database can be either *active* (online) or *archive* (offline), as specified by the `kind` element.

18.1. Overview

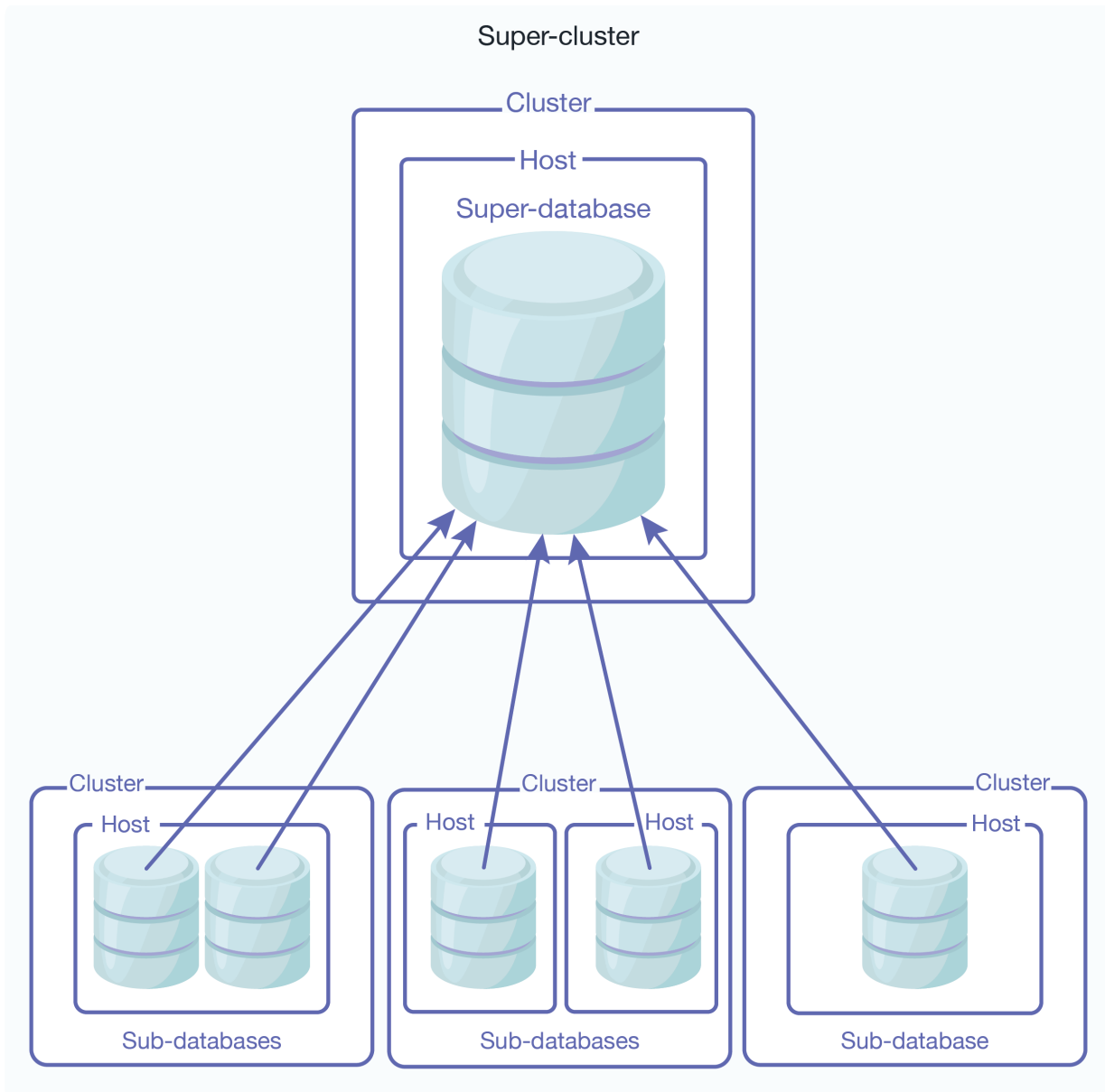
Updates are made on the sub-databases and then made visible for read in the super-database. Here is an illustration of a super-database and its sub-databases configured on a single cluster:



Here is an illustration of is a super-database configured with sub-databases on different clusters. The cluster hosting the super-database must be coupled with the foreign clusters hosting the sub-databases. For details on how to couple clusters, see [Section 4.3.3, "Coupling Clusters" \[27\]](#).

**NOTE**

Each foreign cluster should have multiple bootstrap hosts so that, if one bootstrap host goes down, the super-database can use the other bootstrap host to query the sub-databases on that cluster.



The following list describes the characteristics of super-databases and sub-databases:

- Only one level of sub-databases is supported for a super-database, which means that a sub-database cannot also be configured as a super-database with sub-databases of its own.
- Updates to the sub-databases are made visible on the super-database. You cannot write to a super-database and have the update propagated to its sub-databases. A super-database must have local forests for it to be updated. However, configuring a super-database with local forests is not recommended.
- Sub-databases and their super-databases must have the same index settings. Otherwise, queries will not work.

- Because super-databases and their sub-databases are effectively a single database, you cannot have documents with the same URI in super-databases and their sub-databases. It is a best practice to use directories to ensure that your document URIs are unique.
- You cannot run Flexible Replication on a super-database.
- When sub-databases are distributed across foreign clusters, the Security and Schemas databases must be the same for accessing the databases on each cluster. To ensure this, you should use Database Replication to replicate the Security and Schemas database on each cluster.
- When inserting data to a sub-database on a foreign cluster, you can read the inserted document on the super-database after the `request-timestamp` moves past the commit timestamp of the insert. Typically, this takes a few seconds.

18.2. Creating a Super-database

You can call the `POST: /manage/v2/databases` resource address to create a super-database. To create a super-database, simply specify which databases are to be its sub-databases.

For example, to define the `mySuperDatabase` database as a super-database containing the `subDB1`, `subDB2`, and `subDB3` sub-databases on the same cluster, enter this command:

```
$ curl --anyauth --user user:password -X POST \
-d '{"database-name": "mySuperDatabase",
"subdatabases": [
"subdatabase":{"cluster-name":"localhost", "database-name":"subDB1"},
"subdatabase":{"cluster-name":"localhost", "database-name":"subDB2"},
"subdatabase":{"cluster-name":"localhost", "database-name":"subDB3"}]
}'
-H 'Content-type: application/json' \
http://MyHost:8002/manage/v2/databases
```

18.3. Creating a Super-cluster

Before creating a super-cluster, you must couple the clusters as described in [Section 4.3.3, “Coupling Clusters” \[27\]](#).

For example, to define the `mySuperCluster` database as a super-cluster containing the `subDB1`, `subDB2`, and `subDB3` sub-databases on different clusters, enter this command:

```
$ curl --anyauth --user user:password -X POST \
-d '{"database-name": "mySuperCluster",
"subdatabases": [
"subdatabase":{"cluster-name":"cluster1", "database-name":"subDB1"},
"subdatabase":{"cluster-name":"cluster2", "database-name":"subDB2"},
"subdatabase":{"cluster-name":"cluster3", "database-name":"subDB3"}]
}'
-H 'Content-type: application/json' \
http://MyHost:8002/manage/v2/databases
```



NOTE

The maximum capacity for super-clusters is 32 clusters.

18.4. Viewing Super-databases and Sub-databases

You can call the `GET: /manage/v2/databases/{id|name}/super-databases` resource address to return a list of the super-databases associated with a sub-database. For example, to view the super-databases of the `subdb1` database, do the following:

```
$ curl --anyauth --user user:password -X GET \  
-H 'Content-type: application/xml' \  
http://MyHost:8002/manage/v2/databases/subdb1/super-databases
```

You can call the `GET: /manage/v2/databases/{id|name}/sub-databases` resource address to return a list of the sub-databases associated with a super-database. For example, to view the sub-databases of the `superdb1` database, do the following:

```
$ curl --anyauth --user user:password -X GET \  
-H 'Content-type: application/xml' \  
http://MyHost:8002/manage/v2/databases/superdb1/sub-databases
```



NOTE

Since updates can happen at both the super-database and the sub-database level, duplicate URIs are more likely in super-databases. Some automatically generated URIs may produce duplicates at the super-database level. This is true not only for automatically-generated URIs for graph documents, but also may be a problem for the bitemporal LSQT documents, and for directory properties fragments created with automatic-directory-creation. Duplicate URIs will generate a `DUPURI` exception.

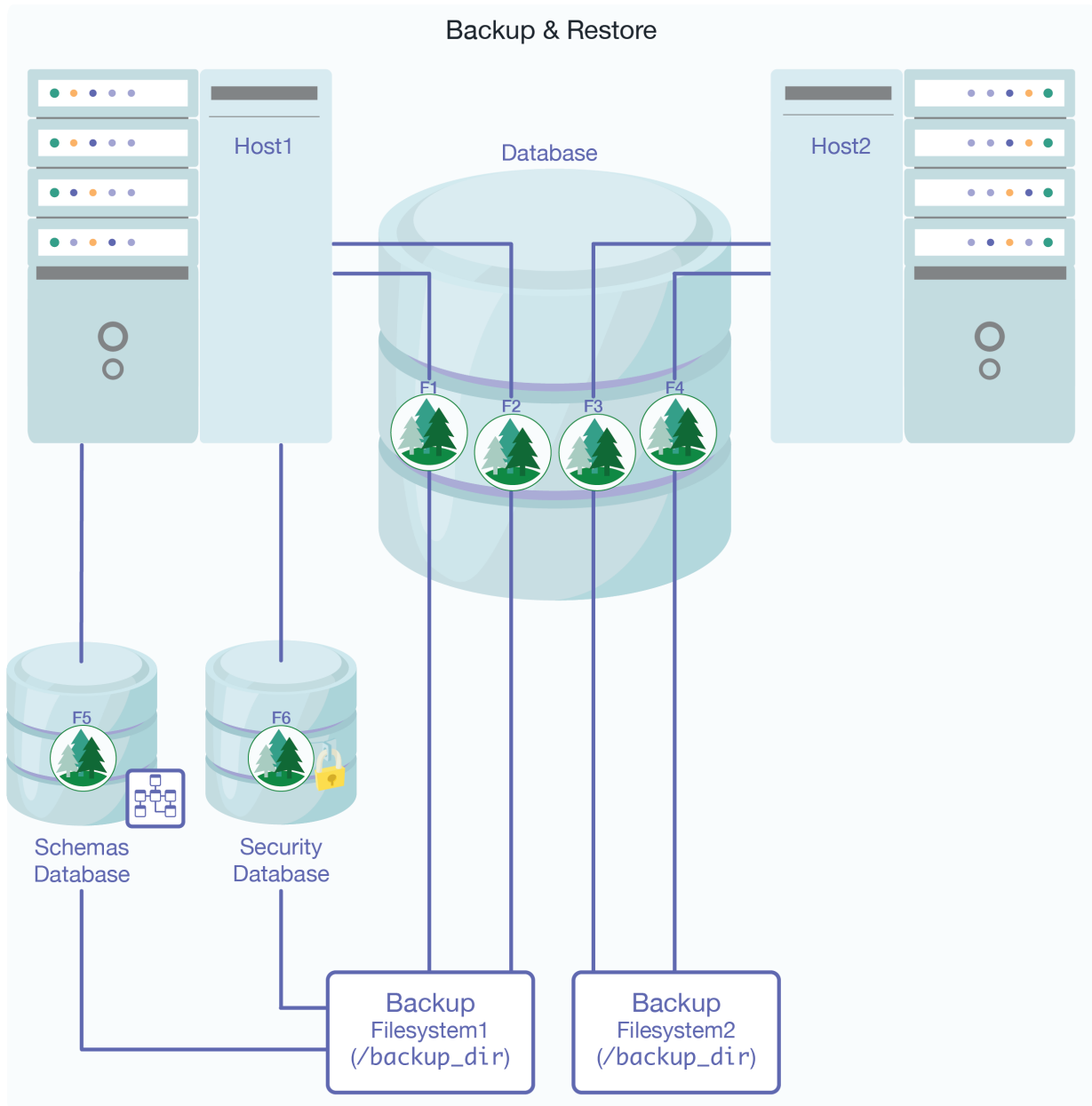
19. Backing Up and Restoring a Database

MarkLogic Server provides a facility to make a consistent backup of a database. This section describes the backup and restore architecture and provides procedures for backing up and restoring a database.

19.1. Backup and Restore Overview

Database backup and restore operations in MarkLogic Server are distributed over all of the data nodes in a cluster (that is, all of the nodes that contain forests), and provide consistent database-level backups and restores.

The directory you specify for a backup or restore operation must exist on each data node associated with the database (it can be either a shared or unshared directory). For example, if you have a data node on Host1 with forests F1 and F2, and another data node on Host2 with forests F3 and F4, then the backup directory you specify must exist on both Host1 and Host2. The following figure shows such a configuration, where the Schemas and Security databases have forests F5 and F6 respectively, and they are also attached to Host1.



19.1.1. Consistent, Database-Level Backup

By default, when you back up a database you back up everything associated with it, including these items:

- The configuration files.
- The Security database, including all of its forests.
- The Schemas database, including all of its forests.
- All of the forests of the database you are backing up.

If you choose to back up all forests, you will have a backup that you can restore to the exact same state as when the backup begins copying files.

You can also back up any individual forests that you choose, choosing only the ones you need to back up. These forest-level backups are consistent for the data in the forest and any other forests included in the backup, but might not be consistent with changes that occur in other forests not included in the backup.

You can also choose not to back up the Security and Schemas databases. While having backups of these databases that are synchronized with the database backups is important to get the exact same view of the system as when the backup began, you might have separate processes for backing up these databases that can ensure proper consistency. For example, if they do not change frequently, you may only need to back them up when they change.

The database-level backup and restore in MarkLogic Server provides the flexibility for you to decide how much or how little you want to back up or restore. The choices you make depend on the amount of change in your system and your unique backup and restore requirements.

19.1.2. Admin Interface

You use the Admin Interface to initiate backup and restore operations. Use the **Backup/Restore** tab for each database configured in your system to initiate backup and restore operations. For specific procedures for backup and restore operations, see [Section 19.4, “Backing Up a Database” \[169\]](#) and [Section 19.5.2, “Restoring a Database without Journal Archiving” \[173\]](#).

19.1.3. Backup and Restore Transactions

Backup and restore operations are transactional and therefore guarantee a consistent view of the data. They do not lock the database, however. Therefore, if the data in a database changes after a backup or restore operation begins but before it completes, those changes are not reflected in the backup or restore operation. Similarly, changes to the Security and Schemas databases during a backup or restore operation are allowed, but will not be reflected in the backup or restore.

Database and Forest administrative tasks such as drop, clear, and delete cannot take place during a backup; any such operation is queued up and will initiate after the backup transaction has completed.

19.1.4. Backup Directory Structure

When you back up a database, you specify a backup directory. That directory **must exist** on each host in your configuration, and it **must be readable and writable** by the user running MarkLogic Server (by default, `daemon` on UNIX and the local System user on Windows). When you back up multiple databases, a good practice is to create one backup directory for each and name them accordingly. Because of the importance of database backup integrity, MarkLogic recommends backing up to a reliable filesystem. The backup directory structure for each host is the same, except that the forests are only backed up on the host from which they are served.

Below the specified backup directory, a subdirectory is created with a name based on the date when the backup begins. Each of these subdirectories contains one backup. Here is the basic backup directory structure:

```

<specified_backup_dir>/
  <date_1>-1/
    *.xml
    BackupTag.txt
    Forests/
      <security_forest_1>/
        <forest_files_and_directories>
      <security_forest_n>/
        <forest_files_and_directories>
      <schemas_forest_1>/
        <forest_files_and_directories>
      <schemas_forest_n>/
        <forest_files_and_directories>
      <database_forest_1>/
        <forest_files_and_directories>
      <database_forest_n>/
        <forest_files_and_directories>
      <triggers_forest_1>/
        <forest_files_and_directories>
      <triggers_forest_n>/
        <forest_files_and_directories>
  <date_1>-n/
    <backup_directory_structure>
  <date_n>-1/
    <backup_directory_structure>
  <date_n>-n/
    <backup_directory_structure>

```

For example, if you back up a database to the `/space/backups/Documents` directory on September 1, 2004, a directory structure similar to this one is created:

```

/space/backups/Documents
  20040901-1/
    *.xml
    BackupTag.txt
    Forests/
      Documents/
        Label
        000001e1/
        Journals/
      Schemas/
        Label
        000001e1/
        Journals/
      Security/
        Label
        000001e1/
        Journals/
      Triggers/
        Label
        000001e1/
        Journals/
/space/backups/Modules
  ...

```

Incremental backups are stored in the directory under the full backup. In this example, the backup directory (`backup-dir`) is `/space/backups/Documents`, and the incremental backup directory (`incremental-dir`) is not used:

```

/space/backups/Documents
  20140801-1223942093224 (full backup on 8/1)
  20140802
    331006226070 (incremental backup on 8/2)
  20130803
    1341007528950 (incremental backup on 8/3)

```

The first part, 20140801, is the year, month, and day of the backup. The second part, 1223942093224, is the hour, minute, second, and nanosecond of the backup.

In this example, the backup directory (backup-dir) is `/space/backups/Documents`, and the incremental backup directory (incremental-dir) is `/space/incremental`:

```

/space/backups/Documents
  20140801-1223942093224 (full backup on 8/1)

/space/incremental
  20140801-1223942093224
    20140802
      331006226070 (incremental backup on 8/2)
    20140803
      341007528950 (incremental backup on 8/3)
    
```

The directory 20130801-1223942093224 is created on `/space/incremental` so that when the backup 20130801-1223942093224 is purged, its incremental backups can be purged easily.

If an incremental backup directory is specified, after the first incremental backup is done, the full backup can be archived to another location. The subsequent incremental backups do not need to examine the full backup.



NOTE

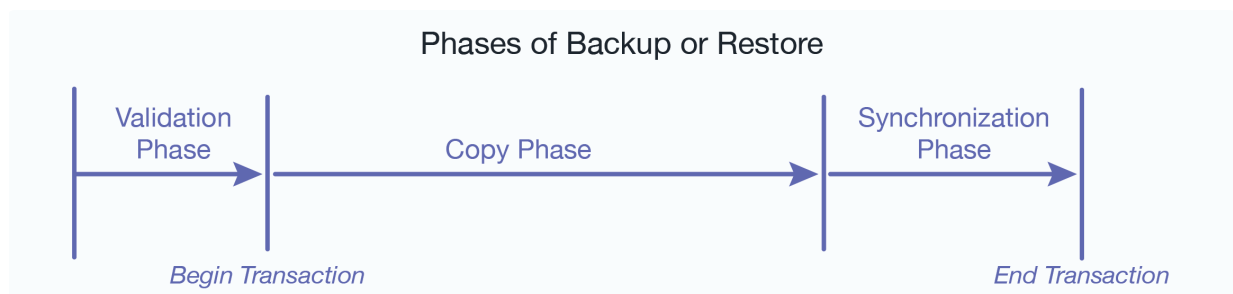
After you restore an incremental backup, you can no longer use the previous full backup location for ongoing incremental backups. After the restore, you need to make a fresh full backup and use the full backup location for ongoing incremental backups. This means that after the restore of an incremental backup, scheduled backups need to be updated to use the fresh full backup location.

19.1.5. Phases of Backup or Restore Operation

Backup and restore operations are divided into the following phases:

- Validation
- Copy
- Synchronization

The following figure shows the phases of a backup or restore operation:



Validation Phase

The validation phase is where the backup directories are checked to make sure that all of the needed files exist and that all of the needed backup directories exist and are writable. For backup operations, they are checked for sufficient disk space. For restore operations, the configuration files are read and

the other backup files are checked to make sure they appear to be valid. The validation phase does not actually write any data and is completely asynchronous.

Copy Phase

The copy phase is where the files are actually copied to or from the backup directory. The configuration files are copied at the beginning of the backup operation, and at this point a timestamp is written to the `BackupTag.txt` file. The copy phase might take a significant amount of time, depending on the size of the database. The start of the copy phase starts a transaction; if the transaction fails on a restore operation, the database remains unchanged from its original state.

Synchronization Phase

During a backup or restore operation, the synchronization phase is where cleanup tasks such as deleting temporary files takes place, leaving the database in a consistent state. During a restore operation, the synchronization phase also takes the old version of the database offline and replaces it with the newly restored version.



NOTE

Any “cold” administrative tasks (tasks that require a server restart) will cause any backup or restore operations to fail. Do not perform any “cold” administrative tasks during a backup or restore operation. For a list of “hot” and “cold” operations, see [Section 31, “Appendix A: ‘Hot’ versus ‘Cold’ Admin Tasks” \[268\]](#).

19.1.6. Notes about Backup and Restore Operations

This section provides notes and restrictions about backing up and restoring MarkLogic Server databases.

- For backing up and restoring a database with encryption, see [Backup and Restore](#) in *Securing MarkLogic Server*.
- The backup files are platform specific—backups on a given platform should only be restored onto the same platform. This is true for both database and forest backups.
- You can restore an individual forest using a database backup by unchecking all forests except the one you want to restore on the Confirm Restore screen (see [Step 11](#) in [Section 19.5.2, “Restoring a Database without Journal Archiving” \[173\]](#)).
- We recommend using the database-level backup/restore, not the forest-level backup/restore. If you do use the forest-level backup/restore, note that you cannot restore a backup created with the forest-level backup as a database-level restore operation; forest-level backups created with the forest backup/restore utility must be restored from the forest restore utility. For details, see [Section 22.7, “Restoring a Forest” \[208\]](#).
- The restore operation is designed to restore into a database that has the same configuration settings as the one that was backed up, but it neither requires nor checks that the configurations are the same. The restore operation must occur on a database that has its configuration defined. Also, the restore operation does not change the database configuration files. Because the configuration files hold all of the database configuration information such as index options, fragmentation, range indexes, and so on, the restored database will take on the configuration information of the database to which it is restored. If this configuration information is different from the database that was backed up, and if reindexing is enabled, the database will reindex to the new configuration after the restore completes.
- If a database’s backup is canceled, the in-flight backup is deleted. A database backup can be canceled by clicking the cancel button for the backup in the host status page in the Admin Interface, by the host or cluster being restarted (either from the Admin Interface or from the `xmmp:restart`

command), or by errors in the backup (such as out-of-disk space errors). The process of deleting the in-flight backup during a clean restart might take some time, which can increase the time it takes to restart MarkLogic Server. If you are restarting using the startup scripts (`/sbin/service MarkLogic <command>`) on UNIX systems and the control panel on Windows systems), then the script will delete as much of the backup as it can in 20 seconds; if any backup is in-flight during these types of system shutdown or restart operations, then you should manually remove them after the operation.

- After you restore from an incremental backup, you can't use the previous full backup location for ongoing incremental backups. You will need to make a fresh full backup after the restore and use that full backup location for the ongoing incremental backups. This means that after the restore of an incremental backup, any scheduled backups will need to be updated to use the new full backup location.

19.2. Backing Up Databases with Journal Archiving

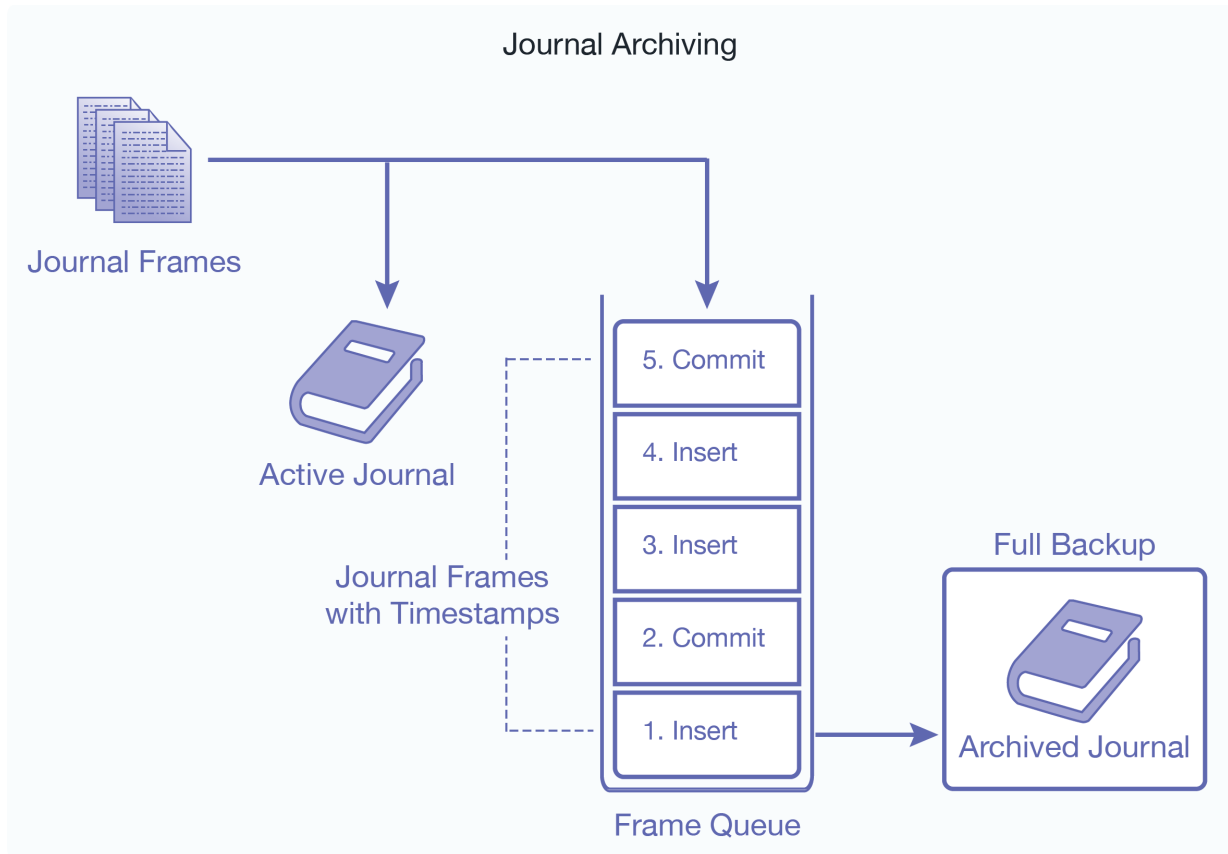
The backup/restore operations with journal archiving enabled provide a point-in-time recovery option that enables you to restore database changes to a specific point in time between full backups with the input of a wall clock time. When journal archiving is enabled, journal frames are written to backup directories by near synchronously streaming frames from the current active journal of each forest.




NOTE

When you create scheduled backups with journal archiving enabled, and then later delete the backup, it does not stop journal archiving from occurring even though the backups stop happening. The `xdmp:stop-journal-archiving` function must be explicitly called to stop journal archiving.

When journal archiving is enabled, you will experience longer restore times and slightly increased system load as a result of the streaming of journal frames.



 **NOTE** Journal archiving can only be enabled at the time of a full backup. If you restore a backup and want to reenale journal archiving, you must perform a full backup at that time.

When journal archiving is enabled, you can set a lag limit value that specifies the amount of time (in seconds) in which frames being written to the forest's journal can differ from the frames being streamed to the backup journal. For example, if the lag limit is set to 30 seconds, the archived journal can lag behind a maximum of 30 seconds worth of transactions compared to the active journal. If the lag limit is exceeded, transactions are halted until the backup journal has caught up.

The active and backup journal are synchronized at least every 30 seconds. If the lag limit is less than 30 seconds, synchronization will be performed at least once in that period. If the lag limit is greater than 30 seconds, synchronization will be performed at least once every 30 seconds. The default lag limit is 15 seconds.

The decision on setting a lag limit time is determined by your Recovery Point Objective (RPO), which is the amount of data you can afford to lose in the event of a disaster. A low RPO means that you will restore the most data at the cost of performance, whereas a higher RPO means that you will potentially restore less data with the benefit of less impact to performance. In general, the lag limit you chose depends on the following factors:

A lower lag limit implies:

- Accurate synchronization between active and backup journals at the potential cost of system performance.
- Use when you have an archive location with high I/O bandwidth and your RPO objective is low.

A higher lag limit implies:

- Delayed synchronization between active and backup journals, but lesser impact on system performance.
- Higher server memory utilization due to pending frames being held in memory.
- Use when you have an archive location with low I/O bandwidth and your RPO objective is high.

19.3. Incremental Backup

An incremental backup stores only the data that has changed since the previous full or incremental backup. Typically a series of incremental backups are done between full backups. Incremental backups are more compact than archived journals and are faster to restore. It is possible to schedule frequent incremental backups (for example, by ranges of hours) because an incremental backup generally takes less time to complete than a full backup. In normal conditions, it is recommended an incremental backup not be configured for a frequency less than every four hours.

To enable an incremental backup, set Incremental backup to `true` while initiating or scheduling a backup. See [Section 19.4, “Backing Up a Database” \[169\]](#) for details. Full and incremental backups need to be scheduled separately. An example configuration might be:

- Full backups scheduled monthly
- Incremental backups scheduled daily

A full backup and a series of incremental backups can allow you to recover from a situation where a database has been lost. Incremental backup can be used with or without journal archiving. If you enable both incremental backup and journal archiving, you can replay the journal starting from the last incremental backup timestamp. See [Section 19.2, “Backing Up Databases with Journal Archiving” \[166\]](#) for more about journal archiving.



NOTE

When you restore from an incremental backup, you need to do a full backup before you can continue with incremental backups.

Incremental backup and journal archiving both provide disaster recovery. Incremental backup uses less disk space than journal archiving, and incremental backup is faster than using journal archiving.



NOTE

If MarkLogic Server cannot memory-map files from the backup in the underlying file system, it cannot create an incremental backup. So MarkLogic incremental backups require that the backup file system support memory-mapping operations (mmap).

For recovery you only need to specify the timestamp for the recovery to start and the server will figure out which full backup and which incremental backup(s) to use. You only need to schedule the incremental backup; the server will link together (or chain) the sequence the incremental backups

automatically. See [Section 19.5.4, “Restoring from an Incremental Backup with Journal Archiving” \[176\]](#) for details.

19.3.1. Including New Forests in Incremental Backups

Incremental backup supports backup of a forest added since last full backup. If you add a new forest after a full backup of your database, you can include the new forest as part of your next incremental backup.

1. Attach the new forest to your database. It automatically appears in the list of forests to be backed up in the Confirm backup step ([Step 12 in Section 19.4.1, “Backing Up a Database Immediately” \[169\]](#)).
2. Select the forest to include it in the backup and click **ok**. See [Section 19.4, “Backing Up a Database” \[169\]](#) for more information.

19.3.2. Using Journal Archiving with Incremental Backups

Incremental backup improves both time and space restore requirements over journal archiving, but it's not an either/or decision. You can, and should, use both where appropriate. If your goal is to be able to restore to any arbitrary point in time while minimizing potential data loss, follow these steps:

1. Configure a scheduled full backup at some coarse granularity (for example, weekly) and enable journal archiving.
2. Configure a scheduled incremental backup at some finer granularity (for example, every four hours), and specify `purge-journal-archiving=true`.
3. Set `retain until backup` on the database Merge Policy so that deleted fragments are retained until they have been included in an incremental backup. See [Section 15.2, “Setting Merge Policy” \[109\]](#) or `admin:database-set-retain-until-backup` for details.

This configuration means that journal archives are only needed since the most recent incremental backup, and the older ones can be purged once there is another incremental backup. Enabling `retain until backup` ensures that the incremental backups have sufficient state to restore the database to any point since the previous incremental backup.

When you restore, the full and incremental backups can be used to return to any point in time prior to the most recent backup, and the journal archive will only be used if your restore point is more recent than the last incremental backup.

19.4. Backing Up a Database

You can either initiate a database backup immediately or you can schedule a backup to occur in the future.

The backup procedures include options to specify journal archiving and/or incremental backup. You can choose to do a full backup or incremental backup, with or without journal archiving enabled.

19.4.1. Backing Up a Database Immediately

Follow these steps to initiate a database backup:

1. Log into the Admin Interface as a user with the `admin` role.
2. Click **Databases** in the left tree menu. A list of databases appears.
3. Click your target database.
4. Click the **Backup/Restore** tab.
5. Enter the directory to which you want the database backed up in the **Backup to directory** field.

**NOTE**

The backup directory path must exist on all hosts that serve any forests in the database. The directory you specified can be an operating system mounted directory path, it can be an HDFS path, or it can be an S3 path. For details on using HDFS and S3 storage in MarkLogic, see [Disk Storage Considerations](#) in the *Query Performance and Tuning Guide*. Additionally, if you are using Windows and are backing up to a remote Windows path, you must set the registry settings and permissions as described in [Windows Shared Disk Registry Settings and Permissions](#).

6. If you want to encrypt your backup, enter an encryption password.
7. If you have configured forests for local-disk failover, you can optionally set **Include replica forests** to **true** if you want to include the replica forests in the backup. For details on configuring forests for local-disk failover, see [Configuring Local-Disk Failover for a Forest](#) in the *Scalability, Availability, and Failover Guide*.
8. Set **Incremental backup** to **true** to create an incremental backup. The default is a full backup (false).
9. Set **Archive journals** to **true** and set the Journal archiving lag limit if you want to enable point-in-time recovery. The journal archiving lag limit is described in [Section 19.2, “Backing Up Databases with Journal Archiving” \[166\]](#).

**NOTE**

If journal archiving is enabled, you cannot include auxiliary forests, as they should have their own separate backups.

Summary	Configure	Status	Forest Status	Backup/Restore	Load	Create	Help
---------	-----------	--------	---------------	----------------	------	--------	------

Backup The *App-Services* Database.

Backup to directory
The backup directory pathname for this database. Each database should use a different backup directory. Required.

Encryption password
Optional password to use for encrypting or decrypting backup files. Password must be between 16 and 1000 characters.

Confirm encryption password
Optional password to use for encrypting or decrypting backup files. Password must be between 16 and 1000 characters.

Include replica forests true false

Incremental backup true false

Archive journals true false

10. Click **OK**.
11. If a directory creation error appears, then the directory is not writable. Either change the permissions on an existing directory or create a new directory with the proper permissions (readable and writable by the user running MarkLogic Server, by default `daemon` on UNIX and the local System user on Windows) and click **OK** again.
12. The **Confirm backup** screen appears and lists all the forest selected for back up.
13. Click **OK** to begin the backup immediately, or deselect forests that you do not want to back up.

**NOTE**

If you deselect any of the forests to backup, you might not have a completely consistent view of the database to restore. Only deselect any forests if you are sure you understand the implications of what you are backing up. To guarantee the exact same view of the database, backup all of the forests associated with the database, including the Schemas and Security database forests.

14. After the backup is underway, the Admin Interface redirects you to the **Database Status** page.
15. You can refresh the **Database Status** screen to view the progress of the backup. The **Backups** table lists when the backup was started, provides an estimate of the amount of time left, and lists other status information about the backup operation.
16. When the backup is complete, the entry in the backup table disappears.

If the status for any of the forests was something besides “completed,” then an error occurred during the backup operation. Check the `Mark_Logic_Data/Logs/ErrorLog.txt` file for any errors, correct them, and try the backup operation again.

19.4.2. Scheduling a Database Backup

You can schedule database backups to periodically back up a database. You can schedule backups to occur daily, weekly, monthly, or you can schedule a one-time backup. You can create as many scheduled backups as you want.

To create a scheduled backup, follow these steps in the Admin Interface:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click **Scheduled Backups** in the left tree menu for the database.
4. On the **Scheduled Backup** page, you can delete any existing scheduled backups if you no longer need them.
5. Click the **Create** tab.
6. In the **Backup Directory** field, enter the absolute path to the backup directory. The backup directory must have permissions such that the MarkLogic Server process can read and write to it.

**NOTE**

The backup directory path must exist on all hosts that serve any forests in the database. The directory you specified can be an operating system mounted directory path, it can be an HDFS path, or it can be an S3 path. For details on using HDFS and S3 storage in MarkLogic, see [Disk Storage Considerations](#) in the *Query Performance and Tuning Guide*.

7. In the **Backup Type** field, select the appropriate value.
 - If you select **minutely**, in the **Backup Period** field, enter how many minutes between each backup.
 - If you select **hourly**, enter how many hours between each backup. The `Backup Minute` setting specifies how many minutes after the hour the backup is to start. Note that the `Backup Minute` setting does not add to the interval.
 - For **daily**, enter how many days between each backup and the time of day.
 - For **weekly**, enter how many weeks between each backup, check one or more days of the week, and the time of day for the backup to start.
 - For **monthly**, enter how many months between each backup, select one day of the month (1-31), and the time of day for the backup to start.

- For one-time, enter the backup start date in MM/DD/YYYY notation (for example, 07/29/2009 for July 29, 2009) and time in 24:00 notation.
8. Enter the time of day to start the backup.
 9. Enter the maximum number of backups to keep. When you reach the specified maximum number of backups, the next backup will delete the oldest backup. Specify 0 to keep an unlimited number of backups.
 10. Choose whether you want the backups to include the security database, the schemas database, and/or the triggers database for this scheduled backup.
 11. Choose whether you want the backups to include the replica forests, as well as the master forests.
 12. Choose whether you want to schedule an incremental backup or a full backup.
 13. Choose whether you want the backups to enable Journal Archiving for point-in-time recovery. For details on Journal Archiving, see [Section 19.2, “Backing Up Databases with Journal Archiving” \[166\]](#).

**NOTE**

If Journal Archiving is enabled, you cannot include auxiliary forests, as they should have their own separate backups.

14. If you have enabled Journal Archiving, you can change the lag limit to control the amount of time in seconds in which a journal being backed up can differ from the current active journal.
15. Click **OK** to create the scheduled backup.

The backups will automatically start according to the specified schedule.

19.5. Restoring a Database from a Backup

This section describes a number of ways to restore a database from a backup.

**NOTE**

Depending on how the backup was made and what has changed since then, some restore operations may require a combination of these procedures.

**WARNING**

Do not restore the App-Services database from another cluster because this type of backup causes the following error: `MANAGE-TIMESTAMPOLD: Config files out of date on host.`

19.5.1. Admin Interface for Database Restore

This section describes the Admin Interface used to restore a database.

To access the database restore page, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click the **Backup/Restore** tab.
4. Scroll down to the Restore section.

5. Complete the Restore form as needed. The database restore settings are in the [table](#).
6. Click **OK**.

Database Restore Setting	Description
Restore from directory	Specifies the fully-qualified pathname for the directory from which to restore a backup. If the top-level backup directory is specified, then the restore operation restores the most recent backup. If a specific backup is specified, then that backup is restored.
Encryption password	An optional password to use for encrypting or decrypting backup files. Password must be between 16 and 1000 characters.
Include Replica Forests	Specifies whether to include the replica forests used for local-disk failover in the backup.
Use incremental backup	Specifies whether to use incremental backup.
Use journal archive	Specifies whether to enable the point-in-time recovery feature.
Forest topology changed	Specifies whether the forest topology has changed the last backup.
Include auxiliary databases	Specifies whether to include the auxiliary databases.
Restore to time	Specifies the time to which the database is to be restored. Leave blank for latest restore time.

19.5.2. Restoring a Database without Journal Archiving

This section describes how to restore a database if no journal archiving was enabled for the last backup.



NOTE

If your last backup enabled Journal Archiving, stop here and follow the procedure described in [Section 19.5.3, “Restoring Databases with Journal Archiving” \[175\]](#).

To restore an entire database from a backup, follow these steps:



NOTE

You must have the Admin role to complete this procedure.


1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click the **Backup/Restore** tab.
4. Scroll down to the Restore section.
5. Enter the directory in which the back up exists in the **Restore From directory** field.
6. If the backup was encrypted, enter the encryption password.



NOTE

If you enter a directory that contains multiple backups of the same database, the latest one is used. If you want to choose a particular backup to restore, enter the *date_stamp* subdirectory corresponding to the backup you want to restore. For details of the directory structure, see [Section 19.1.4, “Backup Directory Structure” \[162\]](#).

7. If you have configured forests for local-disk failover, you can optionally set **Include replica forests** to true if you want to restore the replica forests from the backup. In order to use this option, you must have enabled the option to include the replica forests in the backup. For details on configuring forests for local-disk failover, see [Configuring Local-Disk Failover for a Forest](#) in the *Scalability, Availability, and Failover Guide*.
8. If you want to restore an incremental back up, set **Use incremental backup** to true.



NOTE

If you restore from an incremental backup, you can't use the previous full backup location for ongoing incremental backups. You need to make a fresh full backup after the restore and use the full backup location for the ongoing incremental backups. After doing a restore from an incremental backup, any scheduled backups will need to be updated to use the new full backup location.

9. Leave **Use journal archive**. set to **false**.

May 15, 2023 10:08 AM

- Server
- Groups
- Databases**
- App-Services
- Documents**
- Forests
- Sub-Databases
- Flexible Replication
- Database Replication
- Fragment Roots
- Fragment Parents
- Triggers
- Merge Policy
- Scheduled Backups
- Content Processing
- Element Range Indexes
- Attribute Range Indexes
- Field Range Indexes
- Path Namespaces
- Path Range Indexes
- Element Word Lexicons
- Attribute Word Lexicons
- Word Query

Restore The Documents Database.

Restore from directory

The backup directory pathname for this database. Each database should use a different backup directory.

Encryption password

Optional password to use for encrypting or decrypting backup files. Password must be between 16 and 1000 characters.

Include replica forests true false

Use incremental backup true false

Use journal archive true false


Forest topology changed true false

Include auxiliary databases true false

Restore to time:

Leave blank for latest restore time or use xs:DateTime-Format like 2023-05-15T10:08:06.4691775-04:00

10. Click **OK**.
11. The **Confirm Restore** screen appears and lists all the forest selected for restoring. The **Confirm Restore** screen also lists the date the backup was performed and the server version used for the backup you selected.
12. By default, all of the forests associated with a database are checked to restore. If you do not want to restore all of the forests, deselect any forests you do not want to restore.



NOTE

If you deselect any of the forests to restore, you might not be restoring a completely consistent view of the database. Only deselect any forests if you are sure you understand the implications of what you are restoring. To guarantee the exact same view of the database, restore all of the forests associated with the database, including the Schemas and Security database forests.

13. Click **OK** to begin the restore operation. The `Restores` table lists when the restore was started, provides an estimate of the amount of time left, and lists other status information about the restore operation.

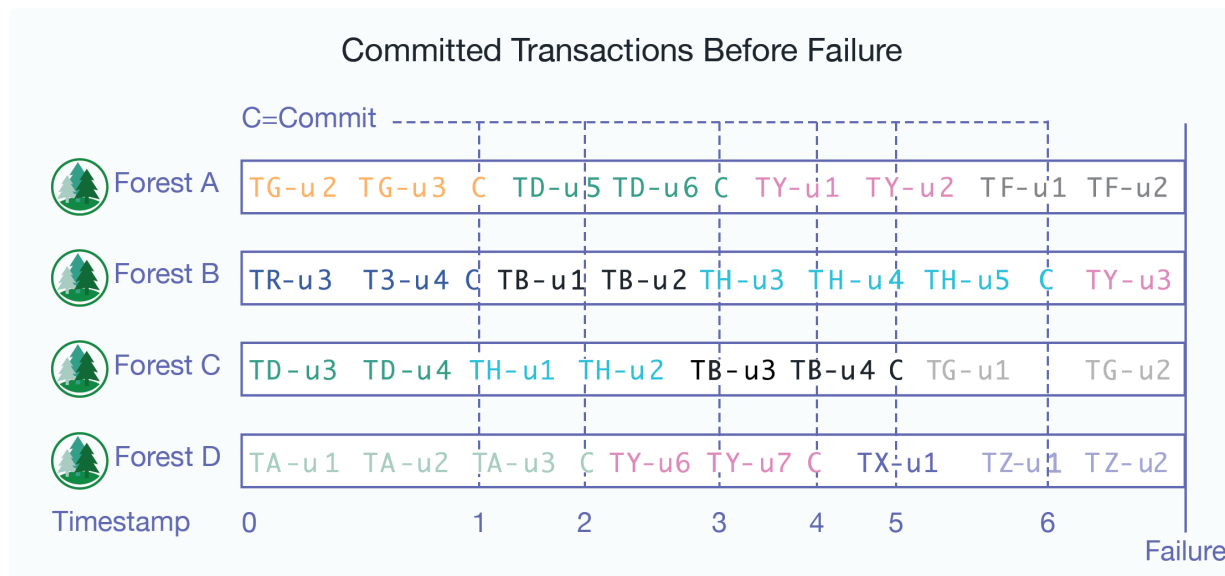
When the restore is complete, the entry in the backup table disappears. If the status for any of the forests was something besides "completed," then an error occurred during the restore operation. Check

the `Mark_Logic_Data/Logs/ErrorLog.txt` file for any errors, correct them, and try the restore operation again.

19.5.3. Restoring Databases with Journal Archiving

After you restore a database with Journal Archiving enabled, each forest will likely have committed its last transaction at different timestamps.

For example, the illustration below shows four forests and their committed transactions. Updates for each transaction are identified by the convention 'T#-u#' and commits are identified by a 'C'. Each forest completed its last commit at a different point in time when the restore is finished. In this example, we are restoring from timestamp 0 to 6, Forest A has only committed transactions up to timestamp 3 while Forest B has committed transactions up to timestamp 6. This means that, in order to return the database to a transactionally consistent state, all forests must be rolled back to timestamp 3 or earlier.



Your options for recovering your data and returning the database to a transactionally consistent state are as follows:

- Restore as much data as possible. Follow the procedure described in [Section 19.5.5, “Restoring to the Safe Timestamp” \[177\]](#).
- Restore data at a specific timestamp. Follow the procedure described in [Section 19.5.6, “Restoring to a Specific Timestamp” \[178\]](#).
- Restore data at a specific timestamp based on the state of some sample documents. Follow the procedure described in [Section 19.5.7, “Restoring Based on Sample Documents” \[179\]](#).

The following sections describe how to use the XQuery API to restore the database. You can also use the Admin Interface to accomplish some of the tasks.



NOTE

If you are using XA distributed transaction processing, a restore to a point in time may revive some XA transactions that were prepared before the target restore time, and committed/aborted after that time. For details on how to identify XA transactions, see [Heuristically Completing a MarkLogic Server Transaction in *Developing with XCC*](#).

You cannot roll back through a database clear operation, so you should check the server logs for points in time that any clear operations occurred.

19.5.4. Restoring from an Incremental Backup with Journal Archiving

To restore from an incremental backup, the server uses the base backup in the backup tag to get a series of incremental backups that lead to the full backup. The restore then starts with a full backup and restores using the incremental backups in reverse order. You need to specify the full backup directory and optionally the incremental backup directory. If no restore timestamp is specified, the server finds the latest backup from which to restore. Once you have completed this process, you can use journal archiving to restore the database to the current time.

If a restore timestamp is specified, the server finds a backup where the restore timestamp is between the minimum query timestamp and the backup timestamp. If no backup meets the requirement and there is a journal archive, the server finds the latest backup with backup timestamp smaller than the restored timestamp. It restores to that backup and then replays the journal to the restored timestamp.

If the journal archive exists, the server will find the backup timestamp of the last incremental backup and replay the journal starting from that timestamp.



NOTE

Once you restore from an incremental backup, you can no longer use the previous full backup location for ongoing incremental backups. After the restore, you need to make a fresh full backup and use that full backup location for the ongoing incremental backups. This means after the restore from an incremental backup, any scheduled backups will need to be updated to use the new full backup location. Using the old full backup location for incremental backup after a restore will cause an error.

This procedure describes how to restore a database to the current point in time using a full backup, one or more incremental backup, and journal archiving. You need to have a full backup using journal archiving and one or more incremental backups using journal archiving.

To restore a database to the current point in time, follow these steps:



NOTE

You must have the Admin role to complete this procedure.


1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click the **Backup/Restore** tab.
4. Scroll down to the Restore section.
5. In the **Restore from directory** field, enter the directory where the backup exists.



NOTE

If you enter a directory that contains multiple backups of the same database, the latest one is used. If you want to choose a particular backup to restore, enter the *date_stamp* subdirectory corresponding to the backup you want to restore. For details of the directory structure, see [Section 19.1.4, "Backup Directory Structure" \[162\]](#).

6. If you have configured forests for local-disk failover, you can optionally set **Include replica forests** to **true** if you want to restore the replica forests from the backup. In order to use this option, you must have enabled the option to include the replica forests in the backup. For details on configuring forests for local-disk failover, see [Configuring Local-Disk Failover for a Forest](#) in the *Scalability, Availability, and Failover Guide*.
7. Set **Use incremental backup** to **true**. Set **Use journal archive** to **true**. Leave **Restore to time** blank or enter a time in xs:DateTime-Format.



NOTE
For Journal archiving to work, you need a Restore to time, otherwise the restore will proceed with last Incremental backup it finds at the location. Also, the Merge Timestamp should be older than the Restore Time.

When restoring a backup with journal archiving enabled, be sure to change the merge timestamp from 0 to a non-zero value. Using zero for the merge timestamp will result in an error when restoring with journal archiving and restore-to-time set to zero. The merge timestamp must be set to a non-zero value.

8. Click **OK** to begin the restore process.
9. The Confirm restore screen lists the options you selected for restoring. Click **OK**.

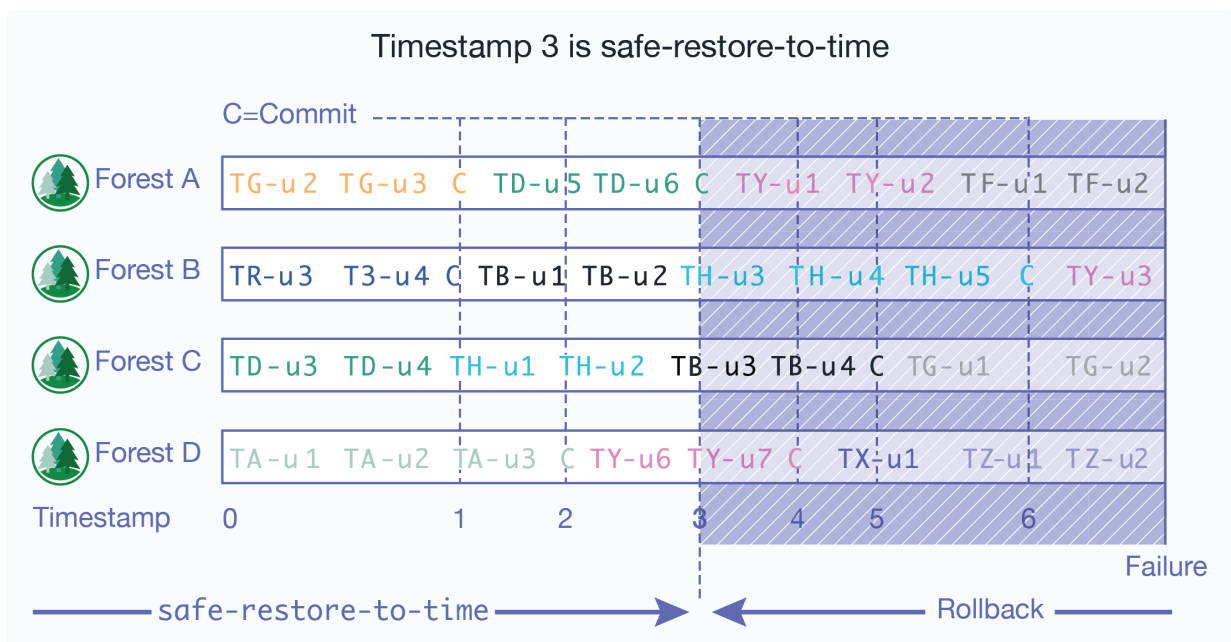
The Restores table lists when the restore was started, provides an estimate of the amount of time left, and lists other status information about the restore operation.

When the process is complete, the Restores table entry will disappear.

19.5.5. Restoring to the Safe Timestamp

If you want to restore as much data as possible, you can restore your data to the minimum safe timestamp.

For example, the database you want to restore has four forests, as shown below. You use the `xdrm:host-status` function to locate the `safe-restore-to-time` value, which is earliest of the four last-commit timestamps. In this example, the `safe-restore-to-time` is the timestamp of the last committed transaction in Forest A.



The following procedure describes how to restore to the minimum timestamp using the XQuery API.

**NOTE**

This same procedure can be done using the Admin Interfaces described in [Section 15.2, “Setting Merge Policy” \[109\]](#), [Section 19.5.1, “Admin Interface for Database Restore” \[172\]](#), and [Section 21.9, “Rolling Back a Transaction” \[200\]](#).

1. Use the `admin:database-get-merge-timestamp` function to get the current merge timestamp. Save this value so it can be reset after you have completed the rollback operation.
2. Use the `admin:database-set-merge-timestamp` function to set the merge timestamp to any time before your minimum safe timestamp. This will preserve fragments in merge after this timestamp until you have rolled back your forest data.
3. Use the `xdmp:database-restore` function with `$journal-archiving` set to `fn:true()` and `$restoreToTime` set to `null()` to restore the database to the latest timestamp.
4. After the restore operation has completed, use the `xdmp:forest-rollback` function to roll back the forests to the `safe-restore-to-time` timestamp returned by the `xdmp:host-status` function.

For example, if you are restoring the Documents database, you can use the following query to rollback your forest data:

```
xquery version "1.0-ml";
declare namespace host = "http://marklogic.com/xdmp/status/host";
let $timestamp :=
  xdmp:wallclock-to-timestamp(
    xs:dateTime(xdmp:host-status(xdmp:host("your-host.com"))
      /host:restore-jobs/host:restore-job/host:safe-restore-to-time
      /fn:data())
  )
return
  xdmp:forest-rollback(
    xdmp:database-forests(xdmp:database("Documents")),
    $timestamp)
```

5. Use `admin:database-set-merge-timestamp` function to set the merge timestamp back to the value you saved in [Step 1](#).

19.5.6. Restoring to a Specific Timestamp

This procedure describes how to restore a database to a specific timestamp using the XQuery API.

**NOTE**

This same procedure can be done using the Admin Interfaces described in [Section 15.2, “Setting Merge Policy” \[109\]](#), [Section 19.5.1, “Admin Interface for Database Restore” \[172\]](#), and [Section 21.9, “Rolling Back a Transaction” \[200\]](#).

1. Use the `admin:database-get-merge-timestamp` function to get the current merge timestamp. Save this value so it can be reset after you have completed the rollback operation.
2. Use the `admin:database-set-merge-timestamp` function to set the merge timestamp to any time before the restore timestamp. This will preserve fragments in merge after this timestamp until you have rolled back your forest data.
3. Use the `xdmp:database-restore` function with `$journal-archiving` set to `fn:true()` and `$restoreToTime` set to the restore timestamp to restore the database.
4. After the restore operation has completed, use the `xdmp:forest-rollback` function to roll back the forests to the restore timestamp. For example, if you are restoring the Documents database

and the restore timestamp is `2011-09-13T10:50:21.201832-07:00`, your `xdmp:forest-rollback` function call would be:

```
xdmp:forest-rollback(
  xdmp:database-forests(xdmp:database("Documents")),
  xdmp:wallclock-to-timestamp(
    xs:dateTime("2011-09-13T10:50:21.201832-07:00")))
```

5. Use `admin:database-set-merge-timestamp` function to set the merge timestamp back to the value you saved in [Step 1](#).

19.5.7. Restoring Based on Sample Documents

You may want to use the state of some sample documents to determine the time at which to restore the database.



NOTE

This same procedure can be done using the Admin Interfaces described in [Section 15.2, “Setting Merge Policy” \[109\]](#), [Section 19.5.1, “Admin Interface for Database Restore” \[172\]](#), and [Section 21.9, “Rolling Back a Transaction” \[200\]](#).

To restore to the state of sample documents using the XQuery API, follow these steps:

1. Use the `admin:database-get-merge-timestamp` function to get the current merge timestamp. Save this value so it can be reset after you have completed the rollback operation.
2. Use the `admin:database-set-merge-timestamp` function to set the merge timestamp to any time before the backup was taken. This will preserve fragments in merge after this timestamp until you have rolled back your forest data.
3. Use the `xdmp:database-restore` function with `$journal-archiving` set to `true` and `$restoreToTime` set to `null ()` to restore the database to the latest timestamp.
4. After the restore operation has completed, use point-in-time queries described in the [Point-In-Time Queries](#) in the *Application Developer’s Guide* to determine the time at which the sample documents last looked correct.
5. Use the `xdmp:forest-rollback` function to roll back the forests to the timestamp used for the successful point-in-time queries. For example, if you are restoring the Documents database and the documents at the timestamp `2011-09-13T10:57:25.201832-07:00` look correct, your `xdmp:forest-rollback` function call would be:

```
xdmp:forest-rollback(
  xdmp:database-forests(xdmp:database("Documents")),
  xdmp:wallclock-to-timestamp(
    xs:dateTime("2011-09-13T10:57:25.201832-07:00")))
```

6. Use `admin:database-set-merge-timestamp` function to set the merge timestamp back to the value you saved in [Step 1](#).

19.5.8. Restoring a Reconfigured Database

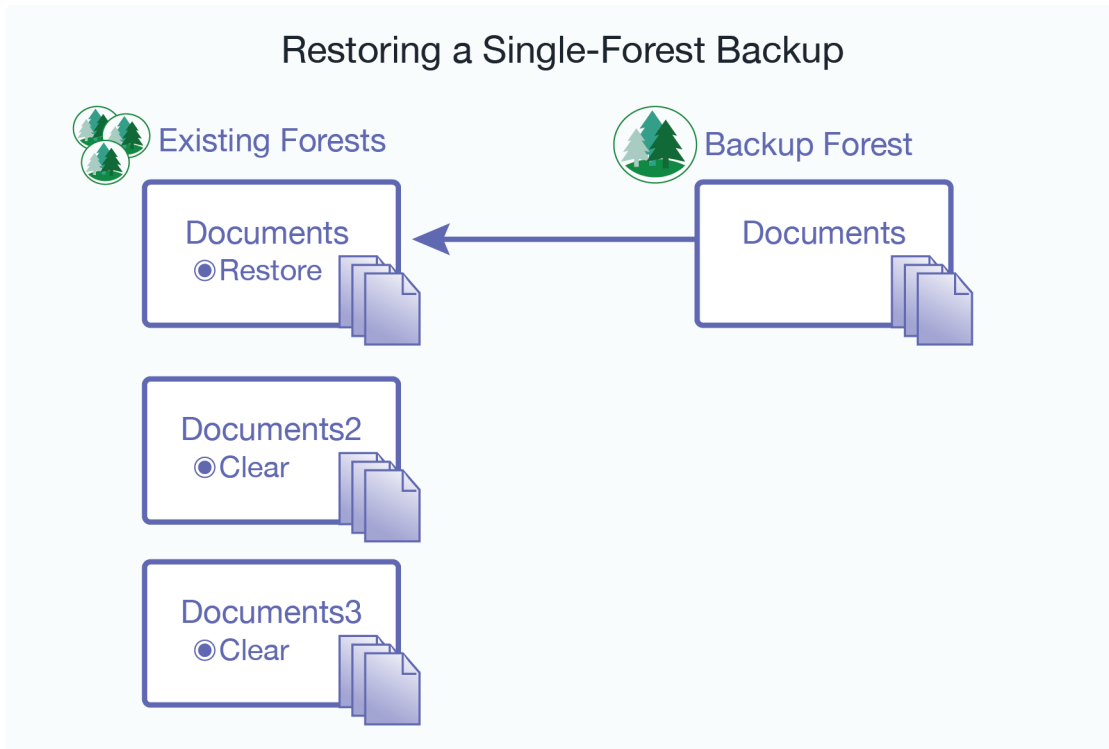
You can restore a database from a backup, even if forests have been added to or subtracted from the database after the backup. When the number of database forests are asymmetrical to the backup forests, these mapping rules apply:

- Restore a single database forest from a single backup forest.
- Restore a single database forest from multiple backup forests.

When restoring a database that has added or subtracted forests since the backup, click on the **Backup/Restore** tab, go to the Restore section of the page and set the **Forest topology changed** field to **true** and click **OK**.

The confirmation page appears showing the existing forests for the database and the backed up forests.

For example, you want to restore from a backup that was done when the **Documents** database had only one forest (**Documents**) and the restore operation is done after adding two more forests (**Documents2** and **Documents3**) to the **Documents** database. You can only restore a backup forest to a single existing forest. In this example, we are populating the **Documents** forest from the backup of the **Documents** forest.



The **Confirm Restore** page below shows the restore operation. This operation is restoring the **Documents** forest from the **Documents** backup forest. To ensure that the **Documents** database is restored with the data from the backup, set the **Documents2** and **Documents3** forests to **clear** to remove any data added since the backup.

Confirm restore

Confirm that you want to restore the following forests to database **Documents** from directory `/ml_local/backupTest612`

Documents	<input checked="" type="radio"/> restore <input type="radio"/> clear <input type="radio"/> no change	From Backup Forest: Documents ▼	More Forests
Documents3	<input type="radio"/> restore <input checked="" type="radio"/> clear <input type="radio"/> no change	From Backup Forest: Documents ▼	More Forests
Documents2	<input type="radio"/> restore <input checked="" type="radio"/> clear <input type="radio"/> no change	From Backup Forest: Documents ▼	More Forests

Incremental Backup: **false**

Use Journal Archive: **false**

RestoreToTime:

Backup was completed: **2023-06-12T08:35:39**

Server version used: **11.0.2**

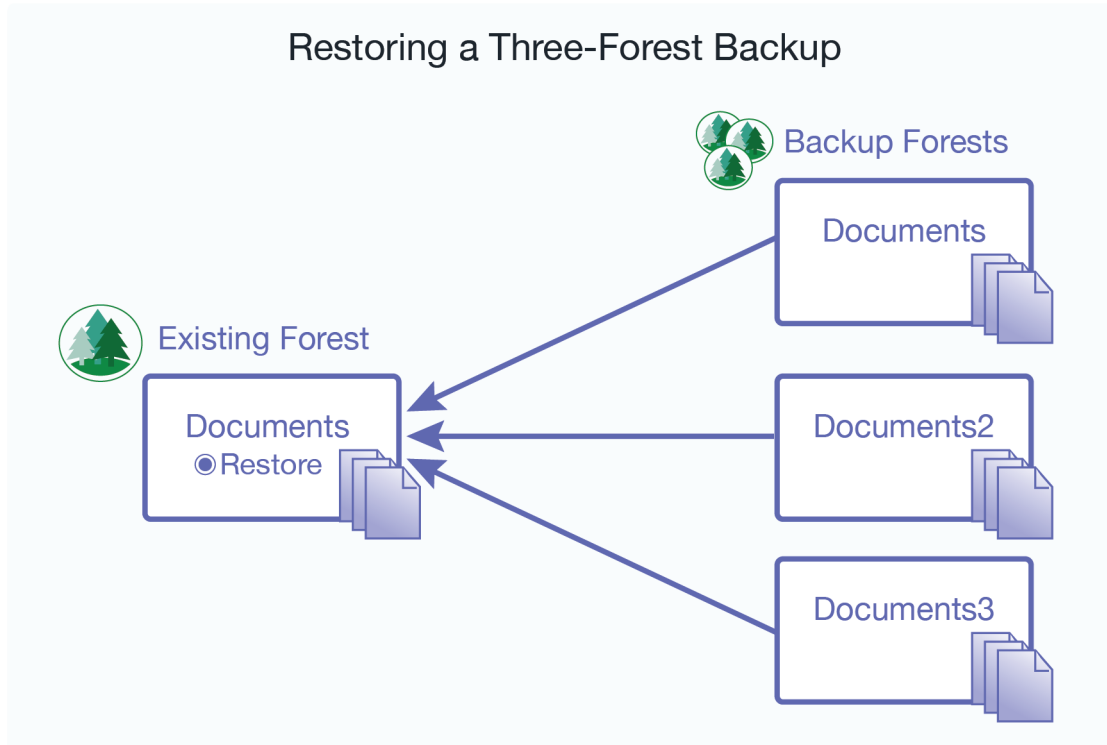
Cancel
OK

The restore options for each existing forest are:

Setting	Description
restore	Restore forest from backup forest.
clear	Do not restore forest and clear any data from the existing forest.
no change	Do not restore forest and leave the contents of the existing forest unchanged.

To restore from a backup that contains more than one forest, click the **More Forests** button and chose the additional backup forests from the pull down menus, as shown:

For example, you want to restore from a backup that was done when the **Documents** database had three forests, **Documents**, **Documents2**, and **Documents3** and the restore operation is done after deleting the **Documents2** and **Documents3** forests. In this example, we are populating the singular **Documents** forest from the **Documents**, **Documents2**, and **Documents3** backup forests.



To restore a database that has added or subtracted forests since the backup, along with the auxiliary databases (**Security**, **Schemas**, and **Triggers**); click on the **Backup/Restore** tab; go to the **Restore** section of the page; enable the **Forest topology changed** and **Include auxiliary databases** options, and click **Ok**.



NOTE

The **Include auxiliary databases** option is only relevant when **Forest topology changed** is enabled.

19.6. Backing Up and Restoring a Database Following Local Disk Failover

Following a failure of a host that contains a master forest configured for local disk failover, the database attached to the master forest fails over to the replica forest. This section describes how to back up the surviving replica forest data and restore the data after the host containing the master forest has been restored. In the example procedure described in this section, the `Documents` database is attached to the `Documents-master` forest on one host and is configured for local-disk failover to the `Documents-rep` forest on another host.

For details on how to configure local disk failover, see the [Configuring Local-Disk Failover for a Forest](#) section in the *Scalability, Availability, and Failover Guide*.

The procedure in this section is based on this scenario:

1. Before the failure, the Documents-master forest is in the open state and the Documents-rep forest is in the sync replicating state:

Forest	Host	State	Documents
Documents-master	gordon-3.marklogic.com	open	29,001
Documents-rep	gordon-3.marklogic.com	sync replicating	29,001

2. A failure occurs on the host containing the Documents-master forest and the Documents database automatically fails over to the Documents-rep forest. The Documents-rep forest is now in the open state and servicing updates on behalf of the Documents database.



NOTE

The configuration of the Documents database remains unchanged from before the failover:

Forest	Host	State	Documents
Documents-master	This forest has an error		
Documents-rep	gordon-3.marklogic.com	sync replicating	29,001

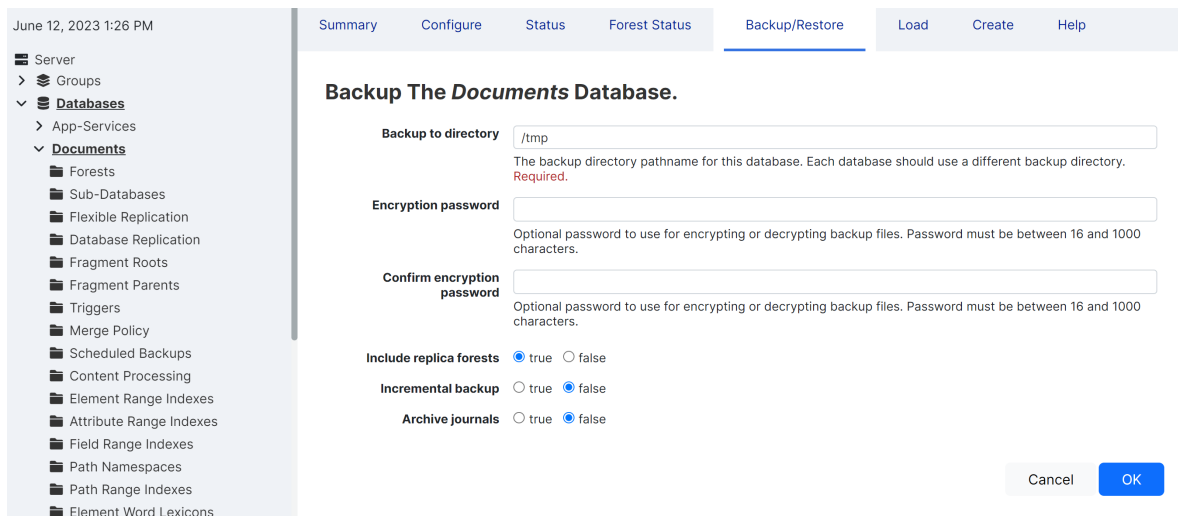
To back up the Documents-rep forest, follow these steps:



NOTE

Both the backup and restore procedures must be done on the host that contains the Documents-rep forest.

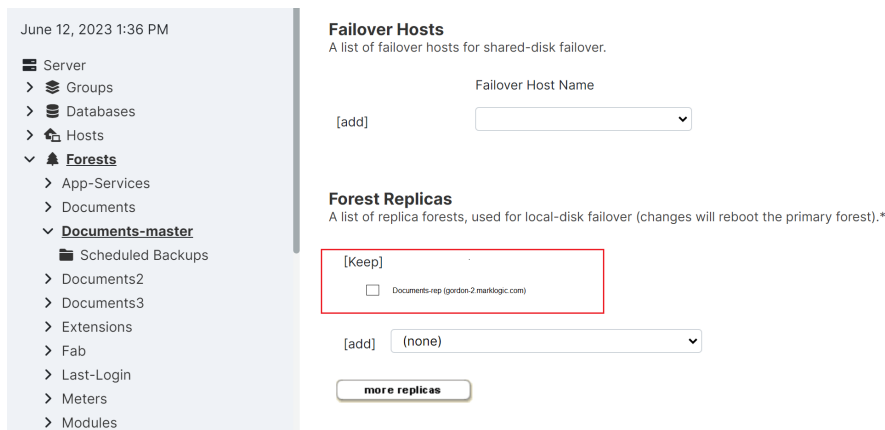
1. On the host machine that contains the Documents-rep forest, back up the Documents database. Leave Include Replica Forests set to true.



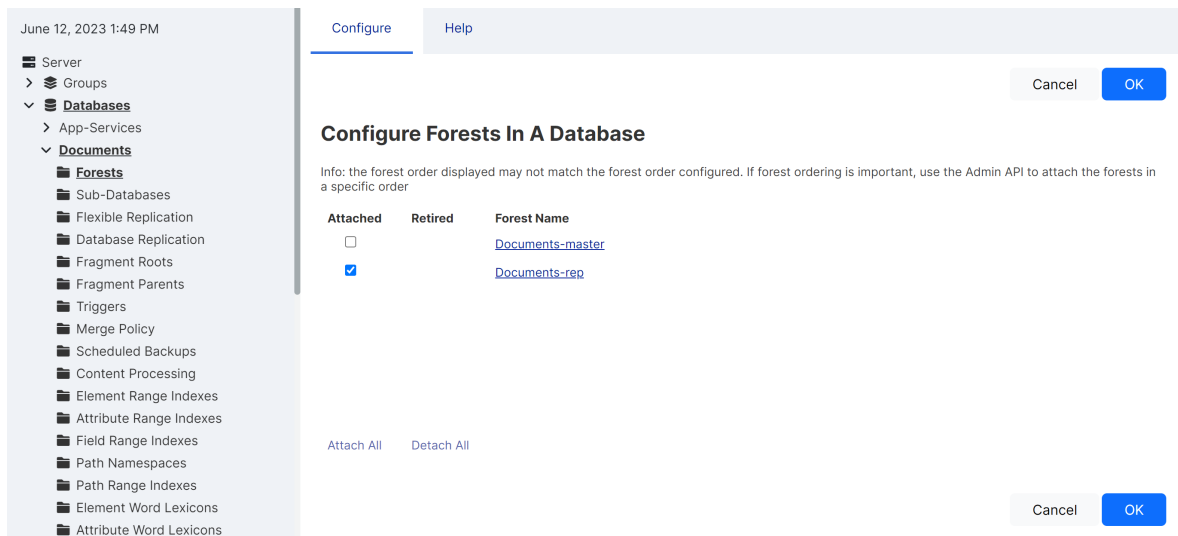
2. Select only the Documents-rep forest for backup.
3. Once the host containing the Documents-master forest is restored, the Documents-master forest becomes the replica forest and receives replicated updates from the Documents-rep forest.

Before you can restore data from the Documents-rep forest that you backed up after the failover, you must reconfigure local disk failover from the Documents-rep forest to the Documents-master forest, so that the Documents-master forest is the new replica forest:

1. In the configuration page Documents-master forest, disable replication to Documents-rep forest.



2. Navigate to the Forests configuration page for the `Documents` database. Unattach the `Documents-master` forest and attach the `Documents-rep` forest.



3. Navigate to the configuration page for the `Documents-rep` forest and select the `Documents-master` forest for local-disk failover.
4. On the host containing the `Documents-rep` forest, confirm that the forest is in the `open` state and restore the `Documents` database from the backup taken after the failover.
5. Make sure only the `Documents-rep` forest is selected for restoration.
6. Once the `Documents-rep` forest is restored, the updates are replicated to the `Documents-master` forest.

Forest	Host	State	Documents
Documents-master	gordon-3.marklogic.com	sync replicating	29,001
Documents-rep	gordon-3.marklogic.com	open	29,001

20. Rolling Upgrades

Users with highly available clusters under heavy transaction loads may want to upgrade to a newer version of MarkLogic in a seamless manner. A rolling upgrade, where hosts in a cluster are upgraded one by one, is one approach to addressing this need. Rolling upgrades are used to upgrade a large cluster with many hosts to a newer version of MarkLogic Server without incurring any downtime in availability or interruption of transactions. A rolling upgrade may also be used to apply patches to multiple hosts.

The goal in performing a rolling upgrade is to have zero downtime of your server availability or transactional data. This is most useful for large high availability (HA) clusters that have a large number of ongoing transactions. A rolling upgrade can be performed on both a primary cluster and a disaster recovery (DR) cluster.



NOTE

Do not change your application to take advantage of any new features until all the nodes in your cluster have been upgraded. In addition, you should avoid making any configuration changes to your cluster during a rolling upgrade.

This section describes rolling upgrades.

20.1. Understanding Rolling Upgrades

A rolling upgrade incrementally installs a later version of MarkLogic Server (host by host), rather than having to take down the whole cluster to install the newer version. Performing a rolling upgrade means that your cluster may be in a mixed state (where more than one version of MarkLogic Server is running) for some period of time during the upgrade process. During the process, the features in the newer version of MarkLogic will not be available until the whole cluster has been committed to the new version. Because of this you may have to change or modify some of your application code prior to starting the rolling upgrade, so that code will work in a mixed environment. For example, JavaScript code may need modification (because of the new version of V8 for server-side JavaScript) before you commit the upgrade.

The primary security, schema, and auxiliary forests must be on the same host, and that host should be the first host you upgrade when upgrading a cluster.



NOTE

In a mixed node cluster, before the upgrade has been committed, the node that has been upgraded will be read-only. This only applies to server configuration and not content databases. This is to prevent any configuration changes from the node. We strongly recommend that you not make any configuration changes until you have finished upgrading the entire cluster.

20.1.1. When Cluster Has Nodes at Different Software Version Levels

The rolling upgrade feature is designed to enable business continuity while a cluster is being upgraded. The window of time when a cluster has nodes of varying versions should be small. During this time, do not make application code changes and/or configuration changes.

Configuration changes involve the following:

- Changes to index, forest, database, application server, host, group, and cluster setting
- Changes to security settings such as adding/changing/deleting roles, users, privileges, credentials, certificates, etc.
- Adding/removing/updating TDE templates
- Adding/removing/updating redaction rules

In addition, do not perform any manual merges and disable reindexing while the cluster has nodes that are at different software version levels. Changing error log settings and adding trace events to debug issues should be fine.

20.1.2. Rolling Upgrade Process

You can upgrade your cluster with a minimal amount of transactional downtime (less than 5-10 minutes) without using the rolling upgrade feature. Consider whether the tradeoff in added complexity warrants using rolling upgrades instead of the regular upgrade process. See [Upgrading from Previous Releases](#) in the *Installation Guide* for information about the regular upgrade process.

Here are the steps in the rolling upgrade process:

- Backup - back up any hosts that you are going to upgrade. See [Section 19, “Backing Up and Restoring a Database” \[160\]](#) for details.
- Preparation - prepare any code or application that you may need to prior to the upgrade. See [Section 20.6, “Interaction with Other MarkLogic Features” \[195\]](#) for details.
- Upgrade - perform the actual upgrade.
- Cleanup

Before you start your upgrade, you will need to backup the hosts you are going to upgrade. Then do any preparation of code or applications that is necessary prior to the upgrade. See [Section 20.6, “Interaction with Other MarkLogic Features” \[195\]](#) for possible preparations.

When you have completed the upgrade, you may need to perform some clean up.

20.1.3. Rolling Upgrade Status in the Admin Interface

To view the rolling upgrade status in the Admin Interface, follow these steps:

1. Click **Server** in the left tree menu.
2. Click the **Upgrade** tab to view the upgrade progress of each host in the cluster. The displayed version number is the highest version number on any host in the cluster.



NOTE

If a rolling upgrade is in progress, a Sync icon will appear to the right of the version number from which you are upgrading, which is located above the left tree menu. Click the **Sync** icon to navigate to the Upgrade tab.

If a rolling upgrade is not in progress, all hosts in the cluster are running the same version. Click the **Upgrade** tab to verify the version number.

20.1.4. Effective Version and Software Version

Until you commit the upgrade, the *effective version* of the hosts in the cluster is the earlier version, not the newer version. The effective version is the version that the cluster as a whole is running. The *software version* is the version of MarkLogic Server that is installed on each host. You will be prompted to upgrade the Security database when you log in to the Admin Interface.



NOTE

After committing a rolling upgrade, you can only restore to the later version, not to the earlier version. Running your cluster in an uncommitted state is equivalent to running in the previous (earlier) version of MarkLogic.

An upgrade of the Security database is required after you have committed the new version of MarkLogic.

20.2. Example—Rolling Upgrade

The following procedure is a simplified, step-by-step process for a rolling upgrade on a small, three-host cluster. Here is the general outline: 1) Back up all of your hosts; 2) Make any changes to software applications; 3) Proceed with the rolling upgrade, failing over and upgrading each node; 4) Verify that you can commit the upgrade; 5) Change the cluster effective version to the new version; and 6) Do any necessary cleanup.

In addition, prior to starting the upgrade, you may need to modify some of your existing software to run in a mixed version cluster. See [Interaction with Other MarkLogic Features](#) for details.



NOTE

- When an OS upgrade is required, perform a separate rolling upgrade from the upgrade of the MarkLogic Server.
- Prior to any activity, for clusters configured with local disk failover, the entire system should be restored to the state in which primary forests are acting as primary forests and replica forests are acting as replica forests and in which replica forests are all in the mount state of `sync replicating`. For clusters configured with shared disk failover, it is recommended that forests be mounted on their primary hosts.

To perform the rolling upgrade, follow these steps:

1. Back up all hosts in your existing cluster. See [Backing Up and Restoring a Database](#) for details on backing up your hosts.
2. Modify any code that needs to be modified. See [Interaction with Other MarkLogic Features](#) for a list of potential software issues.
3. Start the rolling upgrade on the host that contains your primary security and schema forests.
4. Configure your load balancer to avoid sending further transactions towards the host to be upgraded.
5. Wait for all existing and queued transactions towards the target host to complete before proceeding with the next steps.

6. [UPGRADING THE FIRST NODE ONLY] Trigger the forest failover for your security, schema, and other auxiliary forests.

You can use this API:

```
curl -X POST --anyauth --user admin:admin -d "state=restart" "http://node1:8002/
manage/v2/forests/Security"
curl -X POST --anyauth --user admin:admin -d "state=restart" "http://node1:8002/
manage/v2/forests/Schemas"
```

Using this API prioritizes the failover of your security and schema forests over the failover of your content forests. This priority order minimizes the impact of the time needed to remount the replica auxiliary forests.



NOTE

- While your replica security forest is remounting, no one can authenticate.
- While your replica schema forest is remounting, no TDEs, redaction, etc. are available.
- While your replica trigger forest is remounting, no pre- or post-commit triggers can occur.
- While your replica module forest is remounting, no data services, REST API extensions, etc. are available.

7. Take down the host and start the upgrade:
- Stop MarkLogic. Use this cURL command so that you can also take advantage of the fast failover feature:

```
curl -X POST --anyauth --user admin:admin -d "state=shutdown&failover=true"
"http://node1:8002/manage/v2/hosts/node1"
```



NOTE

"Fast" does not mean "instantaneous." It will still take some time to remount the replica forests as primary.

- Uninstall the existing RPM:


```
rpm uninstall [rpm filename]
```
 - Install the new RPM:


```
rpm install [rpm filename]
```
 - Bring the host back up, and start MarkLogic:


```
sudo /sbin/service MarkLogic start
```
8. Wait for the forests on this node to catch up with replication. Local forests' mount state should be in sync replicating:

This command

```
curl --anyauth --user admin:admin "http://{host}:8002/manage/LATEST/forests/{id|name}?
view=status&format=json"
```

should return a result including this property:

```
{
  ...
  "status-properties": {
    "state": {
      "units": "enum",
      "value": "sync replicating"
    }
  }
  ...
}
```

9. [IMMEDIATELY AFTER UPGRADING FIRST NODE] Trigger the forest failover for your replica security, schema, and other auxiliary forests that are acting as primary using this REST API:

```
curl -X POST --anyauth --user admin:admin -d "state=restart" "http://node1:8002/
manage/v2/forests/Security-replica"
curl -X POST --anyauth --user admin:admin -d "state=restart" "http://node1:8002/
manage/v2/forests/Schemas-replica"
```

Using this API prioritizes the failover of your security and schema forests over the failover of your content forests. This priority order minimizes the impact of the time needed to remount the primary auxiliary forests.



NOTE

- While your primary security forest is remounting, no one can authenticate.
- While your primary schema forest is remounting, no TDEs, redaction, etc. are available.
- While your primary trigger forest is remounting, no pre- or post-commit triggers can occur.
- While your primary module forest is remounting, no data services, REST API extensions, etc. are available.

10. Repeat [Step 3 - Step 8](#) for each of the hosts in the cluster. (You will need to perform the upgrade process node by node.)
11. Trigger a failover for the replica forests that are acting as primary, especially the security and schema forests:

```
curl -X POST --anyauth --user admin:admin -d "state=restart" "http://{nodeX}:8002/
manage/v2/forests/{content-replica-X}"
```

12. When you have completed all of the host upgrades, check the software version and the effective version for the cluster, and then commit the upgrade:
- Use this query to check if the cluster is ready to commit the upgrade. It will return `true` when the cluster is ready:

```
xquery version "1.0-ml";
import module namespace admin = "http://marklogic.com/xdmp/admin"
  at "/MarkLogic/admin.xqy";
admin:can-commit-upgrade()
```

- Upgrade the security database on the host cluster:

```
curl -X POST --anyauth --user admin:admin \
  --header "Content-Type:application/json" \
  -d '{"operation": "security-database-upgrade-local-cluster"}' \
  "http://localhost:8002/manage/v2"
```

- After committing the upgrade, verify it by retrieving the effective software version of the cluster: Use this query:

```
xquery version "1.0-ml";
import module namespace admin = "http://marklogic.com/xdmp/admin"
  at "/MarkLogic/admin.xgy";
let $config := admin:get-configuration()
return
  admin:cluster-get-effective-version($config)
```

Or use this cURL command:

```
curl -X POST --anyauth --user admin:admin -d "state=shutdown&failover=true"
"http://node1:8002/manage/v2/clusters/{cluster-name}?format=json"
```

The cluster version, returned in the property `effective-version`, should match the intended version. For example, if your target version is 10.0-5, make sure that the `effective-version` is 10000500.

20.3. Performing Rolling Upgrades

You can perform a rolling upgrade via scripting through the REST Management APIs or by using the XQuery APIs. You can also perform a rolling upgrade on an AWS cluster. This section describes the different options for configuring and performing a rolling upgrade.

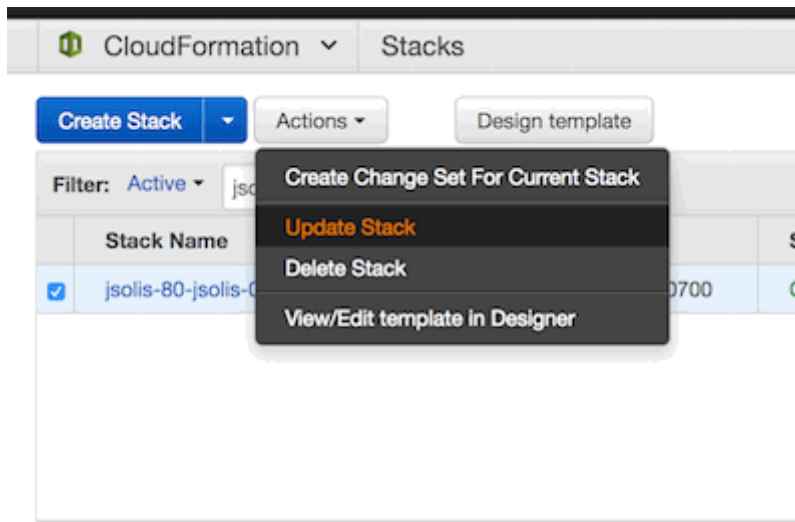
20.3.1. Upgrading an EC2 Instance

The process for performing a rolling upgrade in EC2 (AWS) is fairly simple. It is very similar to a normal update of the Cloud Formation templates. See [Upgrading MarkLogic on AWS](#) in the *MarkLogic Server on Amazon Web Services (AWS) Guide* for details about a normal update.

This example assumes an existing 3-node cluster from Cloud Formation templates. Before you upgrade your instance, you need to upgrade your Cloud Formation template to reference the new AMI (9.0 CF template). See [MarkLogic on Amazon Web Services \(AWS\)](#) for details about upgrading your templates.

Here are the additional steps:

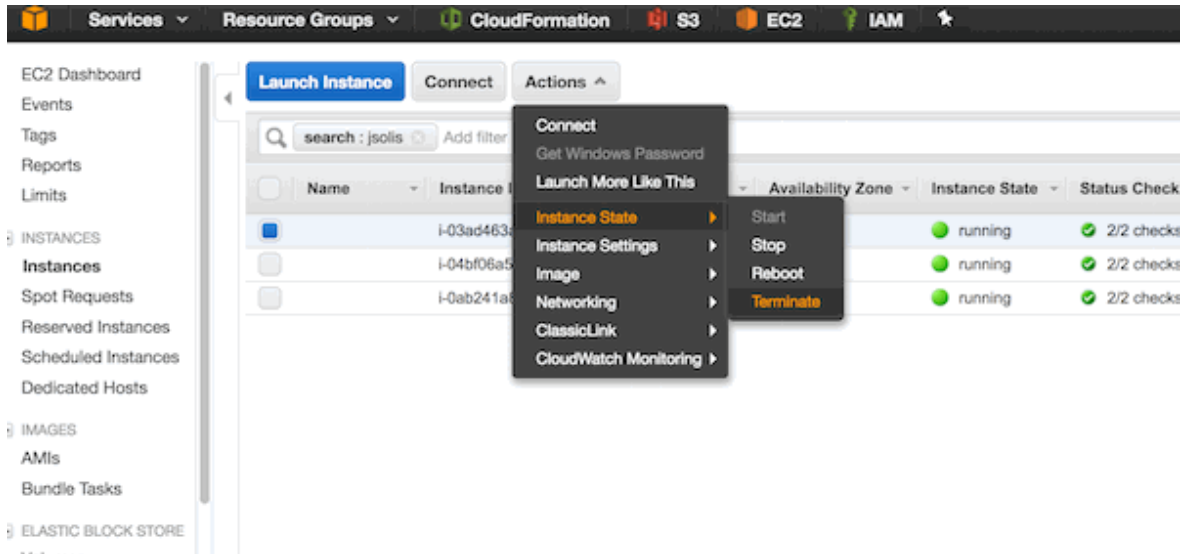
1. Back up any important data before performing the upgrade.
2. Update the stack with your updated Cloud Formation template. Make sure the stack update is complete.



NOTE

We do not recommend that you automatically swap out the Cloud Formation template. Instead, make a copy of your existing template (if it contains the AMI IDs), edit just the AMI IDs, and then use that for the update. (If the AMI ID is passed as a parameter or other means, use those means).

3. In the EC2 dashboard, terminate one instance at a time and wait for it to be replaced with a new one. Starting with the "master" instance or node that contains the Security database. The host will automatically be restarted by the managed cluster feature.

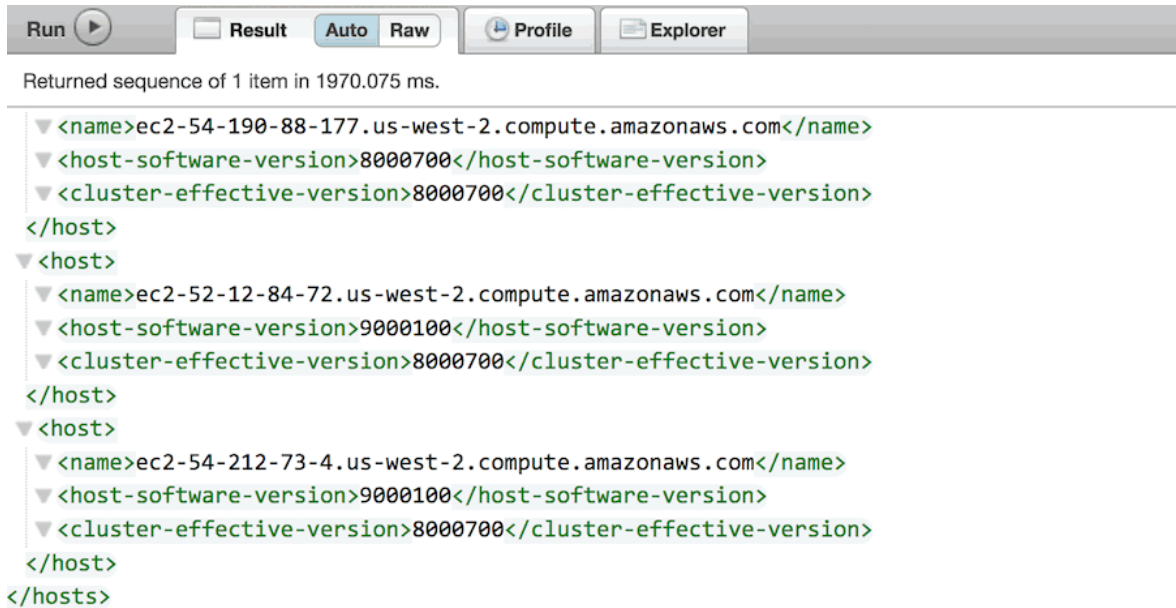


Wait for the host to come back up (with new host name).

4. In the EC2 dashboard, repeat the process and terminate node2.
5. When all nodes have been updated, check the cluster state from the Query Console using this query:

```
xquery version "1.0-m1";

<hosts>{
  for $i in xdm:host-name(xdm:hosts())
  return (
    let $response := xdm:http-get(concat("http://localhost:8002/manage/v2/hosts/",
    $i,"?view=status&format=json"),
    <options xmlns="xdmp:http">
      <authentication method="digest">
        <username>admin</username>
        <password>admin</password>
      </authentication>
      <headers>
        <content-type>application/json</content-type>
      </headers>
    </options>)
    return (
      <host>
        <name>{$response[2]/*:name/data()}</name>
        <host-software-version>
          {$response[2]/*:software-version/value/data()}
        </host-software-version>
        <cluster-effective-version>{$response[2]/*:effective-version/value/data()}
        </cluster-effective-version>
      </host>
    ) ) }</hosts>
```



```

Returned sequence of 1 item in 1970.075 ms.
<hosts>
  <host>
    <name>ec2-54-190-88-177.us-west-2.compute.amazonaws.com</name>
    <host-software-version>8000700</host-software-version>
    <cluster-effective-version>8000700</cluster-effective-version>
  </host>
  <host>
    <name>ec2-52-12-84-72.us-west-2.compute.amazonaws.com</name>
    <host-software-version>9000100</host-software-version>
    <cluster-effective-version>8000700</cluster-effective-version>
  </host>
  <host>
    <name>ec2-54-212-73-4.us-west-2.compute.amazonaws.com</name>
    <host-software-version>9000100</host-software-version>
    <cluster-effective-version>8000700</cluster-effective-version>
  </host>
</hosts>

```

6. Make a call from 8001 to check host status. The URL pattern to check this is `http://{ec2-hostname}:8001/host-summary.xqy?section=host`.
7. Repeat [Step 1](#) - [Step 6](#) for node3.
8. When the node3 update is complete, check to verify that the upgrade is complete by checking the cluster effective version:

```

xquery version "1.0-ml";

<hosts>{
  for $i in xdmp:host-name(xdmp:hosts())
  return (
    let $response := xdmp:http-get(concat("http://localhost:8002/manage/v2/hosts/",
    $i,"?view=status&format=json"),
    <options xmlns="xdmp:http">
      <authentication method="digest">
        <username>admin</username>
        <password>admin</password>
      </authentication>
      <headers>
        <content-type>application/json</content-type>
      </headers>
    </options>)
    return (
      <host>
        <name>{$response[2]//*:name/data()}</name>
        <host-software-version>
        {$response[2]//*:software-version/value/data()}
        </host-software-version>
        <cluster-effective-version>
        {$response[2]//*:effective-version/value/data()}
        </cluster-effective-version>
      </host>
    )
  )
}</hosts>

```

```

Returned sequence of 1 item in 502.059 ms. (-4892.118 ms. compared to previous run)

<hosts>
  <host>
    <name>ec2-54-190-88-177.us-west-2.compute.amazonaws.com</name>
    <host-software-version>9000100</host-software-version>
    <cluster-effective-version>9000100</cluster-effective-version>
  </host>
  <host>
    <name>ec2-52-12-84-72.us-west-2.compute.amazonaws.com</name>
    <host-software-version>9000100</host-software-version>
    <cluster-effective-version>9000100</cluster-effective-version>
  </host>
  <host>
    <name>ec2-54-212-73-4.us-west-2.compute.amazonaws.com</name>
    <host-software-version>9000100</host-software-version>
    <cluster-effective-version>9000100</cluster-effective-version>
  </host>
</hosts>

```

9. Navigating anywhere in the Admin Interface will prompt you to upgrade your Security database. To upgrade the Security database, go to `http://{ec2-hostname}:8001/security-upgrade.xqy`. When that has been done, then the upgrade is complete.

20.3.2. Rolling Upgrades Using XQuery

The XQuery Admin APIs can be used to set up and perform a rolling upgrade through the Query Console. This section contains sample code that you can use from the Query Console.

To get the host versions via REST, use this code:

```

xquery version "1.0-m1";

<hosts>{
  for $i in xdmp:host-name(xdmp:hosts())
  return (
    let $response := xdmp:http-get(concat("http://localhost:8002/manage/v2/hosts/",
    $i,"?view=status&format=json"),
    <options xmlns="xdmp:http">
      <authentication method="digest">
        <username>admin</username>
        <password>admin</password>
      </authentication>
      <headers>
        <content-type>application/json</content-type>
      </headers>
    </options>)
    return (
      <host>
        <name>{$response[2]//*:name/data()}</name>
        <host-software-version>{$response[2]//*:software-version/value/data()}</host-software-
        version>
        <cluster-effective-version>{$response[2]//*:effective-version/value/data()}</cluster-
        effective-version>
      </host>
    )
  )
}
</hosts>

```

To complete the upgrade, log onto the Admin Interface to upgrade the Security database.

**NOTE**

Committing the upgrade results in the updated configuration being saved with a re-read delay of 5 seconds to ensure that all online hosts have received the new file before XDQP connections start dropping.

See [Step 9](#) in [Section 20.3.1, “Upgrading an EC2 Instance” \[191\]](#). If the servers don’t have the correct version, there may be a host that is in maintenance mode. The `admin:can-commit-upgrade` function will return `true` if all servers have the correct software version. See [Section 20.5.1, “Admin APIs” \[195\]](#) for more about the XQuery Admin APIs available.

20.3.3. Rolling Upgrades on Both Production and DR Clusters

Upgrade the disaster recovery cluster first. It is important to upgrade the disaster recovery cluster first, since the newer version of the software will be able to receive fragments and journal frames encoded on the master cluster.

Once the disaster recovery cluster has been upgraded, then upgrade the production cluster.

20.4. Rolling Back a Partial Upgrade

As long as you have not committed your upgrade, you can reinstall the earlier version of the server on each node.

In the event that you need to roll back an upgrade that has not been completed and committed, you can roll back the partial upgrade by re-installing the previous version of MarkLogic on the machines that have been upgraded.

20.5. APIs for Rolling Upgrades

The APIs in this section are available for managing rolling upgrades in a MarkLogic cluster.

20.5.1. Admin APIs

These Admin API functions are available for rolling upgrades:

- `admin:cluster-get-effective-version`
Returns the cluster’s effective MarkLogic version.
- `admin:can-commit-upgrade`
Returns `true` if the cluster is ready to commit the upgrade, returns `false` otherwise.

20.5.2. REST Management APIs

The following REST Management endpoints provide useful information and functionality when performing a Rolling Upgrade operation.

- `GET:/manage/v2/properties` - includes effective version
- `GET:/manage/v2/hosts?view=status` - includes version and effective-version.
- `GET:/manage/v2/hosts/{id|name}?view=status` - includes version and effective-version.

20.6. Interaction with Other MarkLogic Features

For existing features that will work as expected in the old version, a rolling upgrade will not have any impact. Some existing features may not work as expected until the rolling upgrade is complete and the cluster has been committed to the newer version.

20.7. Important Points to Note before Performing Rolling Upgrades

- The cluster needs to be configured for High Availability (HA). This includes configuring replica forests for your security, schema, and other auxiliary forests.
- Do not change your application to take advantage of features in this version until all the nodes in your cluster have been upgraded
- You may have to change or modify some of your application code prior to starting the rolling upgrade, so that code will work in a mixed environment.
- The primary security, schema, and auxiliary forests must be on the same host, and that host should be the first host you upgrade when upgrading a cluster.
- In a mixed node cluster, before the upgrade has been committed, the node that has been upgraded will be read-only. This is to prevent any configuration changes from that node. We strongly recommend that you not make any configuration changes until you have finished upgrading the entire cluster.
- The window of time when a cluster has nodes of varying versions should be small. During this time, do not make application code changes and/or configuration changes. Configuration changes involve the following:
 - Changes to index, forest, database, application server, host, group, and cluster settings
 - Changes to security settings such as adding/changing/deleting roles, users, privileges, credentials, certificates, etc.
 - Adding/removing/updating TDE templates
 - Adding/removing/updating redaction rules
- Do not perform any manual merges and disable reindexing while the cluster has nodes that are at different software version levels. Changing error log settings and adding trace events to debug issues should be fine.
- After committing a rolling upgrade, you can only restore to the later version, not to the earlier version. Running your cluster in an uncommitted state is equivalent to running in the previous (earlier) version of MarkLogic. New features are not available until the upgrade has been committed.
- An upgrade of the security database is required after you have committed the new version of MarkLogic.
- It is important to upgrade the disaster recovery cluster first, since the newer version of the software will be able to receive fragments and journal frames encoded on the master cluster.
- Before stopping the MarkLogic process on a node, configure your load balancer to divert transactions away from the node being upgraded, and allow all existing and queued transactions to complete.
- Mounting a forest takes time. Prioritize the failover of forests for your security, schema, and other auxiliary databases.
- Before stopping the MarkLogic process on a node, make sure that all acting replica forests on the other nodes have caught up in replicating content.

20.8. Other Upgrade Options

There are alternatives to rolling upgrades for applying patches or upgrading your hosts. You can perform an upgrade to hosts in a cluster with very minimal downtime. See [Upgrading from Previous Releases](#) in the *Installation Guide* for more information.

21. Hosts

A host is an instance of MarkLogic Server. A host is not configured individually but as a member of a group. A host is added to the *Default* group if it is not joined to another group during the installation process. For example, in cases where MarkLogic is running in a single host environment, the host is added to the *Default* group.

Forests are created on hosts and added to a database to interact with HTTP, ODBC, and XDBC Servers running on the same or other hosts.

See [Section 5, “Groups” \[34\]](#) and [Section 12, “Databases” \[81\]](#) for more details on hosts as they relate to groups and databases.

A host is managed from both the **Group** and **Hosts** configuration screens.

Using the Admin Interface to manage hosts is covered here. Managing hosts programmatically is covered in [Host Maintenance Operations](#) in the *Scripting Administrative Tasks Guide*.

21.1. Adding a Host to a Cluster

This only applies for multi-host clusters. For information about installing MarkLogic and a more detailed procedure about joining a cluster, see the [Installation Guide](#).

To add a host to a cluster, follow these steps in the Admin Interface:

1. Install MarkLogic Server on the host if it is not already installed.
2. Start MarkLogic Server.
3. Access the Admin Interface on the host in which you want to add to the cluster and accept the license agreement.
4. After the server restarts, you will be prompted to join a cluster.
5. Enter the DNS name or the IP address of one of the machines in the cluster. For example, if this is the second host you are installing, you can enter the DNS name of the first host you installed.
6. You will be prompted for an admin username and password. Enter the admin username and password for the security database used by the cluster. Click **OK**.
7. Select a Group to assign this host. Click **OK**.
8. Click **OK** to confirm that you are joining the cluster.
9. Click **OK** for the confirmation message that indicates that you have joined the cluster.

21.2. Changing the Group of the Host

To change the group to which a host belongs, follow these steps in the Admin Interface:

1. Click the **Hosts** icon in the left tree menu.
2. Click the name of the host you want to change, either on the tree menu or the summary page.
3. Select from the available groups in the Group drop-down menu.
4. Click **OK** to confirm the change.

Changing the group to which a host belongs is a “cold” task; the server restarts to reflect the changes.

21.3. Shutting Down or Restarting a Host

To shut down or to restart a host, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Users**.
3. Click the **Status** tab at the top right.

4. Click the **Shutdown** or the **Restart** button as appropriate.
5. Click **OK** to confirm to confirm the shutdown or restart operation.
6. If you have forest failover enabled for any of the host forests, you will see a **Immediately fail over forests to replica hosts** option. Check the box to fail over the forests to replica hosts.



NOTE

The restart operation normally completes within a few seconds. It is possible, however, for it to take longer under some conditions (for example, if the Security database needs to run recovery or if the connectivity between hosts in a cluster is slow). If it takes longer than a few seconds for MarkLogic Server to restart, than the Admin Interface might return a `503: Service Unavailable` message. If you encounter this situation, wait several seconds and then reload the Admin Interface.

21.4. Clearing a Forest on a Host

Clearing a forest on a host permanently deletes the data in the forest. The configuration information of the forest will be preserved. For example, you may want to clear the forest if you want to load new data into the same configuration.

To clear the data from a forest, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Users**.
3. Click **Forests** under the selected host.
4. Click the **Clear** button corresponding to the forest you want to clear.
5. Click **OK** to confirm clearing the data from the forest.

21.5. Deleting a Forest on a Host

Deleting a forest on a host permanently deletes the data in the forest as well as the configuration information. A forest cannot be deleted if it is still attached to a database. You must first detach the forest from the database before you can delete from a host.

Assuming that the forest is not attached to any database, follow these steps to delete a forest from a host:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Users**.
3. Click **Forests** under the selected host.
4. Click the **Delete** button corresponding to the forest you want to delete.
5. Click **OK** to confirm deleting the forest from the host.
6. Click the **Delete** button.
7. Click **OK** to confirm dropping the host.

Deleting a host is a “hot” admin task for the other hosts in the group.

21.6. Leaving the Cluster

A host has to leave a cluster before being moved to another cluster. Leaving a cluster is also a way to switch a host from a single host environment to a multi-host environment or vice versa. A host cannot leave a cluster if there are still forests assigned to it or if it has any foreign clusters associated with it; you must delete all forests assigned to the host and de-couple any clusters associated with a host

before you can leave the cluster. In a single-host environment, a host cannot leave a cluster because it will always have forests assigned to it.

To make a host leave a cluster, follow these steps:

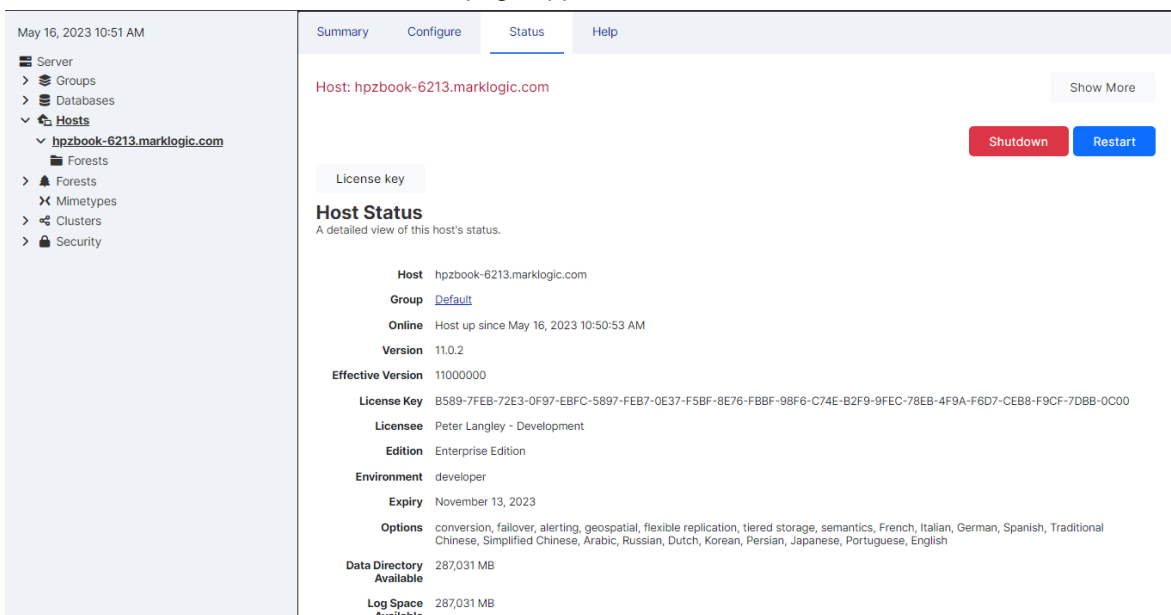
1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Users**.
3. Click the **Leave** button.
4. Click **OK** to confirm leaving the cluster. The host restarts to load the new configuration.
5. Click **OK** to self-install initial databases and application servers. A prompt to join a cluster appears.
6. To join another cluster, enter the name of one of the hosts in that cluster and click **OK**. Otherwise, click **Skip**.
7. Set up an admin user name and password if prompted.
8. Log in with the admin user name and password if prompted. The Admin Interface appears.

21.7. Displaying License Options

In addition to the features that come standard in MarkLogic, there are optionally licensed features that you may want to take advantage of for more advanced projects.

To display the license options for a host, follow these steps in the Admin Interface:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Users**.
3. Click the **Status** tab. The **Host Status** page appears:



The License Options are listed in the **Options** field:

Option	Description
Advanced Geospatial	<p>This License Option is required to use these features:</p> <ul style="list-style-type: none"> • <code>geo:complex-polygon-contains</code> or <code>geo:complex-polygon-intersects</code> APIs (polygon/polygon intersection) • Double-precision coordinates including <code>wgs84/double</code>, <code>etrs89/double</code>, or <code>raw/double</code>. • Use of <code>cts:reverse-query</code> with geospatial constraints (also known as geo alerting). <p>Other uses of Geospatial Search do not require the Advanced Geospatial License Option.</p>

Option	Description
Advanced Security	This License Option is required to use these features: <ul style="list-style-type: none"> • Compartment Security • Redaction • External key management system (KMS) or Keystore • Query-Based Access Control
Semantics	This License Option is required to use SPARQL features. Using APIs that leverage Semantics without SPARQL, such as the SQL API, do not require a Semantics Option license.
Flexible Replication	This License Option is required to use Flexible Replication.
XA	This License Option is required to use XA.
Tiered Storage	This License Option is required to use Tiered Storage.

21.8. Changing the License Key For a Host

At any time, you can change the license key for a host from the Host Status page. You might need to change the license key if your license key expires, if you need to use some features that are not covered in your existing license key, if you upgrade your hardware with more CPUs and/or more cores, if you need a license that covers a larger database, if you require different languages, or for various other reasons. Changing the license key sometimes results in an automatic restart of MarkLogic (for example, if your new license enables a new language).

To change the license key for a host, follow these steps in the Admin Interface:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Users**.
3. Click the **Status** tab. The **Host Status** page appears.
4. Click the **License key** button. The **License Key Entry** page appears.
5. Enter your new license key information. For information about licensing of MarkLogic Server, contact your MarkLogic sales representative.
6. After entering valid information in the Licensee and License Key fields, click **OK**. If it needs to, MarkLogic will automatically restart, and the new license key will take effect.



NOTE

Any optionally licensed features enabled by your license key appear in the **License Key Options** field.

21.9. Rolling Back a Transaction

Stalled or long-running transactions can be rolled back. This discards any updates made by the transaction. The **Status** tab includes a list of transactions which have started but are not yet completed.



NOTE

To rollback the MarkLogic Server portion of a prepared XA transaction, see [Section 22.13, “Rolling Back a Prepared XA Transaction Branch” \[210\]](#).

To rollback a transaction using the Admin Interface, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Users**.
3. Click the **Status** tab at the top of the page. The status view for the host appears.
4. Locate the target transaction in transaction list. If you do not see a transaction list on the status page, then there are no open transactions on this host.
5. Click **rollback** next to a transaction to initiate a transaction rollback. For example:

The screenshot shows the MarkLogic Server interface for host 'hpzbook-6213.marklogic.com'. On the left is a navigation tree with 'Security' selected. The main area displays a table of transactions:

Transaction ID	Name	State	Mode	Timestamp	Run Time	Limit	Source	User
13413518935033877365		active	query	16843327801985129	61.3 ms	600 s	Security	admin

Below the table, a detailed view of the selected transaction is shown, including a 'rollback' button highlighted in yellow. At the bottom, there is a table of aggregate capabilities:

Query Reads	Journal Writes	Save Writes	Merge Reads	Merge Writes	Backup Reads	Backup Writes	Restore Reads	Restore Writes	Large Reads	Large Writes
Rate 0	0.00621703	0	0	0	0	0	0	0	0	0

6. A confirmation dialog appears. Click **OK** to confirm the rollback.

There may be a slight delay between when a rollback is initiated and when the transaction terminates. During this period, the transaction still appears on the host status page, with a transaction status of "awaiting rollback".

22. Forests

This section describes how to use the Admin Interface to manage forests. For details on how to manage forests programmatically, see [Creating and Configuring Forests and Databases](#) and [Database Maintenance Operations](#) in the *Scripting Administrative Tasks Guide*.

22.1. Understanding Forests

A forest is a collection of XML, JSON, text, or binary documents. Forests are created on hosts and attached to databases to appear as a contiguous set of content for query purposes. A forest can only be attached to one database at a time. You cannot load data into a forest that is not attached to a database.

A forest contains in-memory and on-disk structures called *stands*. Each stand is composed of XML, JSON, binary, and/or text fragments, plus index information associated with the fragments. When fragmentation rules are in place, XML documents may span multiple stands. MarkLogic Server periodically *merges* multiple stands into a single stand to optimize performance. See [Section 15, “Understanding and Controlling Database Merges” \[108\]](#) for details on merges.

A forest also contains a separate on-disk Large Data Directory for storing large objects such as large binary documents. MarkLogic Server stores large objects separately to optimize memory usage, disk usage, and merge time. A small object is stored directly in a stand as a fragment. A large object is stored in a stand as a small reference fragment, with the full content stored in the Large Data Directory. The size threshold for storing objects in the Large Object Store and the location of the Large Object Store are configurable through the Admin Interface and Admin API. For details, see [Working With Binary Documents](#) in the *Application Developer’s Guide*.

By default, the operations allowed on a forest are: read, insert, update, and delete. You can control which operations are allowed on a forest by setting the following update types:

Update Type	Description
All	Read, insert, update, and delete operations are allowed on the forest.
delete-only	Read and delete operations are allowed on the forest, but insert and update operations are not allowed unless a forest ID is specified, in which case it results in the document being moved to another forest. If you do not specify a forest ID when updating a document in a delete-only forest, the update throws an exception. This update type is useful when you want to eliminate the overhead imposed by the merge operation, but still allow transactions to delete data from the forest. See Section 22.3, “Making a Forest Delete-Only” [204] for details.
read-only (Can only be set in Configure)	Read operations are allowed on the forest, but insert, update, and delete operations are not allowed. A transaction attempting to make changes to fragments in the forest will throw an exception. This update type is useful when you want to put your forests on read-only media and allow them to be queried. See Section 22.4, “Making a Forest Read-Only” [205] for details.
flash-backup (Can only be set in Configure)	This type puts the forest in read-only mode without throwing exceptions on insert, update, or delete transactions, allowing the transactions to retry. This update type is useful when you want to temporarily quiesce a forest or to disable changes to the forest data when doing a flash backup of the forest. See Section 22.4, “Making a Forest Read-Only” [205] for details.



NOTE

To make the entire database read-only, set all of the forests in the database to `read-only`.


22.2. Creating a Forest

To create a new forest, follow these steps:

1. Click **Forests** in the left tree menu. A list of forests appears.
2. Click your target forest.
3. Click the **Create** tab at the top right. The **Create New Forests** page appears.
4. Enter the name of your forest. Each forest name must be unique.
5. Select the host on which you want the forest to be created.
6. In the **Data Directory** field, enter the path which specifies where the forest data is stored:
 - MarkLogic recommends that you use an absolute path if you specify a data directory. If you do not specify an absolute path for the data directory, your forest will be created in the default data directory.
 - This directory should specify a location on the host’s file system with sufficient capacity to store your data.
 - The name of the forest is used by the system as a directory name. Therefore, the forest name must be a legal directory name and cannot contain any of these 9 characters: \ * ? / : < > | " . Additionally, the name cannot begin or end with a space or a dot (.).
 - The directory you specified can be an operating system mounted directory path, it can be an HDFS path, or it can be an S3 path. For details on using HDFS and S3 storage in MarkLogic, see [Disk Storage Considerations](#) in the *Query Performance and Tuning Guide*.
 - The Forests directory is either a fully-qualified pathname or is relative to the Forests directory, set at installation time based on the directory in which MarkLogic Server is installed. The following table shows the default location Forest directory for each platform:

Platform	Program Directory
Microsoft Windows	C:\Program Files\MarkLogic\Data\Forests
Red Hat Linux	/var/opt/MarkLogic/Forests
Mac OS X	~/Library/Application Support/MarkLogic/Data/Forests or ~/Library/"Application Support"/MarkLogic/Data/Forest or "~/Library/Application Support/MarkLogic/Data/Forests"

7. If you want to specify a different directory to store large objects (such as large binary documents), specify a directory in the **Large Data Directory** field. If you do not specify a large data directory, the directory specified in the **Data Dictionary** field is used. For details on binary file support, see [Working With Binary Documents](#) in the *Application Developer’s Guide*.
8. If you want to specify a high-performance directory to store the journals and as much of the forest data that will fit in this high-performance directory, specify a fast data directory. For further details on disks and the fast data directory, see [Disk Storage Considerations](#) in the *Query Performance and Tuning Guide*.
9. If you want to restrict the types of updates allowed on the field, select the types of updates you want to allow for this forest in the **Updates Allowed** field. See [Section 22.3, “Making a Forest Delete-Only” \[204\]](#) for details.



NOTE
The Read-Only update types described in [Section 22.4, “Making a Forest Read-Only” \[205\]](#) can be set in the **Configure** page of an existing forest.

10. In the **Availability** field, select `online` to make the forest data available to tiered-storage or `offline` to make the data unavailable. For details on tiered storage, see [Section 17, “Tiered Storage” \[131\]](#).
11. In the **Rebalancer Enable** field, specify whether or not you want this forest to participate in the rebalancer process for the database to which this forest is to be attached. For details on the database rebalancer, see [Section 16, “Database Rebalancing” \[118\]](#).

12. If you have enabled the database rebalancer with a document assignment policy of Range, specify the range for this forest in the **Range** field. For details on the range policy, see [Section 16.3.4, “Range Assignment Policy” \[122\]](#).
13. In the **Failover Enable** field, specify whether or not to failover this forest to another host if the primary host goes down. For details on configuring failover on a forest, see [Configuring Shared-Disk Failover for a Forest](#) in the *Scalability, Availability, and Failover Guide*.
14. Select the Failover Host from the **Failover Host Name** drop-down menu.
15. Click **OK**.

Creating a forest is a “hot” admin task; the changes take effect immediately. However, toggling between update types restarts the forest.

22.3. Making a Forest Delete-Only

You can configure a forest to only allow read and delete operations, disallowing inserts and updates to any documents stored in the forest. A delete-only forest is useful in cases where you have multiple forests in a database and you want to manage which forests change. To set a forest to only allow delete operations (and disallow inserts and updates), navigate to the configuration page for the forest you want specify as delete-only and set the `updates allowed` field to `delete-only`.

When a forest is set to delete-only, updates to documents in a delete-only forest that do not specify a forest ID will throw an exception. Updates to documents in a delete-only forest that specify one or more forest IDs of other forests in the database will result in the documents moving to one of those other forests. When a document moves forests, the old version of the document will be marked as deleted, and will be removed from the forest during the next merge.

To specify an update that will move a document in a delete-only forest to an updateable forest, you must specify the forest ID of at least one forest in which updates are allowed. One technique to accomplish this is to always specify all of the forest IDs, as in the following `xdmp:document-insert` example which lists all of the forests in the database for the `$forest-ids` parameter:

```
xdmp:document-insert($uri, $node, (), (), 0,
  xdmp:database-forests(xdmp:database()) )
```



NOTE

You can only move a document from a delete-only forest to a forest that allows updates using an API that takes forest IDs, and then by explicitly setting the forest IDs to include one or more forests that allow updates. The node-level update built-in functions (`xdmp:node-replace`, `xdmp:node-insert-child`, and so on) do not have a forest IDs parameter and therefore do not support moving documents.

Under normal operating circumstances, you likely will not need to set a forest to be delete-only. Additionally, even if the reindexer is enabled at the database level, documents in a forest that is set to delete-only will not be reindexed.

There are cases where delete-only forests are useful, however. One of the use cases for delete-only forests is if you have multiple forests and you want to control when some forests are merging. The best way to control merges in a forest is to not insert any new content in the forest. In this scenario, you can set some of the forests to be delete-only, and then those forests will not merge during that time (unless you manually specify a merge, either with the `xdmp:merge` API or by clicking the **Merge** button in the Admin Interface). After a while, you can rotate which forests are delete-only. For example, if you have four forests, you can make two of them delete-only for one day, and then make the other two delete-only the next day, switching the first two forest back to allowing updates. This approach will

only have two forests being updated (and periodically merging) at a time, thus needing less disk space for merging. For more details about merges, see [Section 15, “Understanding and Controlling Database Merges” \[108\]](#).

22.4. Making a Forest Read-Only

You can configure an existing forest to only allow reads and to disallow inserts, updates and deletes to any documents stored in the forest.

MarkLogic Server supports two read-only forest settings:

- `read-only` — When this update type is set, update transactions on the forest are immediately aborted.
- `flash-backup` — When this update type is set, update transactions on the forest are retried until either the update type is reset or the Default Time Limit set for the App Server is reached.



NOTE

Only existing forests can be set to `read-only` or `flash-backup`. You cannot create a new forest with these settings.

A read-only forest is useful if you want to put your forests on read-only media and allow them to be queried. Another use of `read-only` is to control disk space. For example, in a multi-forest database, it might be useful to be able to mark one or more forests as `read-only` as they reach disk space limits.

One use for `flash-backup` is to prevent updates to the forest during a *flash backup* operation, which is a very fast backup that can be done on some file systems. You can set the `flash-backup` update type to temporarily put the forest in read-only mode for the duration of a flash backup and then reset the update type when the backup has completed. Transactions attempting to make changes to the forest during the backup period are retried.



NOTE

Toggling between `read-only` or `flash-backup` and other forest update types triggers a forest restart. This activity is visible in the log file.

When the `read-only` or `flash-backup` update type is set, the forest will have the following characteristics:

- If a database has at least one updateable forest, and an insert, update or delete without a place key is requested, it will choose one of the updateable forests to perform the operation.
- No merges are allowed on the forest. Attempts to explicitly merge such forests do nothing.
- No re-indexing/re-fragmenting is allowed on the forest.
- You cannot upgrade from the forest. An attempt to upgrade will return an error.
- If a forest is set to `read-only` or `flash-backup`, an insert, update, or delete transaction will either generate an exception (in the case of `read-only`) or retried later (in the case of `flash-backup`).
- You cannot clear, restore, or fully delete the forest. However, you can delete the forest configuration, as described in [Section 22.12, “Deleting a Forest from a Host” \[210\]](#).

- Backups are permitted on the forests. However, they will not modify the last backup time in the forest label. Consequently, the last backup time in the forest will denote the last time the forest was backed up when it wasn't read-only or flash-backup.
- If the database index settings are changed and index detection is set to 'automatic', then the forests will work, but the indexes won't be picked up. If index detection is set to 'none', you will get wrong results.
- You can enable failover on a read-only and flash-backup forest.

22.5. Attaching and Detaching Forests Using the Forest Summary Page

The Forest Summary page lists all of the forests in the cluster, along with various information about each forest such as its status, which host is the primary host, and amount of free space for each forest. It also lists which database each forest is attached to, and allows you to attach and/or detach forests from databases. Alternately, you can use the Database Forest Configuration page to attach and detach a forest, as described in [Section 12.3, "Attaching and/or Detaching Forests to/from a Database"](#) [89].

Follow these steps to attach or detach one or more forests to or from a database:

1. Click the **Forests** icon on the left tree menu. The **Summary** tab appears:
2. For each forest whose database assignment you want to change, select the name of the new database from the **Database** list.

Forest	Status	Database	Primary Host
App-Services	open	App-Services	hpzbook-6213.marklogic.com
Documents	open	App-Services	hpzbook-6213.marklogic.com
Extensions	open	Documents	hpzbook-6213.marklogic.com
Fab	open	Modules	hpzbook-6213.marklogic.com
Last-Login	open	Fab	hpzbook-6213.marklogic.com
Meters	open	Extensions	hpzbook-6213.marklogic.com
Modules	open	Security	hpzbook-6213.marklogic.com
paligoProcessor-content-1	open	Schemas	hpzbook-6213.marklogic.com
paligoProcessor-content-2	open	Triggers	hpzbook-6213.marklogic.com
paligoProcessor-content-3	open	Last-Login	hpzbook-6213.marklogic.com
Schemas	open	Meters	hpzbook-6213.marklogic.com
sec-101	open	sec-101	hpzbook-6213.marklogic.com
Security	open	paligoProcessor-content	hpzbook-6213.marklogic.com
Triggers	open	paligoProcessor-modules	hpzbook-6213.marklogic.com
		paligoProcessor-content	hpzbook-6213.marklogic.com
		paligoProcessor-content	hpzbook-6213.marklogic.com

3. After you have made your selections, click **OK** to save the forest assignment changes.



NOTE

If you change a database assignment from one database to another, it will detach the forest from the previous setting and attach it to the new setting. Be sure that is what you intend to do. Also, if you detach from one database and attach to another database with different index settings, the forest will begin reindexing if `reindexer.enable` is set to `true`.

The forests you attached or detached are now reflected in the database configuration. Attaching and detaching a forest to a database are “hot” admin tasks.

22.6. Making Backups of a Forest

MarkLogic Server backs up forest data by transactionally creating an image copy of a specified forest. You can back up data at the granularity of a forest or of a database. Use the Admin Interface to back up a forest.

Forest-level backups only back up the data in a forest, and are not guaranteed to have a consistent database state to restore. The data in the forest is consistent, but other parts of the database (other forests, the schema database, and so on) might be different when you restore the data. For a guaranteed consistent backup, perform a complete database backup. For information on backing up a database, see [Section 19, “Backing Up and Restoring a Database” \[160\]](#).



NOTE

Forest backups do not provide a journal archive feature, as described for database backups in [Section 19, “Backing Up and Restoring a Database” \[160\]](#). However, you can manually invoke the `xmmp:start-journal-archiving` function during a forest backup to make use of journal archiving with your forest backups.

This section describes the forest backup procedures.

22.6.1. Backing Up a Forest

To initiate a forest backup, follow these steps:

1. Click **Forests** in the left tree menu. A list of forests appears.
2. Click your target forest.
3. Click the **Backup/Restore** tab at the top right.
4. Enter the name of the directory in which you want the backup copy of the forest. You must provide an absolute path. Each directory must be unique for each forest.



WARNING

The software deletes *all* the files in this directory before writing the new backup. To retain multiple generations of backup, specify a different backup directory for each backup.

5. Select **Backup**.
6. Click **OK**.
7. A confirmation message appears. Click **OK** again to confirm the backup.

Your data in the selected forest is now backed up to the specified directory. Backing up your data is a “hot” admin task; the changes take effect immediately.



WARNING

When performing backups on the Windows platform, ensure that no users have the Forests or Data directories (or any subdirectories within them) open while the backup is being made.

22.6.2. Scheduling a Forest Backup

You can schedule forest backups to periodically back up a forest. You can schedule backups to occur daily, weekly, monthly, or you can schedule a one-time backup. You can create as many scheduled backups as you want.

To create a scheduled backup, follow these steps:

1. Click **Forests** in the left tree menu. A list of forests appears.
2. Click your target forest.
3. Click the **Scheduled Backup** link in the tree menu for the forest.
4. Delete any existing scheduled backups if you no longer need them.
5. Click the **Create** tab.
6. In the **Backup Directory** field, Enter the absolute path to the backup directory. The backup directory must have permissions such that the MarkLogic Server process can read and write to it.
7. Select a backup type. Complete the additional fields for the backup type selected. See [the table](#) for information about the additional fields.

Backup Type	Additional Information
Minutely	Enter how often the backup should run in the Backup Period field.
Hourly	In the Backup Period field, enter the number of hours between each backup. In the Backup Minute field, select how many minutes after the hour the backup is to start.
Daily	For daily, enter how many days between each backup and the time of day.
Weekly	For weekly, enter how many weeks between each backup, check one or more days of the week, and the time of day for the backup to start.
Monthly	For monthly, enter how many months between each backup, select one day of the month (1-31), and the time of day for the backup to start.
Once	For one-time, enter the backup start date in MM/DD/YYYY notation (for example, 07/29/2009 for July 29, 2009) and time in 24:00 notation.

8. Enter the time of day to start the backup.
9. Click **OK** to create the scheduled backup.

The backups will automatically start according to the specified schedule.

22.7. Restoring a Forest

You can restore a forest from a backup.

To restore a forest from a backup, follow these steps:



WARNING

Before following this procedure on the Windows platform, ensure that users do not have forests, directories, or any subdirectories, open.

1. Click **Forests** in the left tree menu. A list of forests appears.
2. Click your target forest.
3. Click the **Backup/Restore** tab.
4. Enter the name of the directory that contains the backup copy of the forest.
5. Click **Restore**.
6. Click **OK**.
A confirmation message displays.

7. Confirm that you want to restore data from this backup directory and click **OK**.

Restoring data from your backup is a “hot” admin task; the changes take effect immediately.

22.8. Rolling Back a Forest to a Point In Time

You can use the `xdmp:forest-rollback` function to roll the state of one or more forests back to a specified system timestamp. To roll forest(s) back to an earlier timestamp, you must first set the merge timestamp to keep deleted fragments from that specified timestamp. For details on rolling back a forest, including the procedure to perform a rollback, see [Rolling Back a Forest to a Particular Timestamp](#) in the *Application Developer's Guide* and the `xdmp:forest-rollback` API documentation in the in the MarkLogic documents for Server-Side XQuery APIs.

22.9. Merging a Forest

You can merge forest data as described in [Section 15, “Understanding and Controlling Database Merges” \[108\]](#), merging a forest improves performance and is periodically done automatically in the background by MarkLogic Server. The **Merge** button allows you to explicitly merge the data for this forest.

To explicitly merge the forest, follow these steps:

1. Click **Forests** in the left tree menu. A list of forests appears.
2. Click your target forest.
3. Click the **Merge** button.
A confirmation message appears.
4. Confirm that you want to merge the forest data and click **OK**.

Merging data in a forest is a “hot” admin task; the changes take effect immediately.

22.10. Clearing a Forest

You can clear the document data from a forest using the Admin Interface. Clearing a forest removes all fragments from the forest, but does not remove its configuration information.

To clear all data from a forest, follow these steps:

1. Click **Forests** in the left tree menu. A list of forests appears.
2. Click your target forest.
3. Click the **Clear** button on the **Configure** tab.
A confirmation message displays.
4. Confirm that you want to clear the document data from this forest and click **OK**.

Clearing data in a forest is a “hot” admin task; the changes take effect immediately.

22.11. Disabling a Forest

You can disable a forest using the Admin Interface. Disabling a forest unmounts the forest from the database and clears all memory caches for all the forests in the database. The database remains unavailable for any query operations while any of its forests are disabled.

Disabling a forest does not delete the configuration or document data. The forest can later be re-enabled by clicking **Enable**.

To disable a forest, follow these steps:

1. Click **Forests** in the left tree menu. A list of forests appears.
2. Click your target forest.
3. Click the **Disable** button.

A confirmation message appears.

4. Confirm that you want to disable the forest by clicking **Disable**.

22.12. Deleting a Forest from a Host



NOTE

A forest cannot be deleted if it is still attached to a database. In addition, a Read-Only or Flash-Backup forest cannot be fully deleted. However, the configuration information can be.

To delete a forest, follow these steps:

1. Click **Forests** in the left tree menu. A list of forests appears.
2. Click your target forest.
3. Click the **Delete** button .
A confirmation message appears.
4. Select either **Configuration Only** to delete only the configuration information, or **Full Delete** to delete the configuration information and the document data.
5. Click **OK**.

Deleting a forest is a “hot” task; the changes take effect immediately.

22.13. Rolling Back a Prepared XA Transaction Branch

MarkLogic Server transactions may participate in global, distributed XA transactions. The XA Transaction Manager usually manages the life cycle of transactions participating in an XA transaction, independent of MarkLogic Server. However, it may be necessary to manually rollback the MarkLogic Server portion of a global transaction (called a *branch*) if the Transaction Manager is unreachable for a long time. For details, see [Heuristically Completing a Stalled Transaction](#) in *Developing with XCC*.



NOTE

Heuristic completion bypasses the Transaction Manager and the Two Phase Commit process, so it can lead to data integrity problems. Use heuristic completion only as a last resort.

Before the MarkLogic Server branch of an XA transaction is prepared, the transaction may be rolled back from the host status page of the host evaluating the transaction. See [Section 21.9, “Rolling Back a Transaction” \[200\]](#).

Once the MarkLogic Server branch of an XA transaction enters the prepared state, the transaction appears only on the forest status page of the coordinating forest. To find the coordinating forest, examine the Forest Status page for each forest belonging to the participating database. The transaction will only appear on the status page for the coordinating forest.

To heuristically rollback the MarkLogic Server portion of an XA transaction, follow these steps:

1. Click **Forests** on the left tree menu. The forest summary page appears.

2. Click the name of the coordinating forest.
3. Click the **Status** tab.
4. Locate the target transaction in the transaction list. If you do not see a transaction list on the status page, then this forest is not the coordinating forest for any prepared transactions.
5. Click [rollback] on the right side of the target transaction status to initiate the rollback. The rollback confirmation dialog appears. For example:
6. Click **OK** to confirm the rollback. The rollback completion page appears.
7. Click **OK** to return to the **Forest Status** page.

The rolled back transaction enters the “remember abort” state, indicating MarkLogic Server should remember that the local transaction was aborted until the Transaction Manager re-synchronizes the global transaction. Once re-synchronization occurs, the transaction no longer appears in the forest status. For details, see [Heuristically Completing a MarkLogic Server Transaction](#) in *Developing with XCC*.

You may use the Forest Status page to force MarkLogic Server to forget the rollback without waiting for the Transaction Manager. This is not recommended as it leads to errors and, potentially, a loss of data integrity when the Transaction Manager attempts to re-synchronize the global transaction. If forgetting the rollback is necessary, use the [forget] link in the transaction list on the Forest Status:

23. Security Administration

MarkLogic Server uses a role-based security model. A user's privileges and permissions are based on the roles assigned to the user. This section describes how to use the Admin Interface to manage security objects. For details on how to manage security objects programmatically, see [Creating and Configuring Roles and Users](#) and [User Maintenance Operations](#) in the *Scripting Administrative Tasks Guide*.

For background information on understanding the security model in MarkLogic Server, see [Securing MarkLogic Server](#).

23.1. Security Entities

The key entities in MarkLogic Server's security model are:

- User
A *user* within the model has a set of roles. A user has privileges and permissions within the system based on the roles he is given.
- Role
A *role* gives privileges and permissions to a user. A role may inherit from multiple roles. Role inheritance is an "is-a" relationship. Hence, an inherited role also has the privileges and permissions of its parent(s).
- Execute Privilege
An *execute privilege* grants authorization to perform a protected action. Only roles (and their inherited roles) specified in the execute privilege can perform the action.
- URI Privilege
A *URI privilege* grants authorization to create a document within a protected base URI. Only roles (and their inherited roles) specified in the URI privilege can create the document within the protected base URI.
- Permission
A *permission* protects a document or a collection. Each permission associates a single role with a capability (Read, Update, Insert). A protected document or collection has a set of associated permissions.
- Collection
A *collection* groups a set of documents that are related. A document may belong to any number of collections. A collection exists in the system when a document in the system states that it is part of that collection. However, an associated collection object is not created and stored in the *Security* database unless it is protected.
Permissions created at the collection level apply to the collection but not to documents within the collection. A user needs to have permissions at the both the collection and document level to be able to add documents to a protected collection.
- Amp
An *amp* gives the User additional roles temporarily while the user is performing a certain task (executing a function).
- Certificate Authority
A *certificate authority* (CA) is a trusted third party that certifies the identity of entities, such as users, databases, administrators, clients, and servers. A CA is used by the SSL (Secure Sockets Layer) security standard to provide encrypted protection between browsers and App Servers. When an entity requests certification, the CA verifies its identity and grants a certificate, which is signed with the CA's private key. If the CA is trusted, then any certificate it issues is trusted unless it has been revoked. For details on SSL support in the MarkLogic Server, see [Configuring SSL on App Servers](#) in *Securing MarkLogic Server*.
- Certificate Template

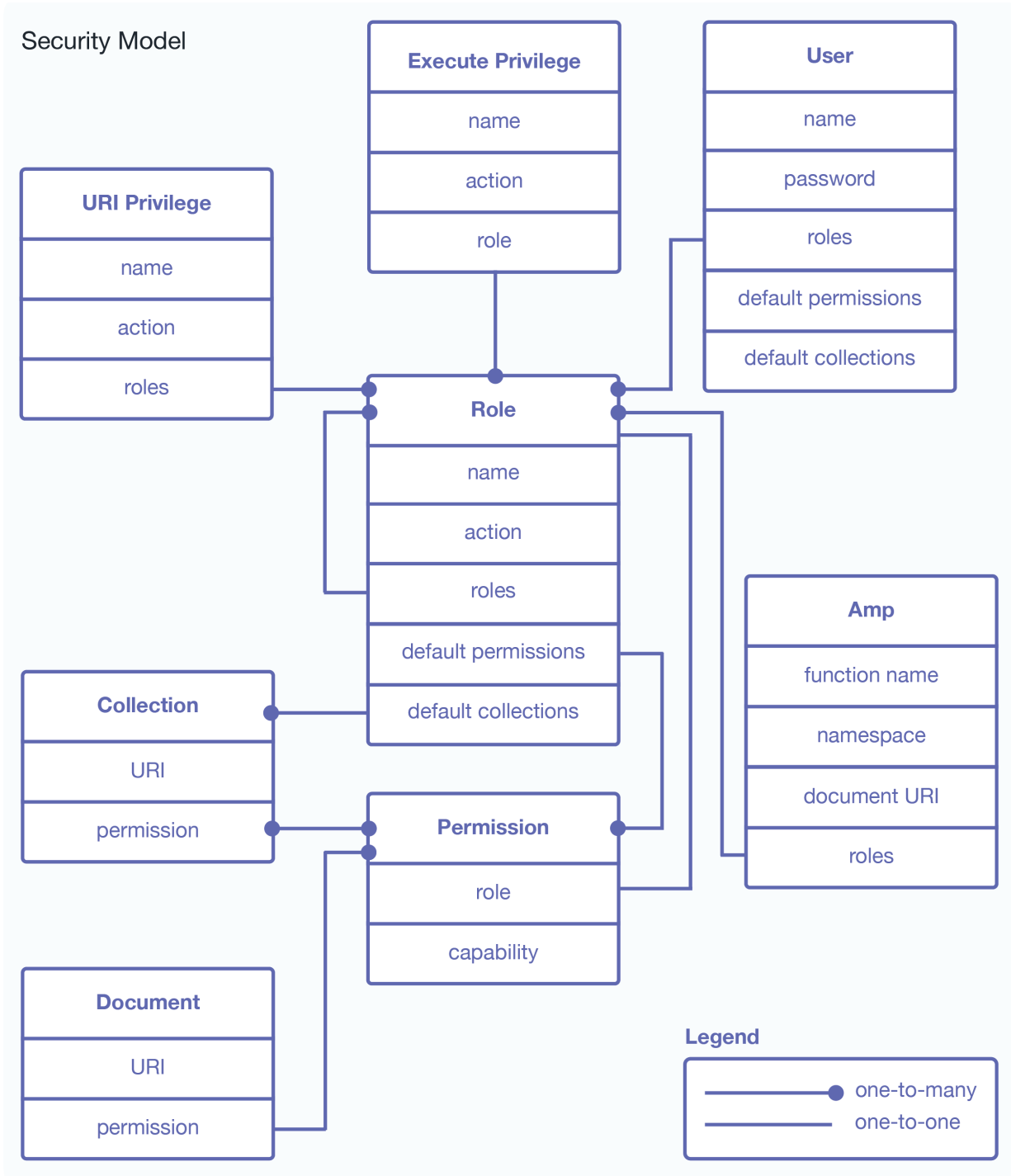
A *certificate template* is a MarkLogic construct that is used to generate certificate requests for the various hosts in a cluster. A certificate template is used by the SSL (Secure Sockets Layer) security standard to provide encrypted protection between browsers and App Servers. The template defines the name of the certificate, a description, and identity information about the owner of the certificate. For details on SSL support in the MarkLogic Server, see [Configuring SSL on App Servers](#) in *Securing MarkLogic Server*.

- External Authentication

An *External Authentication Configuration Object* is used to configure MarkLogic Server for external authentication by LDAP, Kerberos, SAML, or OAuth. An external authentication configuration object specifies which authentication protocol and authorization scheme to use, along with any other parameters necessary. For details on external authentication with MarkLogic Server, see [External Security](#) in *Securing MarkLogic Server*.

- Security Entity Relationships

The following diagram illustrates the relationships between the different entities in the MarkLogic Server security model.



The remaining sections detail the procedures to administer MarkLogic Server security entities. All security administrative tasks are “hot”— the changes take effect immediately without a server restart.

Permissions are not administered through the administrative interface and are not described in detail in this document. For more information on using permissions in MarkLogic Server, see the [XQuery and XSLT Reference Guide](#).

23.2. Users

A User has a set of roles. A user has privileges and permissions within the system based on the roles he is given. A user can perform tasks (execute functions) based on his privileges and access data based on his permissions.

Each user has an associated user name and password. A user also has default collections. When a user creates a document but does not explicitly associate the document with a set of collections, the document is automatically added to the user's default collections. Default permissions can be created for a user. When a user creates a document but does not explicitly set the permissions for the document, the document will be given the user's default permissions.

If security is turned on for an HTTP, ODBC, or XDBC server, all users in the security database will have access to the server. Finer granularity security control to functions in XQuery programs running on the HTTP, ODBC, or XDBC servers are accomplished through the use of `xdmp:security-assert()` within the code. Granular secured access to documents is achieved through the use of permissions associated with each protected document.

23.2.1. Creating a User

To create a user, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Users**.
3. Click the target user.
4. Click the **Create** tab.
5. Enter a name for the user in the **User Name** field.
6. [OPTIONAL] Enter a description for the user.
7. Enter a password.
8. Re-enter the password in the **Confirm Password** field.
9. If the user is to be authorized externally by LDAP or Kerberos, enter one or more Distinguished Names (LDAP) or User Principals (Kerberos) in the **External Names** section. For details on external authorization, see [External Security](#) in *Securing MarkLogic Server*.
10. Under the **Roles** section, check the roles to assign the user.
11. [OPTIONAL] Create default permissions for this user: Select a role and pair the role with the appropriate capability (read, insert, update). If there are more than 3 default permissions you want to add for this user, you can do so on the next screen after you click **OK**.
12. [OPTIONAL] Create default collections for this user: Type in the collection URI for each collection you want to add to the user's default collection. If there are more than 3 default collections you want to add for this user, you can do so on the next screen after you click **OK**.
13. Click **OK**.

The user is now added to the system and the user configuration page appears. If you want to add more default permissions or collections to the user, scroll down to the section for default permissions or collections.

23.2.2. Viewing a User Configuration

To view a user's configuration, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Users**.
3. Click the target user.
4. The user's configuration information is displayed.

23.2.3. Modifying a User Configuration

To modify the configuration for a user, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Users**.
3. Click the target user.

4. Perform any modifications needed to the user's configuration. Modifications might include changing any of the user credentials (including password), adding or removing role assignments, adding or removing default permission settings, or adding or removing default collection settings.



WARNING

Making changes to the to the user configuration affects the access control policy for that user, which can either increase or decrease the activities authorized for the user. For more details on how the security system works, see [Securing MarkLogic Server](#).

5. Click **OK** to save the changes.

The new changes are in effect for all transactions beginning after the user changes are committed.

23.2.4. Deleting a User

To delete a user from the security database, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Users**.
3. Click the target user.
4. Click on the **Delete** button.
5. Click **OK** to confirm deleting the user.

The user is permanently deleted from the security database.

23.3. Roles

MarkLogic Server implements a role-based security model. Therefore, the Role is a central security concept in MarkLogic Server. A role gives a user privileges (both Execute and URI) to perform certain actions in a system. An Execute Privilege allows a user to perform a protected action. A URI Privilege allows a user to create a document under a protected URI. A role also gives a user the permissions to access protected documents.

A role may inherit from multiple roles. The inheritance relationship for roles is an "is-a" relationship. Therefore, a role gets the privileges and permissions of the roles from which they inherit.

MarkLogic Server is installed with the following pre-defined roles:

Role	Description
admin	This role has the privileges and permissions needed to perform administrative tasks. This role has the highest level of access in the system.
admin-builtins	This role has the privileges needed to call the admin-builtins functions.
filesystem-access	This role has the privileges to access the filesystem.
merge	This role has the privileges needed to force a merge in the system.
security	This role has the privileges to perform all the security-related administrative functions.

While you are able to change the configuration settings of these pre-defined roles (except for the `admin` role) or delete any of them, we strongly recommend that you proceed with caution.

A role has default collections. When a user of a role creates a document but does not explicitly associate the document with a set of collections, the document is automatically added to a set of default collections. This set of default collections is the union of the default collections defined for the user, the roles the user has, and the roles from which the user's directly assigned roles inherit.

A role has default permissions. When a user of a role creates a document but does not explicitly set the permissions for the document, the document will be given a set of default permissions. This set of

default permissions is the union of the default permissions defined for the user, the roles the user has, and the roles from which the user's directly assigned roles inherit.

For more details about the role-based security model in MarkLogic Server, see [Securing MarkLogic Server](#).

23.3.1. Creating a Role

To create a role, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Roles**.
3. Click the **Create** tab.
4. Type in a name for the role in the **Role Name** field.
5. Type in a description for the role (optional).
6. If you want to place the role into the named compartment, enter name of the compartment in the **Compartment** field. Compartments provide an additional level of organization and control by grouping together related roles. They act as a higher-level container for roles and can be used to define access privileges for a specific set of resources. For example, you may have a compartment called "Finance" that contains roles such as "Finance Manager," "Accountant," and "Auditor." If a document has any permissions (role/capability pairs) with roles that have a compartment, then the user must have those roles with each of the compartments (regardless of which permission they are in) to perform any of the capabilities.
7. If the role is to be mapped to an LDAP group or an OAuth group, enter one or more group names in the **External Names** section. For details on external authorization, see [External Security in Securing MarkLogic Server](#).
8. Under the **Roles** section, select the roles from which this role will inherit.
9. Under the **Execute Privileges** section, select from the available execute privileges to associate with the role.
10. Under the **URI Privileges** section, select the available URI privileges to associate with the role.
11. Create default permissions for this role (optional). Select a role and pair the role with the appropriate capability (read, insert, update). If there are more than 3 default permissions you want to add for this role, you can do so on the next screen after you click **OK**.
12. Create default collections for this role (optional). Type in the collection URI for each collection you want to add to the role's default collections. If there are more than 3 default permissions you want to add for this user, you can do so on the next screen after you click **OK**.
13. Click **OK**.

The role is now added to the system and the **Role Configuration** page appears. If you want to add more default permissions or collections to the role, scroll down to the section for default permissions or collections.

23.3.2. Viewing a Role

To create a role, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Roles**.
3. Click the target role.
4. View the configuration for the role.

23.3.3. Modifying a Role Configuration

To modify a role configuration, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Roles**.
3. Click the target role.

4. Perform any modifications needed to the role configuration. Modifications might include adding or removing role assignments, adding or removing default permission settings, or adding or removing default collection settings.



WARNING

Making changes to the to the role configuration affects the access control policy for that role, which can either increase or decrease the activities authorized for any users who have that role (either directly or indirectly). For more details on how the security system works, see [Securing MarkLogic Server](#).

5. Click **OK** to save the changes.

The new changes are in effect for all transactions beginning after the user changes are committed.

23.3.4. Deleting a Role

You can delete a role from the security database. The system does not check to see if there are any users with that role before deleting it. A deleted role is automatically removed from all users still assigned to that role. Users who were assigned to the deleted role lose the permissions and privileges given by that role.

To delete a role, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Roles**.
3. Click the target role.
4. Click the **Delete** button.
5. Click **OK** to confirm deleting the role.

The role is now deleted from the security database.

23.4. Execute Privileges

The actions that a user can perform in the server are determined by the set of privileges that they possess. Privileges are connected to users with roles. An Execute Privilege grants authorization to perform a protected action. An execute privilege specifies a protected action, and the roles that can perform the action. Roles that inherit from the specified roles can also perform the protected action. The protected action is represented as a URI.

Once an execute privilege is created, it is enforced in XQuery programs through the use of `xdmp:security-assert(<protected-action-uri>, "execute")` in the code. That is, `xdmp:security-assert(<protected-action-uri>, "execute")` can be added at the entrance to function or a section of code that has been protected. If the system is executing as a user without the appropriate roles as specified by the execute privilege, an exception is thrown. Otherwise, system satisfies the security-assert condition and proceeds to execute the protected code.

23.4.1. Creating an Execute Privilege

To create an execute privilege, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Executive Privileges**.
3. Click the **Create** tab.
4. In the **Privilege Name** field, enter the name of the execute privilege. Use a name that is descriptive of the action this execute privilege will protect. For example, `create-user` is the name of an existing execute privilege that gives a role the authorization to create a user.

5. In the **Action** field, Enter a protected action, represented as a URI. You can use any URI but we recommend you follow the conventions for your company. For example, the URI for the `create-user` execute privilege is `http://marklogic.com/xdmp/privileges/create-user`.
6. Under the **Roles** section, select the roles that are allowed to perform the protected action.
7. Click **OK**.

The execute privilege is now added to the security database. You can now use the `xdmp:security-assert()` function in your code to associate this privilege with a protected operation.

23.4.2. Execute Privilege: grant-my-privilege

A user with the `grant-my-privileges` privilege can assign privileges that they already possess to roles that they are allowed to modify. This feature works in conjunction with the “data roles” feature. The `grant-my-privileges` privilege is useless in isolation, as its only purpose is to assign privileges to roles.

To access the `grant-my-privilege` feature:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Executive Privileges**.
3. Look in the **Privilege** column and scroll down until the **grant-my-privileges** link appears.
4. Click **grant-my-privileges**. The **Execute Privilege** screen opens.
5. Select the roles to assign to the privilege.
6. Click **OK** when you are done to save the changes.

The precise set of privileges that a user can assign is determined by the privileges that they already possess. It is not possible for a user to assign a privilege that they do not possess (admin, for example). If a user attempts to change the privileges associated with a role, the request will succeed if (*and only if*) the following conditions apply:

- The user has the “grant-my-privileges” privilege. A user without this privilege cannot make any changes to the privileges associated with a role.
- The user has the “create-data-roles” privilege and the necessary granular edit privilege for the role that they are modifying. Without these privileges, they cannot modify the role.
- The user possesses all of the privileges that they are attempting to add or remove from the role.

23.4.3. Viewing an Execute Privilege

To view an execute privilege, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Executive Privileges**.
3. Click on the name of the execute privilege that you want to view.
4. View the configuration for the execute privilege.

23.4.4. Modifying an Execute Privilege

To modify an execute privilege, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Executive Privileges**.
3. For the privilege which you want to modify, view the configuration as described in [Section 23.4.3, “Viewing an Execute Privilege” \[219\]](#).
4. Perform any modifications needed to the privilege (for example, add or remove role assignments).

**WARNING**

Making changes to the to the execute privilege configuration affects the access control policy for that privilege, which can either increase or decrease the activities authorized for any users who have any of assigned roles (either directly or indirectly). For more details on how the security system works, see [Securing MarkLogic Server](#).

5. Click **OK** to save the changes.

The new changes are in effect for all transactions beginning after the user changes are committed.

23.4.5. Deleting an Execute Privilege

You can delete an execute privilege from the security database. However, an exception will be thrown when a `security-assert()` on the protected action specified in the deleted execute privilege is encountered. That is, a deleted execute privilege behaves like an execute privilege for which no role has been given access to the protected action.

To delete an execute privilege, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Executive Privileges**.
3. Click the name of the execute privilege that you want to delete.
4. On the **Execute Privileges** page click **Delete**.
5. Click **OK** to confirm deleting the execute privilege.

The execute privilege is now deleted from the security database.

23.5. URI Privileges

A URI Privilege grants authorization to create documents under a protected URI. That is, a URI privilege specifies the roles that are allowed to create documents with the protected URI as the base URI (prefix) in the document URI. Roles that inherit from the specified roles can also create the documents under the protected URI.

Unlike an execute privilege, where `xdmp:security-assert()` needs to be called explicitly to protect a function, a URI privilege is automatically enforced. When `xdmp:document-insert()` is called, the system checks the base URIs (prefix) of the document URI specified to see if they might be protected by a URI privilege. If the base URI has an associated URI privilege, it checks the roles of the user to see if any of the user's roles gives the user authorization to create the document within the protected base URI. If the user has the requisite authorization, the document is inserted into the database. Otherwise, an exception is thrown.

Use the procedures in this section to create, manage and maintain URI privileges.

23.5.1. Creating a URI Privilege

To create a URI privilege, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **URI Privileges**.
3. Click on the **Create** tab.
4. In the **Privilege Name** field, enter the name of the URI privilege. Use a name that is descriptive of the base URI to be protected. For example, to restrict the creation of documents under a base URI reserved for the accounting group, you might use the name "accounting_files".
5. In the **Uri** field, enter the base URI to protect. While the base URI does not have to map to an actual directory, it should follow the directory structure convention (for example, `/myfiles/`

accounting_files). In this example, only the user with this URI privilege can create a file with the URI /myfiles/accounting_files/account1.xml.

6. In the **Roles** section, select the roles that are allowed to create documents under the base URI.
7. Click **OK**.

The URI privilege is created and added to the security database.

23.5.2. Viewing a URI Privilege

To view a URI privilege, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **URI Privileges**.
3. Click the target privilege.
4. View the URI privilege.

23.5.3. Modifying a URI Privilege

To modify an execute privilege, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **URI Privileges**.
3. Click the target privilege.
4. Perform any modifications needed to the privilege (for example, add or remove role assignments).



WARNING

Making changes to the to the URI privilege configuration affects the access control policy for that privilege, which can either increase or decrease the activities authorized for any users who have any of assigned roles (either directly or indirectly). For more details on how the security system works, see [Securing MarkLogic Server](#).

5. Click **OK** to save the changes.

The new changes are in effect for all transactions beginning after the user changes are committed.

23.5.4. Deleting a URI Privilege

You can delete a URI privilege from the security database.

To delete a URI privilege, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **URI Privileges**.
3. Click the target privilege.
4. On the **URI Privilege** page for the given privilege, click the **Delete** button.
5. Click **OK** to confirm deleting the URI privilege.

The URI privilege is now deleted from the security database.

23.6. Amps

An Amp gives the user additional roles temporarily while the user is performing a certain task (executing a function). While the user is executing the “amp-ed” function, the user receives additional privileges and permissions given by the additional roles. An amp is useful when a user needs additional privileges and permissions only while the user is executing a certain function.

Giving the user additional roles permanently could compromise the security of the system. On the other hand, an amp enables granular security control by limiting the effect of the additional roles (privileges

and permissions) to a specific function. For example, a user may need a count of all the documents in the database when the user is creating a report. However, the user does not have read permissions on all the documents in the database, and hence does not know the existence of all the documents in the database. An amp can be created for `document-count()` to elevate the user to an `admin` role temporarily while the user is executing the function to count the documents in the system.

An amp is defined by the local name of the function, the namespace and the document URI. The document URI must begin with a forward slash “/” and is treated as being rooted relative to the *Modules* directory in the installation path. When resolving an amp, MarkLogic Server looks for the file using a path rooted relative to the *Modules* directory in the installation path. If it finds a function that matches the local name and namespace using the specified path, it applies the amp to the function.



NOTE

Database names can be used in the trigger and amp creation APIs, thus making it easy to support the same functionality on replica clusters for databases with the same names.

For more details about amps, see [Securing MarkLogic Server](#). For examples of amps, look at one of the amps created during installation. To view an amp, follow the instructions in the section [Section 23.6.2, “Viewing an Amp” \[222\]](#).

Use the procedures in this section to create, manage, and maintain amps.

23.6.1. Creating an Amp

To create an amp, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Amps**.
3. Click on the target amp.
4. Click the **Create** tab.
5. Enter the local name of the function (without parentheses) in which the amp takes effect. For example: `my-function`.
6. Enter the namespace in which the function is defined.
7. Enter the document URI for the document in which the function is defined. This document URI must begin with a forward slash (for example, `/amped-functions.xqy`). The specified document must be placed in the *Modules* directory within the installation path. For example, if `/mydir/my-amps.xqy` is specified in the document uri, `my-amps.xqy` must be placed in *installation-directory/Modules/mydir*.
8. In the **Database** field, select the database where the function is stored. If the function is stored in the *Modules* directory on the filesystem, set the database to **(filesystem)** (which is the default value).
9. Under the **Roles** section, select the additional roles that will be given to the user while the user is executing the function.
10. Click **OK**.

The amp is now added to the security database.

23.6.2. Viewing an Amp

To view an amp, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.

2. Click **Amps**.
3. Click on the target amp.
4. Click on the name of the amp you want to view.
5. View the amp.

23.6.3. Modifying an Amp

To modify an amp, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Amps**.
3. Click on the target amp.
4. Perform any modifications needed to the amp (for example, add or remove role assignments).



WARNING

Making changes to the to the amp configuration affects the access control policy for that amp, which can either increase or decrease the activities authorized for any users who have any of assigned roles (either directly or indirectly). For more details on how the security system works, see [Securing MarkLogic Server](#).

5. Click **OK** to save the changes.

The new changes are in effect for all transactions beginning after the user changes are committed.

23.6.4. Deleting an Amp

You can delete an amp from the security database.

To delete an amp, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Amps**.
3. Click on the target amp.
4. On the **Amp** page, click **Delete**.
5. Click **OK** to confirm deleting the amp.

The amp is now deleted from the security database.

23.7. Protected Collections

A *collection* groups a set of documents that are related and enables queries to target subsets of documents within a database efficiently. A document may belong to any number of collections simultaneously. A collection exists in the system when a document in the system states that it is part of that collection.

A *protected collection* is one for which only authorized users can associate documents with the collection. When you create a protected collection, an associated protection collection object is created and stored in the security database.

You must understand the following key concepts and limitations of protected collections:

- A protected collection dictates who can *add* documents to the collection. It provides no other access control.
- A protected collection does not control access to the documents in the collection. Use document permissions for this purpose.
- Only users with a role that has update permissions for the collection can add documents to the collection or use explicit collection operations such as `xdmp.documentRemoveCollections` to remove a document from a protected collection.

- A user with update permissions on a document can remove the document from a protected collection by reinserting the document with a different set of collections.

Use these procedures in this section to create, manage, and maintain collections.

23.7.1. Creating a Protected Collection

To create a protected collection, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Collections**.
3. Click the **Create** tab.
4. Enter the URI for the collection.
5. In the **Permissions** section, add permissions (role-capability pair) to the collection. Select from the available roles and pick a capability for the role. You should usually select the update capability as this is the only one that affects how users interact with the collection. Only users with a role with the update capability can add documents to the collection; for details, see [Section 23.7, “Protected Collections” \[223\]](#).
6. Click **OK**.

The protected collection is added to the database.

23.7.2. Viewing a Protected Collection

To view a protected collection, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Collections**.
3. Click the target collection.
4. View the collection.

23.7.3. Removing a Permission from a Protected Collection

To remove a permission from a protected collection, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Collections**.
3. Click the target collection.
4. In the **Permissions** section, uncheck the box next to the permission you want to remove.
5. Click **OK**. The permission is removed from the collection.

23.7.4. Deleting a Protected Collection

To delete a protected collection, follow these steps:

1. Click **Security** in the left tree menu. A list of security items appears.
2. Click **Collections**.
3. Click the target collection.
4. Click **Delete** near the top right.
5. Click **OK** to confirm deleting the collection.

The protected collection is deleted from the security database.

23.8. Certificate Templates

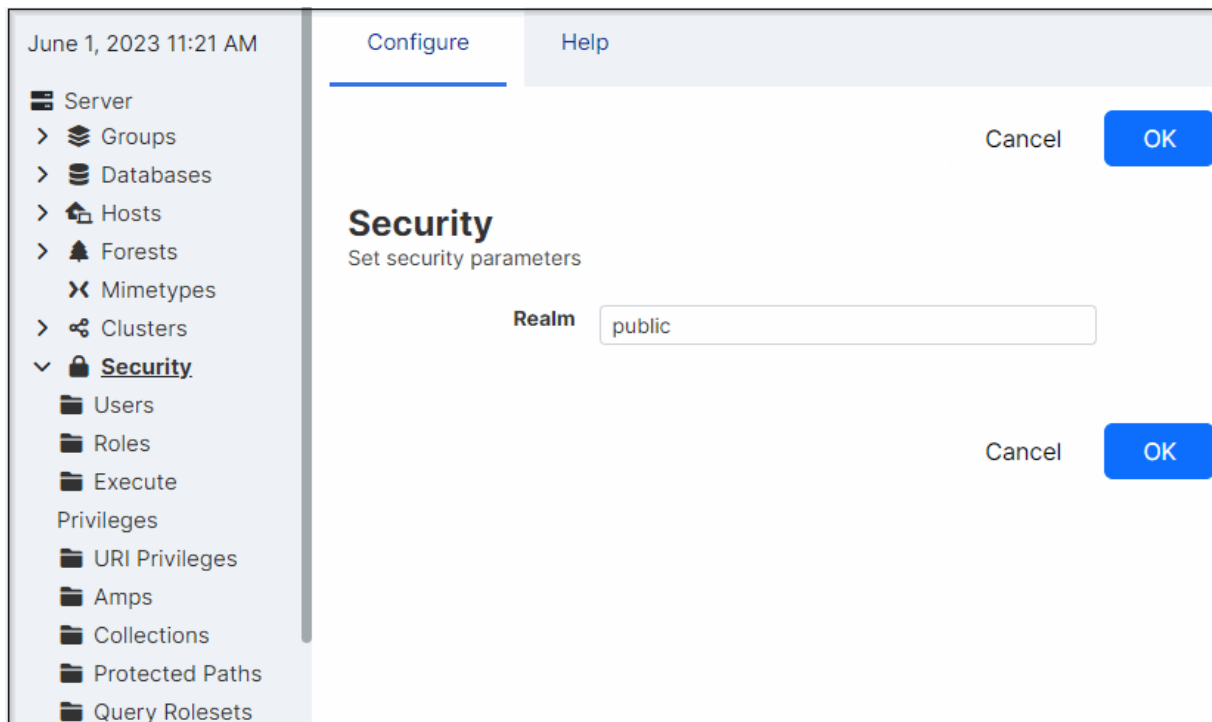
A Certificate Template contains the identification information associated with an SSL certificate. See [Configuring SSL on App Servers](#) in *Securing MarkLogic Server* for details.

23.9. Realm

MarkLogic Server stores the realms for application servers in the security database. Each application server takes its realm from the security database to which it is connected. Realms are used in computing digest passwords.

23.9.1. Setting the Realm

The realm is stored in the security database to which the Admin Interface is connected, and is set at installation time:



23.9.2. Changing the Realm

Changing the realm in the security database invalidates all user digest passwords. This only affects application servers whose `authentication` setting is `digest` or `digestbasic` mode.

In `digest` mode, you need to re-enter all user passwords in the security database. Changing the passwords in the security database will cause the server to recalculate the digest passwords. In `digestbasic` mode, the first time a user logs into the server after the realm is changed, the user will be prompted to enter their passwords multiple times before they are logged into the system. However, the server will automatically recalculate their digest password with the new realm at that time, and they will have a normal login process for future access.



WARNING

If you change the realm, any App Servers that uses digest authentication will no longer accept the existing passwords. This includes the Admin Interface, and includes passwords for users with the `admin` role. Therefore, changing the realm will make it so you can no longer log into the Admin Interface.

If you are sure you want to change the realm after installation despite the warning, follow these steps:

1. Click **Security** in the left tree menu.
2. Click the **Configure** tab. :
3. Change the realm to the desired value.
4. Click **OK**.
5. Click **OK** again on the confirmation page. Note that this will invalidate all digest passwords, including the password for the current user running the Admin Interface if the Admin Interface App Server is set to digest authentication (which is the default setting).

24. Text Indexing

Before loading documents into a database, you have the option of specifying a number of parameters that will impact how the text components of those documents will be treated. This section describes those parameters



NOTE

Text indexes and phrasing parameters are set on a per-database basis.

24.1. Text Indexes

MarkLogic Server allows you to configure, at the database level, which types of text indexes are constructed and maintained during document loading and updating. Each type of index accelerates the performance of a certain type of query. You can specify whether or not each different type of index is maintained for a given database.



NOTE

The index settings are designed to apply to an entire database. If you change any index settings on a database in which documents are already loaded, you must reindex your existing data, either by setting the `reindexer enable` setting to `true` for that database or by reloading the data.

Understanding your likely query set will help you determine which of these index types to maintain. The cost of supporting additional indexes is increased disk space and document load times. As more and more indexes are maintained, document load speed decreases. By default, MarkLogic Server builds a set of indexes that is designed to yield the fast query performance in general usage scenarios.

Text index types are configured on a per-database basis. This configuration should be completed before any documents are loaded into the specified database, although it can be changed later. If you change any index settings on a database in which documents are already loaded, you must reindex your existing data, either by setting the `reindexer enable` setting to `true` for that database or by reloading the data.

In addition to the standard indexes, you can configure indexes on individual elements and attributes in a database. You can create range indexes and/or lexicons on individual elements or attributes in a database. For information on these indexes, see [Section 25, “Range Indexes and Lexicons” \[237\]](#). You can also create named fields which can explicitly include or exclude specified elements. For details on fields, see [Section 14, “Fields Database Settings” \[98\]](#).

This section describes the text indexes in MarkLogic Server.

24.1.1. Understanding the Text Index Settings

The following table describes the different types of indexes available. The indexes are not mutually independent. If both the word search and stemmed search indexes are disabled, the configuration of

the remaining indexes is irrelevant, as they all depend on the existence of the word and/or stemmed-search index.

Index	Default Setting	Description
language	en	Specifies the default language for content in this database. Any content without an <code>xml:lang</code> attribute will be indexed in the language specified here. You should have a license key if you specify a non-English language; if you specify a non-english language and do not have a license for that language, the stemming and tokenization will be generic.
stemmed searches	Off (index is not built)	<p>Controls whether searches return relevance ranked results by matching word stems. A word <i>stem</i> is the part of a word that is common to all of its inflected variants. For example, in English, "run" is the stem of "run", "runs", "ran", and "running".</p> <p>A stemmed search returns more matching results than the exact words specified in the query. A stemmed search for a word finds the same terms as an unstemmed search, plus terms that derive from the same meaning and part of speech as the search term. For example, a stemmed search for <code>run</code> returns results containing <code>run</code>, <code>running</code>, <code>runs</code>, and <code>ran</code>. For details on stemming, see Understanding and Using Stemmed Searches in the <i>Search Developer's Guide</i>.</p> <p>There are three types of stemming: basic (one stem per word), advanced (one or more stems per word), and compounding (advanced plus smaller component words of large compound words).</p> <p>Without either this index or the word searches index, MarkLogic Server is unable to perform relevance ranking and will refuse to execute any <code>cts:word-query()</code>-related built-in function.</p> <p>If both the stemmed search and word search indexes are enabled, MarkLogic Server defaults to performing stemmed searches (unless an unstemmed search is explicitly specified).</p> <p>Turn this index off if you want to disable stemmed searches. If word and stemmed search indexes are both off, then full-text searches are effectively disabled.</p>
word searches (unstemmed)	On (index is built)	<p>Enables MarkLogic Server to return relevance ranked results which match exact words in text elements. Either this index or the stemmed search index is needed for MarkLogic Server to execute any <code>cts:word-query()</code>-related function.</p> <p>For many applications, keeping this word search index off and the stemmed search index on is sufficient to return the desired results for queries.</p> <p>Turn this index on if you want to do exact word-only matches. If word and stemmed search indexes are both off, then full-text searches are effectively disabled.</p>
word positions	Off (index is not built)	<p>Speeds up the performance of proximity queries that use the <code>cts:near-query</code> function and of multi-word phrase searches.</p> <p>Turn this index off if you are not interested in proximity queries or phrase searches and if you want to conserve disk space and decrease loading time. If you turn this option on, you might find that you no longer need <code>fast phrase searches</code>, as they have some overlapping functionality.</p>
fast phrase searches	On (index is built)	<p>Accelerates phrase searches by building additional indexes that describe sequences of words at load (or reindex) time. Without this index, MarkLogic Server will still perform phrase searches, just more slowly.</p> <p>Turn this index off if only a small percentage of your queries will contain phrase searches, and if conserving disk space and enhancing load speed is more important than the performance of those queries.</p>
fast case sensitive searches	On (index is built)	<p>Accelerates case sensitive searches by building both case sensitive and case insensitive indexes at load time. Without this index, MarkLogic Server will still perform case sensitive searches, just more slowly.</p> <p>Turn this index off if only a small percentage of your text searches will be case sensitive, and if conserving disk space and enhancing load speed is more important than the performance of those queries.</p>
fast reverse searches	Off (index is not built)	<p>Speeds up reverse query searches by indexing stored queries. Turn this option on to speed up searches that use <code>cts:reverse-query</code>.</p>
fast diacritic sensitive searches	On (index is built)	<p>Speeds up diacritic-sensitive searches by eliminating some false positive results. Turn this option off if you do not want to do diacritic-sensitive searches.</p>

Index	Default Setting	Description
fast element word searches	On (index is built)	Accelerates searches that look for words in specific elements by building additional indexes at load time. Without this index, MarkLogic Server will still perform these searches, just more slowly. Turn this index off if only a small percentage of your queries rely on finding words within specific document elements, and if conserving disk space and enhancing load speed is more important than the performance of those queries.
element word positions	Off (index is not built)	Speeds up the performance of proximity queries that use the <code>cts:near-query</code> function in an element and of multi-word element phrase searches. Turn this index off if you are not interested in proximity queries and if you want to conserve disk space and decrease loading time.
fast element phrase searches	On (index is built)	Accelerates phrase searches on elements by building additional indexes that describe sequences of words in elements at load (or reindex) time. Without this index, MarkLogic Server will still perform phrase searches, just more slowly. Turn this index off if only a small percentage of your queries will contain phrase searches at the element level, and if conserving disk space and enhancing load speed is more important than the performance of those queries.
element value positions	Off (index is not built)	Speeds up the performance of proximity queries that use the <code>cts:element-value-query</code> function. Turn this index off if you are not interested in proximity queries and if you want to conserve disk space and decrease loading time.
attribute value positions	Off (index is not built)	Speeds up the performance of proximity queries that use the <code>cts:element-attribute-value-query</code> function and speeds up <code>cts:element-query</code> searches that use attribute query constructors. Turn this index off if you are not interested in proximity queries and if you want to conserve disk space and decrease loading time.
field value searches	Off (index is not built)	Speeds up the performance of field value searches that use the <code>cts:field-value-query</code> function. Without this index or the corresponding index on the field definition, queries that use <code>cts:field-value-query</code> will throw an exception. Turn this index off if you are not interested in field value queries and if you want to conserve disk space and decrease loading time.
field value positions	Off (index is not built)	Speeds up the performance of proximity queries that use the <code>cts:field-value-query</code> function. Turn this index off if you are not interested in proximity queries and if you want to conserve disk space and decrease loading time.
trailing wildcard searches	Off (index is not built)	Speeds up wildcard searches where the search pattern contains the wildcard character at the end (for example, <code>abc*</code>). Turn this index on to speed up wildcard searches that match a trailing wildcard. The <code>trailing wildcard search</code> index uses roughly the same space as the <code>three character searches</code> index, but is more efficient for trailing wildcard queries. It does not speed up queries where the wildcard character is at the beginning of the term.
trailing wildcard word positions	Off (index is not built)	Speeds up the performance proximity queries that use trailing-wildcard word searches, such as wildcard queries that use the <code>cts:near-query</code> function and multi-word phrase searches that contain one or more wildcard terms. Turn this index on if you are using trailing wildcard searches and proximity queries together in the same search.
fast element trailing wildcard searches	Off (index is not built)	Faster wildcard searches with the wildcard at the end of the search pattern within a specific element, but slower document loads and larger database files.

Index	Default Setting	Description
three character searches	Off (index is not built)	<p>Speeds up wildcard searches where the search pattern contains three or more consecutive non-wildcard characters (for example, abc*x, *abc, a?bcd). When combined with a codepoint word lexicon, speeds the performance of any wildcard search (including searches with fewer than three consecutive non-wildcard characters). MarkLogic recommends combining the <code>three character search</code> index with a codepoint collation word lexicon. For details on wildcard characters, see Understanding and Using Wildcard Searches in the <i>Application Developer's Guide</i>.</p> <p>When character indexing is turned on, performance is also improved for <code>fn:contains()</code>, <code>fn:matches()</code>, <code>fn:starts-with()</code> and <code>fn:ends-with()</code> for most query expressions.</p> <p>Turn this index on if you want to enable wildcard searches that match three or more characters. If you need wildcard searches to match only two or one characters, then you should enable <code>two character searches</code> and/or <code>one character searches</code>.</p>
three character word positions	Off (index is not built)	<p>Speeds up the performance of proximity queries that use three-character word searches, such as queries that use the <code>cts:near-query</code> function and multi-word phrase searches that contain one or more wildcard terms.</p> <p>Turn this index on if you are using wildcard searches and proximity queries together in the same search.</p>
two character searches	Off (index is not built)	<p>Enables wildcard searches where the search pattern contains two or more consecutive non-wildcard characters. For details on wildcard characters, see Understanding and Using Wildcard Searches in the <i>Application Developer's Guide</i>.</p> <p>When character indexing is turned on in the database, the system also delivers higher performance for <code>fn:contains()</code>, <code>fn:matches()</code>, <code>fn:starts-with()</code> and <code>fn:ends-with()</code> for most query expressions.</p> <p>Turn this index on to speed up wildcard searches that match two or more characters (for example, ab*). This index is not needed if you have <code>three character searches</code> and a word lexicon.</p>
one character searches	Off (index is not built)	<p>Speeds up wildcard searches where the search pattern contains only a single non-wildcard character. For details on wildcard characters, see Understanding and Using Wildcard Searches in the <i>Application Developer's Guide</i>.</p> <p>When character indexing is turned on in the database, the system also delivers higher performance for <code>fn:contains()</code>, <code>fn:matches()</code>, <code>fn:starts-with()</code> and <code>fn:ends-with()</code> for most query expressions.</p> <p>Turn this index on if you want to enable wildcard searches that match one or more characters (for example, a*). This index is not needed if you have <code>three character searches</code> and a word lexicon.</p>
fast element character searches	Off (index is not built)	<p>Turn this index on to improve performance of wildcard searches that query specific XML elements or JSON properties. Also, speeds up element-based wildcard searches. Turn this index on to improve performance of wildcard searches that query specific elements. For details on wildcard characters, see Understanding and Using Wildcard Searches in the <i>Application Developer's Guide</i>.</p>
word lexicons	Off (index is not built)	<p>Maintains a lexicon of all of the words in a database, with uniqueness determined by a specified collation. For details on lexicons, see Section 25, "Range Indexes and Lexicons" [237] and the <i>Application Developer's Guide</i>. For details on collations, see Language Support in MarkLogic Server in the <i>Search Developer's Guide</i>.</p> <p>Speeds up wildcard searches. Works in combination with any other available wildcard indexes to improve search index resolution and performance. When used in conjunction with the <code>three character search</code> index, improves wildcard index resolution and speeds up wildcard searches. If you have <code>three character search</code> and a word lexicon enabled for a database, then there is no need for either the <code>one character</code> or <code>two character search</code> indexes. For best performance, the word lexicon should be in the codepoint collation (http://marklogic.com/collation/codepoint). For details on wildcard searches, see the <i>Application Developer's Guide</i>.</p>
uri lexicon	On (index is built)	<p>Maintains a lexicon of all of the URIs used in a database. The URI lexicon speeds up queries that constrain on URIs. It is like a range index of all of the URIs in the database. To access values from the URI lexicon, use the <code>cts:uris</code> or <code>cts:uri-match</code> APIs.</p>
collection lexicon	On (index is built)	<p>Maintains a lexicon of all of the collection URIs used in a database. The collection lexicon speeds up queries that constrain on collections. It is like a range index of all of the collection URIs in the database. To access values from the collection lexicon, use the <code>cts:collections</code> or <code>cts:collection-match</code> APIs.</p>

24.1.2. Viewing Text Index Configuration

To view text index configuration for a particular database, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Scroll down until the text index settings are visible (starting with the **Language** field).

24.1.3. Configuring Text Indexes

To configure text indexes for a particular database, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Scroll down until the text index settings are visible (starting with the **Language** field).
4. Configure the text indexes for this database by selecting the appropriate radio buttons for each index type.

Click the **true** radio button for a particular text index type if you want that index to be maintained. Click the **false** radio button for a particular text index type if you do not want that index to be maintained.



NOTE

If word searches and stemmed searches are disabled (that is, the **false** radio button is selected for **word searches** and **off** is selected for **stemmed searches**), the settings for the other text indexes are ignored, as explained above.

5. Leave the rest of the parameters unchanged.
6. Scroll to the top or bottom of the screen and click **OK**.
The database now has the new text indexing configurations.

24.2. Phrasing and Element-Word-Query Boundary Control

MarkLogic Server allows you to specify how XML element constructors impact text phrasing and element-word-query boundaries for searches.

24.2.1. Phrasing Control

By default, MarkLogic Server assumes that any XML element constructor acts as a phrase boundary. This means that phrase searches (for example, searches for sequences of terms) will not match a sequence of terms that contains one or more XML element constructors. Phrasing control lets you specify which XML elements should be transparent to phrase boundaries (for example, a bold or italic element), and which XML elements should be ignored for phrase purposes (for example, footnotes or graphic captions).

For example, consider this sample XML fragment:

```
<paragraph>
  These two words <italic>are italicized</italic>. The italic element
  <footnote>Elements are defined in the W3C XML standard.</footnote>
  is a standard part of this document's schema.
</paragraph>
```

By default, MarkLogic Server would extract the following five sequences of text for phrase matching purposes (ignoring punctuation and case for simplicity):

- “these two words”
- “are italicized”
- “the italic element”
- “elements are defined in the w3c xml standard”

- “is a standard part of this document's schema”

If you then attempted to match the phrases “words are italicized” or “element is a standard part” against this XML fragment, no matches would be found, because of the embedded XML element constructors.

In fact, a human looking at this XML fragment would realize that the `italic` element should be transparent for phrasing purposes, and that the `footnote` element is a completely independent text container. Seen from this viewpoint, the XML fragment shown above contains only two text sequences (again, ignoring punctuation and case for simplicity):

- “these two words are italicized the italic element is a standard part of this document's schema”
- “elements are defined in the w3c xml standard”

In this case, “words are italicized” and “element is a standard part” would each properly generate a match. But a search for “the w3c xml standard is a standard” would not result in a match.

MarkLogic Server lets you achieve this type of phrasing control by specifying particular XML element names as `phrase-through`, `phrase-around`, and `element-word-query-through` elements:

Type	Definition
<code>phrase-through</code>	Elements that should not create phrase boundaries (as in the example above, <code>italic</code> should be specified as a <code>phrase-through</code> element).
<code>phrase-around</code>	Elements whose content should be completely ignored in the context of the current phrase (as in the example above, <code>footnote</code> should be specified as a <code>phrase-around</code> element).

Phrase controls are configured on a per-database basis. You should complete this configuration before loading any documents into the specified database; otherwise, in order for the changes to take effect with your existing content, you must either reload the content or reindex the database after changing the configuration.

24.2.2. Element Word Query Throughs

Element-word-query-throughs allow you to specify elements that should be included in text searches that use `cts:element-word-query` on a parent element. For example, consider this XML fragment:

```
<a>
  <b>hello</b>
  <c>goodbye</c>
</a>
```

If you perform a `cts:element-word-query` on `<a>` searching for the word `hello`, the search does not find any matches in this fragment. The following query shows this pattern:

```
cts:search(fn:doc(), cts:element-word-query(xs:QName("a"), "hello"))
```

This query does not find any matches because `cts:element-word-query` only searches for text nodes that are immediate children of the element `<a>`, not text nodes that are children of any child nodes of `<a>`. Because `hello` is in a text node that is a child of ``, it does not satisfy the `cts:element-word-query`.

If you add an `element-word-query-through` for the element ``, however, then the `cts:element-word-query` on `<a>` searching for the word `hello` returns a match. The `element-word-query-through` on `` causes the text node children of `` behave like the text node children of its parent (in this case, `<a>`).

**NOTE**

If an element is specified as a phrase-through, then it also behaves as an element-word-query-through, and therefore you do not need to specify it as an element-word-query-through.

24.2.3. Procedures

The procedures to configure phrase controls for a particular database are described in this section.

Viewing Phrasing and Element-Word-Query Settings

To view element-word-query-through, phrase-through, and phrase-around settings for a particular database, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click **Element-Word-Query-Throughs**, **Phrase-Throughs**, or **Phrase-Arounds**.
4. The configuration page displays.

The following example shows that the Documents database has been configured with a number of phrase-through elements, including the `<abbr>`, `<acronym>`, ``, `<big>`, `
` and `<center>` elements of the XHTML namespace:

Configuring Phrasing and Element-Word-Query Settings

To configure element-word-query-through, phrase-through, and phrase-around settings for a particular database, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Underneath the target database, click **Element-Word-Query-Throughs**, **Phrase-Throughs**, or **Phrase-Arounds**.
4. The remainder of this procedure will assume that you have chosen to configure phrase-through settings. If you wish to configure phrase-around or element-word-query-through settings, the steps are completely analogous, once you have clicked on the corresponding icon.
5. Click the **Create** tab
6. In the **Namespace Uri** field, Enter the URI of the XML element that you are specifying as a phrase-through element.
Every XML element is associated with a namespace. For the phrase-through setting to be precise, you must specify the namespace of the XML element. Leaving the namespace URI field blank specifies the universal unnamed namespace.
Alternatively, you can specify that the element is namespace independent by putting an asterisk (*) in the namespace URI field.

7. Enter the element name in the **Localname** field.
The local name is the name of the XML element that you are specifying as a phrase-through element. If you want to specify more than one element that is associated with the specified namespace, you can provide a comma-separated list of element names.
8. To add more phrase-throughs, click **More Items** and repeat [Step 6](#) - [Step 7](#) for each phrase-through element as needed.
9. Scroll to the top or bottom and click **OK**.
The new phrase-through is added.
If you change the element-word-query-through, phrase-through, or phrase-around settings for a particular database after documents have already been loaded, you should reindex your existing data, either by setting the `reindexer enable` setting to `true` for that database or by reloading the data.

Deleting a Phrasing or Element-Word-Query Setting

To delete an element-word-query-through, phrase-through, or phrase-around setting for a particular database, follow these steps in the Admin Interface:

1. Click Databases in the left tree menu. A list of databases appears.
2. Click your target database.
3. Underneath the target database in the left tree menu, click **Element-Word-Query-Throughs**, **Phrase-Throughs**, or **Phrase-Arounds**.
4. Scroll down to the element that you want to delete.
5. Click **Delete** next to the element that you want to delete.
A confirmation message appears.
6. Confirm the delete operation and click **OK**.
The Phrase-Through or Phrase-Around element is deleted from the database.



NOTE

If you change the element-word-query-through, phrase-through, or phrase-around settings for a particular database after documents have already been loaded, you should reindex your existing data, either by setting the `reindexer enable` setting to `true` for that database or by reloading the data.

24.3. Query Behavior with Reindex Settings Enabled and Disabled

When you load a document into a database, it is indexed based on the index settings at the time of the load. When you issue a query to a database, it is evaluated based on a consistent view of the index settings. This consistent view might not include all of the index features that are enabled in the database. This section describes the behavior of queries at various index-setting states of the database.

24.3.1. Understanding the Reindexer Enable Settings

At the database level, you can enable or disable automatic reindexing by setting the `reindexer enable` setting to `true` or `false` for that database. When the reindexer is enabled, any index or fragment changes to the database settings will cause all documents in the database that are not indexed/fragmented according to the settings to initiate a reindex operation. Note the following about the database settings and the reindex operation:

- When reindexing is enabled, the reindex operation runs as a background task. You can set a higher or lower priority on the reindexing task by increasing or decreasing the setting of the `reindexer throttle`.
- Any new documents added to or updated in the database will get the new database settings. This is true both with reindexing enabled and with reindexing disabled.
- After changing index or fragmentation settings in a database, because new or modified documents get the new settings, the database can get into a state where some documents are indexed/fragmented differently from other documents in the database.
- After changing index or fragmentation settings in a database in which reindexing is enabled, the old documents are reindexed according to the new settings, but the new settings do not take effect for queries until the reindex operation has completed and all documents are indexed to the state matching the database settings.
- After changing index or fragmentation settings in a database in which reindexing is disabled, new and changed documents get the current settings, but queries will not take advantage of the new settings until all documents in the database match the database settings.
- Even if reindexing is disabled, when you add tokenizer overrides to a field, those tokenization changes take effect immediately, so all new queries against the field will use the new tokenization (even if it is indexed with the previous tokenization).

24.3.2. Query Evaluation According to the Lowest Common Denominator

When queries are evaluated, they use the index settings that are calculated for the database at a given time. The current index settings for a query are determined at the time of query evaluation, and are based on the lowest common denominator of (that is, the index/fragmentation settings that are the least of) the following:

- The index/fragmentation settings defined in the database configuration.
- The actual index/fragmentation of documents/fragments in the database.

At any given time, the current lowest common denominator is invalidated upon the following events:

- system startup
- a change to the database configuration settings
- when a reindexing operation completes

If the lowest common denominator is invalidated, it is recalculated the next time a query is issued against the database.

The net impact is that, when index/fragmentation settings have changed on a database after any data is loaded, queries cannot take advantage of the new settings until the new settings meet the lowest common denominator criteria. Depending on the types of index setting changes you make, this can cause queries that behaved one way before index settings were changed to behave differently after the changes. The next section provides a sample scenario to help illustrate this behavior.

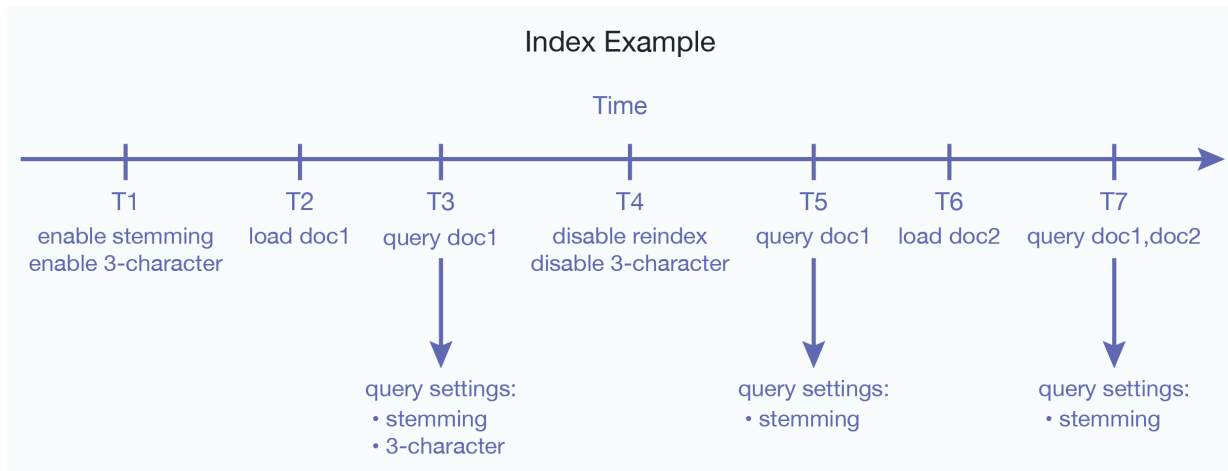
24.3.3. Reindexing Does Not Apply to Point-In-Time Versions of Fragments

If you have set a `merge timestamp` on the database to retain older versions of fragments for point-in-time queries, the older versions of the fragments will retain the indexing properties of the database at the time when they were updated. Because of this, reindexing a database that uses point-in-time queries can cause unpredictable query results. MarkLogic recommends that you do not reindex a database that has the `merge timestamp` parameter set to anything but 0. For details on point-in-time queries, see the [Point-In-Time Queries](#) in the *Application Developer's Guide*. For details on setting the `merge timestamp` parameter, see [Section 15.4, "Merges and Point-in-Time Queries" \[112\]](#).

24.3.4. Example Scenario

This section describes a simple scenario showing the effect of changing index settings on query behavior over time.

The following figure shows how changing the index settings can affect queries that initiate after index setting changes occur:



In this scenario, the query issued at time T3 sees the `doc1` document with stemming and 3-character wildcard indexes enabled. Wildcard queries such as `abc*` will be successful. The same wildcard query at time T5, however, will not be successful, because the 3-character index (which is required for the `abc*` query) was disabled at time T4. Note that the document `doc1` is actually indexed with 3-character and stemming, but the query at time T5 only is able to use the stemming index. At time T7, the database has `doc1` indexed with both stemming and 3-character indexes, but `doc2` only has the stemming index. With reindexing disabled, the query at T7 will use the lowest common denominator, which is in this case stemming.

25. Range Indexes and Lexicons

MarkLogic Server allows you to create, at the database level, indexes and lexicons on elements and attributes according to their QNames. Additionally, you can create range indexes on fields, as described in [Section 14.4.4, “Creating a Range Index on a Field” \[107\]](#).

This section describes how to use the Admin Interface to create range indexes and lexicons. For details on how to create range indexes programmatically, see [Adding Indexes to a Database](#) in the *Scripting Administrative Tasks Guide*.

25.1. Understanding Range Indexes

This section describes the types of range indexes shown in the table below. There are also field range indexes, as described in [Section 14.4.4, “Creating a Range Index on a Field” \[107\]](#).

Type	Description
Element range index	A range index on an XML element or JSON property.
Attribute range index	A range index on an attribute in an XML element.
Path range index	A range index on an XML element, XML attribute, or JSON property as defined by an XPath expression.
Field range index	A range index on a field. For details, see Section 14, “Fields Database Settings” [98] .

MarkLogic Server maintains a universal index for every database to rapidly search the text, structure, and combinations of the text and structure that are found within collections of XML and JSON documents.

In some cases, however, XML and JSON documents can incorporate numeric or date information. Queries against these documents may include search conditions based on inequalities (for example, `price < 100.00` or `date ≥ thisQtr`). Specifying range indexes for these elements, attributes, and/or JSON properties will substantially accelerate the evaluation of these queries.

Defining a range index also allows you to use the range query constructors (`cts:element-range-query` and `cts:element-attribute-range-query`) in `cts:search` operations, making it easy to compose complex range-query expressions to use in searches. For details, see [Using Range Queries in cts:query Expressions](#) in the *Search Developer’s Guide*.

Similarly, you can create range indexes of type `xs:string`. These indexes can accelerate the performance of queries that sort by the string values, and are also used for lexicon queries (see [Section 25.3, “Understanding Word Lexicons” \[239\]](#)).

If you specify a range index on an element, and if you have elements of that name that have complex content (for example, elements with child elements), the content is indexed based on a casting of the element to the specified type of the range index. For example, if you specify a range index of type `xs:string` on an element named `h1`, then the following element

```
<h1>This is a <b>bold</b> title.</h1>
```

is indexed with the value of `This is a bold title`, which is the value returned by casting the `h1` element to `xs:string`. The same type casting applies to range indexes on XML attributes, JSON properties, and fields. This behavior allows you to index complex content without pre-processing the content.

Also, range indexes can improve the performance of queries that sort the results using an `order by` clause and return a subset of the data (for example, the first ten items). For details on this order by optimization using range indexes, see [Sorting Searches Using Range Indexes](#) in the *Query Performance and Tuning Guide*.

MarkLogic Server supports range indexes for both elements and attributes across a wide spectrum of XML data types. For the most part, this list conforms to the XML totally ordered data types:

Type	Description
int	Positive and negative integers
unsignedInt	Positive integers (including 0)
long	Large positive and negative integers
unsignedLong	Large positive integers (including 0)
float	32-bit floating point numbers
double	64-bit floating point numbers
decimal	Large floating point numbers
dateTime	Combined date and time
time	Time (including timezone)
date	Full date (year, month, day)
gYearMonth	Year and month only
gYear	Year only
gMonth	Month only
gDay	Day only
yearMonthDuration	Duration of years and months
dayTimeDuration	Duration of days and time
string	String character data
anyURI	A URI string

It is important to note that the date and time types listed above adhere to the XML specification for dates and times. At present, other date and time formats are not supported by MarkLogic Server range indexes. For a more detailed description of the definition of these data types, consult the W3C XML Schema documents.

Range indexes must be explicitly created using the Admin Interface, the XQuery or JavaScript Admin API, or the REST Management API. To create a range index on a JSON property, use the element range index interfaces or functions. The following table outlines the basic information needed to define each kind of index:

Index Type	Required Information
XML element	The element name, the namespace for the element, the data type of the values found in that element.
XML attribute	The attribute name, the name of the attribute's parent element, a namespace for the element, and the data type of the values found in that attribute.
JSON property	The property name and the data type of the values found in that property.
path	An XPath expression and the data type of the values found in the element, attribute, or JSON property expressed by the XPath.
field	The field name and data type of the values in the field. You must also configure the field definition. For details, see Section 14.4, "Configuring Fields" [105] .

Range indexes are populated during the document loading process, and are automatically kept in sync through subsequent updates to indexed data. Consequently, range indexes should be specified for a database before any XML or JSON documents containing the content to be indexed are loaded into that database. Otherwise, the content must be either reindexed or reloaded to take advantage of the new range indexes.

Use the element range index interfaces and APIs to create indexes for JSON documents. Some restrictions apply. For details, see [Creating Indexes and Lexicons Over JSON Documents](#) in the *Application Developer's Guide*.

You can create the same type of index with a path range index as you can with an element or attribute range index. Path range indexes are useful in circumstances in which an element or attribute range

index will not work. For example, you may have documents with the same element name appearing under different parent elements and you only want to index the elements appearing under one of the parent elements. In this case, a path range index is required to correctly index that element.

When creating a range index with a scalar type of string (`xs:string`), specify a collation as well as the element/attribute QNames or JSON property name. The collation specifies the unique ordering for the string values. You can have multiple range indexes on the same element, attribute, or JSON property with different collations; that is, the collation is part of the unique identifier for the string range index. For details about collations, see the [Encodings and Collations](#) in the *Search Developer's Guide*.

Because a range index stores typed data, if the data you load does not conform to that type, or if it cannot be coerced to conform to the specified type, it cannot be loaded into the document. For each range index, you can specify what to do for invalid values, either `reject` them and have the document load throw an exception and fail, or `ignore` them and log the coercion errors in the `ErrorLog.txt` file at the `Debug` level. The default is to `reject` invalid data.

Range indexes use disk space and consume memory. That is the trade-off for improved performance. Additionally, if you have a large amount of range index data and if your system is updated regularly, you might need to increase the size of your journals. For details on the database journal settings, see [Memory and Journal Settings \[85\]](#).

25.2. Using Range Indexes for Value Lexicons

In addition to speeding up sorting and comparison queries, MarkLogic Server uses range indexes to resolve XML element, XML attribute, JSON property, and field value lexicon queries. These are queries that use the following search APIs:

- `cts:values`
- `cts:value-match`
- `cts:element-attribute-values`
- `cts:element-attribute-value-match`
- `cts:element-values`
- `cts:element-value-match`
- `cts:field-values`
- `cts:field-value-match`

The `cts:values` and `cts:value-match` functions work on any kind of range index and are equivalent to the corresponding index-specific function when called with a reference to the same type of index. For example, the following two function calls are equivalent:

```
cts:values(cts:element-reference(xs:QName("some-element")))
cts:element-values(xs:QName("some-element"))
```

In order to use any of these APIs, you must create range indexes on the element(s), attribute(s), JSON property(s), or field(s) specified in the query. The type of the range index must match the type specified in the lexicon API.

For details about lexicons, see the [Browsing With Lexicons](#) in the *Search Developer's Guide*. For more details on the lexicon APIs, see the *MarkLogic XQuery and XSLT Function Reference*.

25.3. Understanding Word Lexicons

MarkLogic Server allows you to create a word lexicon that is restricted to a particular XML element, XML attribute, JSON property, or field. You can also define a field word lexicon across a collation. A word lexicon stores all of the unique words that are stored in the specified element, attribute, or JSON property. The words are stored case-sensitive and diacritic sensitive, so the words `Ford` and `ford` would be separate entries in the lexicon.

Word lexicons are used in wildcard searches (when wildcarding is enabled). For details, see [Understanding and Using Wildcard Searches](#) in the *Search Developer's Guide*.

To use a word lexicon, use the following search APIs:

- `cts:element-attribute-words`
- `cts:element-attribute-word-match`
- `cts:element-words`
- `cts:element-word-match`
- `cts:field-words`
- `cts:field-word-match`
- `cts:json-property-words`
- `cts:json-property-word-match`

25.4. Understanding Path Range Indexes

A path range index enables you to define a range index on an XML element, XML attribute, or JSON property using an XPath expression. A path range index can give you finer control over what is indexed. For example, if your content contains elements with the same name at multiple levels, but you only want to index one of them, you can use a path range index to target just that one.

This section describes the XPath expressions you can use to define a path range index. For performance reasons, MarkLogic Server restricts you to a subset of XPath when defining a path range index.

25.4.1. Limitations on Index Path Expressions

You can only use subset of XPath for defining path range indexes. The limitations are described in [Path Field and Path-Based Range Index Configuration](#) in the *XQuery and XSLT Reference Guide*.



NOTE

Avoid creating multiple path indexes that end with the same element/attribute, as ingestion performance degrades with the number of path indexes that end in common element/attributes.

You can use `cts:valid-index-path` to test whether or not you can use an XPath expression to define a path range index. For details, see [Section 25.4.3, "Testing the Validity of an Index Path Expression" \[241\]](#).

Note numbers, booleans, and nulls in JSON documents are indexed separately rather than all being treated as text. For details on constructing XPath expressions on JSON documents, see [Traversing JSON Documents Using XPat](#) in the *Application Developer's Guide*.

25.4.2. Examples of Index Path Expressions

The following table provides examples of XPath expressions that are valid and invalid for defining a path range index.



NOTE

Avoid creating multiple path indexes that end with the same element/attribute, as ingestion performance degrades with the number of path indexes that end in common element/attributes.

Valid	Invalid
//a	./a
/a/b/c	/a/b[c=/p/q]
/a/b[c]	/a/b[c=5+3]
/a/b[c=5 and b=3]	
/a/b[1]	
//a/b[c<5]	
//a/b[c="test"]	
/a/*/c	
a/b	
/a[./b]/c	/a[b]/c
a	
/a/(b c)	/a/(/b /c)
	(/a/b/c)[2]
author[first-name="John"][last-name="Smith"]	
author[first-name="John" and last-name="Smith"]	
author[first-name="John" or first-name="Sam"]	
/a/b[./c]	/a/b[./c]
/a/b[c]	/a/b[//c]
	/a/b[/a/b/c]
/a(/.b c)/d	/a(/a/b /a/c)/d
/a/child::*/b	/a/parent::*/b
/a[fn:matches(@expr, 'is')]	/a/[fn:matches(fn:name(.), "Joe")]
/a[fn:contains("this")]	/a/[fn:contains(fn:name(.), "Bob")]

Namespace prefixes are permitted in all valid path expressions. Note that you can also use `fn:matches` and `fn:contains` as part of the path expression, but you cannot use other functions in the path expression. Use `cts:valid-index-path` to test if a path expression is valid for an index path.

25.4.3. Testing the Validity of an Index Path Expression

You can use the XQuery function `cts:valid-index-path` to test whether or not an XPath expression can be used to define a path range index. To test validity, copy the following query into Query Console, modify it to use your path expression, and run it.

```
xquery version "1.0-ml";
cts:valid-index-path("/a/b", fn:true())
```

Use the second parameter to control whether or not to verify that namespace binding definitions are configured for namespace prefixes used in the path expression.

25.4.4. Using Namespace Prefixes in Index Path Expressions

XML namespace prefixes are permitted in all valid path range index expressions, but you must define the namespace binding in your database configuration. For example, if your path expression is `/ns:a/ns:b`, you must configure a namespace binding for the prefix `ns`.

To pre-define a namespace binding, use the Path Namespaces configuration page for your database in the Admin Interface or the XQuery function `admin:database-add-path-namespace`.

For details, see [Section 25.11, "Defining Path Range Indexes" \[244\]](#).

25.5. Viewing Element Range Index Settings

To view the element range indexes that will be applied to documents as they are loaded or reindexed, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click **Element Range Indexes**.

The configuration page appears.

25.6. Defining Element Range Indexes

To define an element range index for an XML element or JSON property, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. In the tree menu, under the selected database, click **Element Range Indexes**.
4. Click the **Add** tab.
5. From the **Scalar Type** list, select the type of the XML element or JSON property.
6. Enter the namespace URI of the XML element. Skip this step for a JSON property index. Every XML element is associated with a namespace. For the description of the element to be precise, you must specify the namespace of the XML element. The asterisk (*) cannot be used to indicate namespace independence. Leaving the namespace URI field blank specifies the universal unnamed namespace.
7. Enter the element or JSON property name in the **Localname** field. The value in the **Localname** field is the name of the XML element to be indexed. If you have more than one element of the same type in the same namespace that you want to index, you can provide a comma-separated list of element names.
8. If you selected a scalar type of **string** in [Step 5](#), the **Collation** field contains a default collation. You can enter a different collation URI or click the **Collation Builder** button for a wizard that constructs the collation URI for you based on the language and other parameters you enter. For details about collations, see [Language Support in MarkLogic Server](#) in the *Search Developer's Guide*.
9. Set the **Range Value Positions** field to **true** to increase the speed of searches that use `cts:near-query` and `cts:element-query` with this index. However, this setting uses more disk space than the default setting of **false**.
10. In the **Invalid Values** field, choose whether to allow insertion of documents that contain elements or JSON properties on which range index is configured, but the value of those elements cannot be coerced to the index data type. You can choose either `ignore` or `reject`. By default, the server rejects insertion of such documents. However, if you choose `ignore`, these documents can be inserted. This setting does not change the behavior of queries on invalid values after documents are inserted into the database. Performing an operation on an invalid value at query time can still result in an error.
11. To add more indexes, click **More Items** and repeat [Step 5 - Step 10](#) for each index as needed.
12. Scroll to the top or bottom and click **OK**.

The new element range index or element word lexicon is added to the database. These rules are applied to XML and JSON documents loaded into the specified database from this point on.



NOTE

If you have reindexing enabled for the database and you specify an element that exists in a document, reindexing will run in the background. When the reindexing is complete, the new index will become available to queries.

25.7. Viewing Attribute Range Index Settings

To view the attribute range indexes that will be applied to documents as they are loaded or reindexed, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.

The configuration page appears.

25.8. Defining Attribute Range Indexes

To define a range index for an attribute of a particular element, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. In the tree menu, under the selected database, click **Attribute Range Indexes**.
4. Click the **Add** tab.
5. From the **Scalar Type** list, select the type of the XML element.
6. Enter the namespace URI of the XML element that contains the attribute you want to index into the **Parent Namespace Uri** field.
Every XML element is associated with a namespace. For the description of the element to be precise, you must specify the namespace of the XML element. The asterisk (*) cannot be used to indicate namespace independence. Leaving the namespace URI field blank specifies the universal unnamed namespace.
7. Enter the element name in the **Parent Localname** field.
The local name is the name of the XML element that contains the attribute to be indexed. If you have more than one element in the same namespace that contains the attribute you want to index, you can provide a comma-separated list of element names.
8. Enter the namespace URI of the attribute that you want to index into the **Namespace URI** field.
Every XML attribute is associated with a namespace. For the description of the attribute to be precise, you must specify the namespace of the XML attribute. The asterisk (*) cannot be used to indicate namespace independence. Leaving the **Namespace URI** field blank specifies the universal unnamed namespace.
9. Enter the attribute name in the **Localname** field.
The local name is the name of the XML attribute to be indexed. If you have more than one attribute in the same namespace within the specified parent element(s) that you want to index, you can provide a comma-separated list of attribute names.
10. If you selected a scalar type of **string** in [Step 5](#), the **Collation** field contains a default collation. You can enter a different collation URI or click the **Collation Builder** button for a wizard that constructs the collation URI for you based on the language and other parameters you enter. For details about collations, see [Language Support in MarkLogic Server](#) in the *Search Developer's Guide*.
11. Set the **Range Value Positions** field to **true** to increase the speed of searches that use `cts:near-query` and `cts:element-query` with this index. However, this setting uses more disk space than the default setting of **false**.
12. In the **invalid values** field, choose whether to allow insertion of documents that contain attributes on which range index is configured, but the value of those attributes cannot be coerced to the index data type. You can choose either **ignore** or **reject**. By default, the server rejects insertion of such documents. However, if you choose **ignore**, these documents can be inserted. This setting does not change the behavior of queries on invalid values after documents are inserted into the database. Performing an operation on an invalid value at query time can still result in an error.
13. To add more indexes, click **More Items** and repeat [Step 5](#) - [Step 12](#) for each attribute index as needed.
14. Scroll to the top or bottom and click **OK**.

The new attribute index is added to the database. These rules are applied to XML documents loaded into the specified database from this point on.

**NOTE**

If you have reindexing enabled for the database and you specify an element-attribute pair that exists in a document, reindexing will run in the background. When the reindexing is complete, the new index will become available to queries.

25.9. Viewing Path Range Index Settings

To view the path range indexes that will be applied to documents as they are loaded or reindexed, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click **Path Range Indexes** in the tree menu, under the selected database.

The configuration page appears.

25.10. Defining Namespace Prefixes Used in Path Range Indexes and Fields

When you define a path range index over XML documents and your path uses namespace prefixes, you must pre-define any namespace bindings used in the path expression. These namespace bindings can be used by multiple path range indexes.

To define a namespace binding, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click **Path Namespaces** in the tree menu, under the selected database.
4. Click the **Add** tab.
5. In the **Prefix** field, enter the namespace prefix you intend to use for the element or attribute in the XPath expression in your path range index.
6. In the **Namespace Uri** field, enter the namespace URI of the XML element or attribute in the XPath expression.
7. Click **OK**.

25.11. Defining Path Range Indexes

To define a range index expressed by an XPath expression, follow these steps:

1. If you are creating a path range index over XML data, create bindings for any namespaces prefixes used in your index XPath expression. For details, see [Section 25.10, “Defining Namespace Prefixes Used in Path Range Indexes and Fields” \[244\]](#).
2. Click **Databases** in the left tree menu. A list of databases appears.
3. Click your target database.
4. In the tree menu, under the selected database, click **Path Range Indexes**.
5. Click the **Add** tab.
6. From the **Scalar Type** list, select the type of the XML element, XML attribute, or JSON property for which you want to build a range index.
7. Enter the XPath expression in the **Path Expression** field. For XML, you can use any namespace prefix you created in [Step 1](#). XPath expressions are summarized in [XPath Quick Reference](#) in the *XQuery and XSLT Reference Guide*. Not all XPath features are supported by path range indexes. For details, see [Section 25.4, “Understanding Path Range Indexes” \[240\]](#).

**NOTE**

You can use the `cts:valid-index-path` function to test whether the path is syntactically correct for use in a path range index.

You cannot have a path span across a fragment root. Paths should be scoped within fragment roots.

8. If you selected a scalar type of **string** in [Step 6](#), the **Collation** field contains a default collation. You can enter a different collation URI or click the **Collation Builder** button for a wizard that constructs the collation URI for you based on the language and other parameters you enter. For details about collations, see [Language Support in MarkLogic Server](#) in the *Search Developer's Guide*.
9. Set the **Range Value Positions** field to **true** to increase the speed of searches that use `cts:near-query` and `cts:element-query` with this index. However, this setting uses more disk space than the default setting of **false**.
10. In the **invalid values** field, choose whether to allow insertion of documents that contain XML elements, XML attributes, or JSON properties on which range index is configured, but the value of those elements, attributes, or properties cannot be coerced to the index data type. You can choose either **ignore** or **reject**. By default, the server rejects insertion of such documents. However, if you choose **ignore**, these documents can be inserted. This setting does not change the behavior of queries on invalid values after documents are inserted into the database. Performing an operation on an invalid value at query time can still result in an error.
11. To add more indexes, click **More Items** and repeat [Step 6 - Step 10](#) for each index as needed.
12. Scroll to the top or bottom and click **OK**.

The new path range index is added to the database. These rules are applied to XML or JSON documents loaded into the specified database from this point on.

**NOTE**

If you have reindexing enabled for the database and you specify an XML element, XML attribute, or JSON property that exists in a document, reindexing will run in the background. When the reindexing is complete, the new index will become available to queries.

Once you have created a path range index, you cannot change the path expression. Instead, you must remove the existing path range index and create a new one with the updated path expression.

25.12. Viewing Element Range Index Settings

To view the element range indexes that will be applied to documents as they are loaded or reindexed, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click **Element Range Indexes**.

The configuration page appears.

25.13. Defining Element Word Lexicons

To define a lexicon for an XML element or JSON property, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. In the left tree menu, under the target database, click **Element Word Lexicons**.
4. Click the **Add** tab.
5. If you are defining a lexicon on an XML element, enter the namespace URI of the XML element into the **Namespace Uri** field.
Every XML element is associated with a namespace. For the description of the element to be precise, you must specify the namespace of the XML element. The asterisk (*) cannot be used to indicate namespace independence. Leaving the **Namespace Uri** field blank specifies the universal unnamed namespace.
6. Enter the XML element or JSON property name into the **Localname** field.
The local name is the name of the XML element or JSON property to be indexed. If you have more than one element of the same type in the same namespace that you want to index or more than one property name, you can provide a comma-separated list of names.
7. The **Collation** field contains a default collation. If you want the lexicon to use a different collation, enter the collation URI. Click **Collation Builder** to use a wizard to construct the URI. For details about collations, see [Language Support in MarkLogic Server](#) in the *Search Developer's Guide*.
8. To add more word lexicons, click **More Items** and repeat [Step 5](#) - [Step 7](#) for each lexicon as needed.
9. Scroll to the top or bottom and click **OK**.

The new range index or word lexicon is added to the database. These rules are applied to XML or JSON documents loaded into the specified database from this point on.



NOTE

If you have reindexing enabled for the database and you specify an element that exists in a document, reindexing will run in the background. When the reindexing is complete, the new index will become available to queries.

25.14. Viewing Attribute Word Lexicon Settings

To view the lexicon that will be applied to documents as they are loaded or reindexed, follow these steps:

1. Click the **Databases** icon on the left tree menu.
2. Locate the database for which you want to view a lexicon, either in the tree menu or in the **Database Summary** table.
3. Click the name of the database for which you want to view a lexicon.
4. Click the **Attribute Word Lexicons** icon in the tree menu, under the selected database.

The **Element-Attribute Word Lexicon** page appears.

25.15. Defining Attribute Word Lexicons

To define a lexicon for an attribute of a particular element, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Under the selected database, click **Attribute Word Lexicons**.
4. Click the **Add** tab.
5. In the **Parent Namespace Uri** field, enter the namespace URI of the XML element that contains the attribute you want to index into the field.

Every XML element is associated with a namespace. For the description of the element to be precise, you must specify the namespace of the XML element. The asterisk (*) cannot be used to indicate namespace independence. Leaving the **Parent Namespace Uri** field blank specifies the universal unnamed namespace.

6. Enter the element name in the **Parent Localname** field.
The local name is the name of the XML element that contains the attribute to be indexed. If you have more than one element in the same namespace that contains the attribute you want to index, you can provide a comma-separated list of element names.
7. Enter the namespace URI of the attribute that you want to index into the **Namespace Uri** field.
Every XML attribute is associated with a namespace. For the description of the attribute to be precise, you must specify the namespace of the XML attribute. The asterisk (*) cannot be used to indicate namespace independence. Leaving the **Namespace URI** field blank specifies the universal unnamed namespace.
8. Enter the attribute name into the **Localname** field.
The local name is the name of the XML attribute to be indexed. If you have more than one attribute in the same namespace within the specified parent element(s) that you want to index, you can provide a comma-separated list of attribute names.
9. The collation box appears with a default collation. If you want the lexicon to use a different collation than the default, enter the collation URI into the **Collation** field. Click the **Collation Builder** button to use a wizard that constructs the collation URI for you. For details about collations, see the [Language Support in MarkLogic Server](#) in the *Search Developer's Guide*.
10. To add more element-attribute word lexicons, click **More Items** and repeat [Step 5 - Step 9](#) for each attribute index as needed.
11. Scroll to the top or bottom and click **OK**.

The new attribute index or attribute word lexicon is added to the database. These rules are applied to XML documents loaded into the specified database from this point on.



NOTE

If you have reindexing enabled for the database and you specify an element-attribute pair that exists in a document, reindexing will run in the background. When the reindexing is complete, the new index will become available to queries.

25.16. Defining Value Lexicons

Value lexicons are implemented using range indexes of type `xs:string` on the element(s), attribute(s), JSON properties, or fields specified in a query. Therefore, to create a value lexicon, you create a range index of type `xs:string` for the specified element(s), attribute(s), JSON properties, or fields. Use an element range index for a JSON property value lexicon.

25.17. Deleting Range Indexes or Lexicons

To delete element or attribute indexes or lexicons for a specific database, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Determine the type of range index to delete and click **Element Range Indexes**, **Attribute Range Indexes**, **Path Range Indexes**, **Element Word Lexicons**, or **Attribute Word Lexicon**. The configuration page for the appropriate index appears.
4. Locate the index you want to delete and click **Delete**. A confirmation message appears.
5. Confirm the delete and click **OK**.

The index or lexicon is deleted from the database.

25.18. Defining Field Range Indexes

Fields provide a convenient mechanism for querying a portion of the database based on XML element QNames or JSON property names. You can define a field, and then create a range index or word or value lexicon over it. For details, see [Section 14, “Fields Database Settings” \[98\]](#).

26. Fragments

When loading data into a database, you have the option of specifying how XML documents are partitioned for storage into smaller blocks of information called fragments. For large XML documents, size can be an issue, and using fragments may help manage performance of your system. In general, fragments for XML documents should be sized between 10K and 100K. Fragments set too small or too big can slow down performance, so proper fragment sizing is important.

The actual fragmentation of an XML document is completely transparent to an application developer. At the application level, the document appears to be a single integral structure, regardless of how it is stored and managed as fragments on disk. Fragmentation is an application-transparent tuning mechanism.

However, fragmentation *does* impact relevance ranking. The relevance-ranking algorithm considers both term frequency within a target piece of content and overall term frequency within the database to rank results by relevance. Rather than consider term frequency across the entire XML document for ranking purposes, MarkLogic Server considers term frequency within the individual fragment (and its descendants) being ranked. Consequently, different fragmentation strategies may impact relevance rankings—particularly in situations when a single fragment may straddle multiple XML structures that you are trying to differentiate on a relevance basis.

With MarkLogic Server, you specify fragmentation *rules* that are used to partition your XML documents. These rules are applied one document at a time. However, fragmentation rules are specified at the database level—on the assumption that databases contain many documents with similar structures where the same fragmentation rules should be applied.

Fragmentation rules are applied to documents during document loads, updates, and database reindexing. Specifying additional fragmentation rules after documents have been loaded causes future updates and/or reindexing of those documents to use the new fragmentation rules, but does not change the fragmentation of existing documents (if `reindex enable` is set to `true`, however, the documents will eventually be reindexed and take on the new fragmentation policy). As a result, if you want to change the fragmentation rules for already loaded content, you will have to reload your documents or reindex the database so that your new fragmentation rules can take effect.

This section describes managing fragmentation rules.

26.1. Choosing a Fragmentation Strategy

Proper fragmentation is important to performance. Before you specify how to fragment the XML data being loaded, you need to plan your fragmentation strategy. Apply the following guidelines:

- Fragments are described generically using XML element names.
- Fragments for XML documents should be between 10K and 100K in size (these are just general guidelines; in some situations, larger or smaller fragment sizes can work fine, and there are many factors that will affect performance for a given fragment size including disk block size, how many fragments are in the database, how often fragments are accessed, the types of queries used in the application, and so on).
- Fragments can be (and in many cases, should be) nested hierarchically.
- Smaller fragment sizes allow more efficient element-level updates in the database, but excessively small fragments can slow down both loading speed and query performance.
- Larger fragment sizes can also slow down query performance by requiring excessive loading of data from disk in resolving queries.
- In general, within the size range set above, larger fragment sizes deliver higher-performance overall than smaller fragment sizes.
- Text and small binary documents must fit in a single fragment. Therefore, set the database `in memory tree size` parameter to 1 to 2 MB larger than your largest text or small binary

file. The largest small binary file size is always constrained by the “large size threshold” database configuration setting.

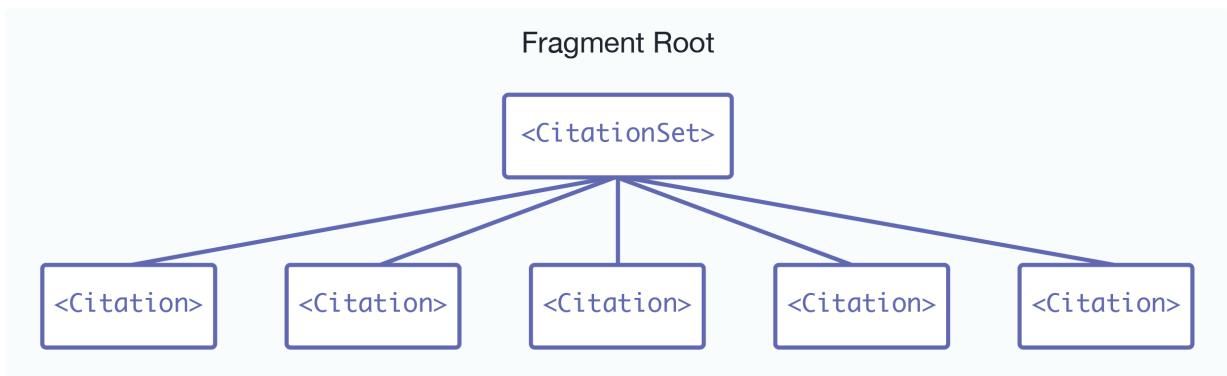
After you decide how to fragment your data, you can use the Fragment Roots or Fragment Parents method.

Both methods turn your fragmentation strategy into concrete rules for the system.

26.1.1. Fragment Roots

If a document contains many instances of an XML structure that share a common element name, then these structures make sensible fragments. With MarkLogic Server, you can use this common element name as a fragment root.

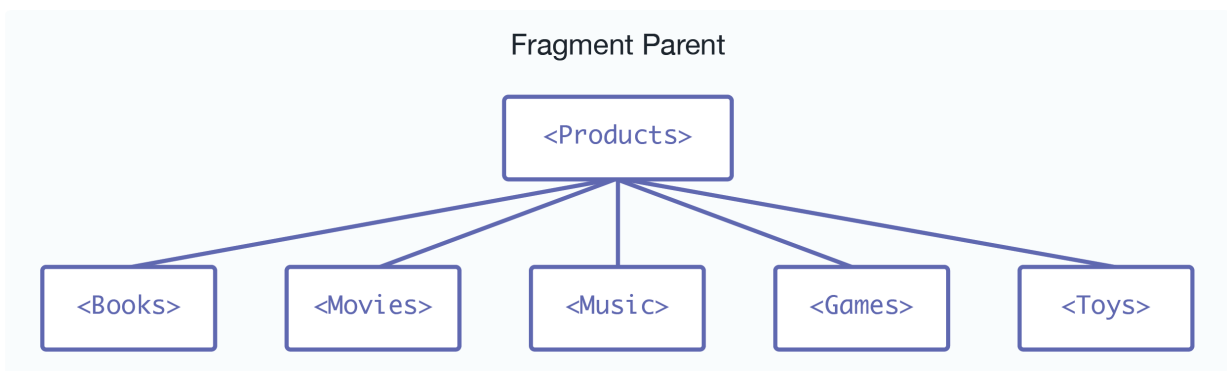
The following diagram shows an XML document rooted at `<CitationSet>` that contains many instances of a `<Citation>` node. Each `<Citation>` node contains further XML and averages between 15K and 20K in size. Based on this information, `<Citation>` is a sensible element to use as a fragment root:



26.1.2. Fragment Parents

If your document contains many different XML substructures, each of which is a good candidate to be a fragment, then it would be time consuming to specify each substructure as a fragment root. Instead, you can specify fragments by setting the parent of these substructures to be a fragment parent—so that every substructure under this parent becomes a separate fragment, regardless of its name.

The following diagram shows a document with substructures of different names:



In this case, you can use the `<Products>` element as a fragment parent, and the `<Books>`, `<Movies>`, `<Music>`, `<Games>`, and `<Toys>` children automatically become fragments.

26.2. Defining Fragment Roots

To define a rule for a fragment root, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click the **Fragment Roots** icon.
4. Click the **Create** tab.
5. Enter the namespace URI of the XML element that you are using as a rule for the fragment root into the **Namespace Uri** field.

Every XML element is associated with a namespace. For the fragment rule to be precise, you must specify the namespace of the XML element. Leaving the **Namespace Uri** field blank specifies the universal unnamed namespace.

Alternatively, you can specify that the rule for the fragment root is namespace-independent by putting an asterisk (*) into the **Namespace Uri** field.
6. Enter the element name into the **Localname** field.

The local name is the name of the XML element used as the root of a fragment. If you have more than one fragment root rule associated with the specified namespace, you can provide a comma-separated list of element names.
7. To add more fragment roots, click **More Items** and repeat [Step 5 - Step 6](#) for each fragment root as needed.
8. Scroll to the top or bottom and click **OK**.

The new fragment root rules are added to the database. These rules are applied to XML documents loaded into the specified database from this point on.

26.3. Defining Fragment Parents

To define a rule for a fragment parent, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click **Fragment Parents** .
4. Click the **Create** tab.
5. Enter the namespace URI of the XML element that you are using as a rule for the fragment parent into the **Namespace Uri** field.

Every XML element is associated with a namespace. For the fragment rule to be precise, you must specify the namespace of the XML element. Leaving the namespace URI field blank specifies the universal unnamed namespace.

Alternatively, you can specify that the rule for the fragment root is namespace-independent by putting an asterisk (*) into the **Namespace Uri** field.
6. Enter the element name into the **Localname** field.

The local name is the name of the parent XML element whose children will be fragment roots. If you have more than one fragment parent rule associated with the specified namespace, you can provide a comma-separated list of element names.
7. To add more fragment parents, click **More Items** and repeat [Step 5 - Step 6](#) for each fragment parent as needed.
8. Scroll to the top or bottom and click **OK**.

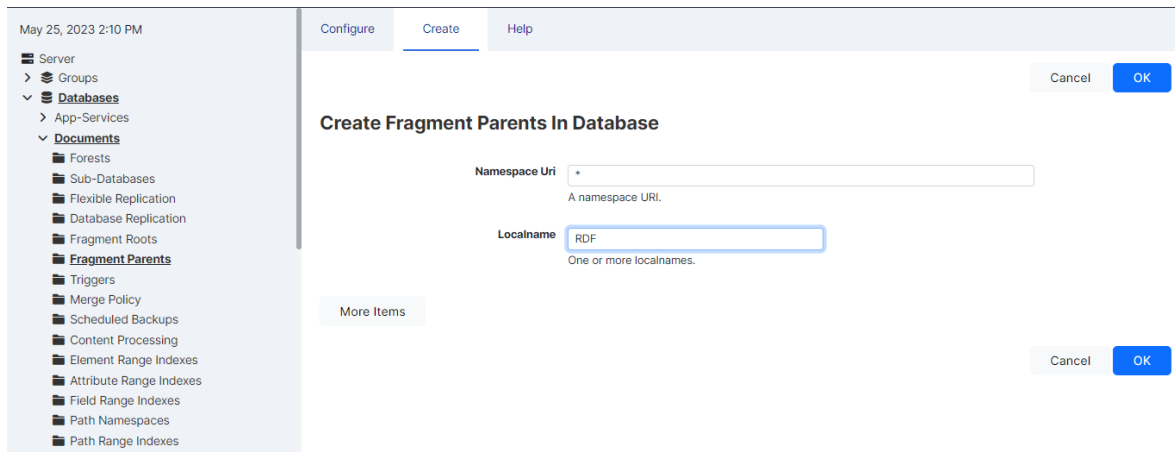
The new fragment rules are added to the database. These rules are applied to XML documents loaded into the specified database from this point on.

26.4. Viewing Fragment Rules

To view fragment rules that are in effect, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click either **Fragment Roots** or **Fragment Parents** icon, under the specified database.

The following example shows that the Documents database has only one rule defined for a fragment parent. The rule states that any direct child of an <RDF> element, regardless of the namespace for the <RDF> element, should form the root of a fragment:



26.5. Deleting Fragment Rules

To delete fragment rules for a specific database, follow these steps:

1. Click **Databases** in the left tree menu. A list of databases appears.
2. Click your target database.
3. Click either **Fragment Roots** or **Fragment Parents** under the specified database.
4. Locate the fragment rule you want to delete and click **Delete**. A confirmation message appears.
5. Confirm the delete and click **OK**.

The fragment rule is dropped from the database.



NOTE

Deleting fragment rules has no impact on the fragmentation that has already been applied to documents loaded into the database, unless reindexing is enabled for the database.

27. Namespaces

This section describes how to use the Admin Interface to manage namespaces. For details on how to manage namespaces programmatically, see [Group Maintenance Operations](#) in the *Scripting Administrative Tasks Guide*.

Namespaces are a powerful mechanism used to differentiate between potentially ambiguous XML elements. Namespaces can be defined within individual XQuery programs. They can also be defined using the Admin Interface.

Namespaces can be defined for a group to apply to all HTTP, ODBC, XDBC, and WebDAV servers in a group or for a particular HTTP, ODBC, XDBC, or WebDAV server. However, a namespace cannot be defined to apply to a particular forest, database, or XQuery program.

For more information about namespaces, see the [Understanding XML Namespaces in XQuery](#) in the *XQuery and XSLT Reference Guide*, which provides a detailed description of XML namespaces and their use. Be sure to review this information before using the Admin Interface to manage your namespaces.

27.1. Defining Namespaces for a Group

To define namespaces for a group, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.
2. Click your target group.
3. Under the target group, click **Namespaces**.
4. Click the **Add** tab.
5. Enter a prefix for your namespace into the **Prefix** field.
6. Enter a URI for your namespace into the **Namespace Uri** field.
If you are defining a prefix for the universal unnamed namespace, leave the **Namespace Uri** field blank.
7. To add more namespace definitions, click **More Items** and repeat [Step 5 - Step 6](#) for each namespace as needed.
8. Scroll to the top or bottom and click **OK**.

The namespace is now defined in the group.

27.2. Defining Namespaces for an HTTP, ODBC, or XDBC Server

To define namespaces using the Admin Interface for an HTTP, ODBC, or XDBC Server, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.
2. Click your target group.
3. Click **App Servers**.
4. Under **App Servers**, click on the name of the App server for which you want to define the namespace.
5. Under the App Server, click **Namespaces**.
6. Click the **Add** tab
7. Enter a prefix for your namespace into the **Prefix** field.
8. Enter a URI for your namespace into the **Namespace Uri** field.
If you are defining a prefix for the universal unnamed namespace, leave the **Namespace Uri** field blank.
9. To add more namespace definitions, click **More Items** and repeat [Step 7 - Step 8](#) for each namespace as needed.

10. Scroll to the top or bottom and click **OK**.

The namespace is now defined for the App Server.

27.3. Viewing Namespace Settings for a Group

To view namespaces you have defined in the Admin Interface, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.
2. Click your target group.
3. Under the target group, click **Namespaces**.
4. Click **Namespaces** on the left tree menu, under the specified group.

27.4. Viewing Namespace Settings for an HTTP, ODBC, or XDBC Server

To view namespaces you have defined in the Admin Interface, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.
2. Click your target group.
3. Under the target group, click **App Servers**.
4. Click your target App Server.
5. Under the target App Server, click **Namespaces**
6. Under the App Server, click **Namespaces**

27.5. Deleting Namespaces for a Group

To delete namespaces that you defined in the Admin Interface, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.
2. Click your target group.
3. Under the target group, click **Namespaces**.
4. Locate the namespace to be deleted and click **Delete**. A confirmation message appears.
5. Confirm the delete and click **OK**.

The namespace is deleted from the group.

27.6. Deleting Namespaces for an HTTP, ODBC, or XDBC Server

To delete namespaces that you defined in the Admin Interface for an HTTP, ODBC, or XDBC server, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.
2. Click your target group.
3. Under the target group, click **App Servers**.
4. Click your target App Server.
5. Under the target App Server, click **Namespaces**
6. Locate the namespace to be deleted and click **Delete**. A confirmation message appears.
7. Confirm the delete and click **OK**.

The namespace is deleted from the App Server.

28. Understanding and Defining Schemas

This section describes schemas and lists procedures for defining them. For more information on the Schema database, loading schemas into MarkLogic Server, and using schemas in your applications, see [Loading Schemas](#) in the *Application Developer's Guide*.

28.1. Understanding Schemas

A schema is a data dictionary for your XML content. To specify a schema, you need to define the namespace to which the schema applies as well as the location of the schema file.

Schemas define the types of elements within XML documents. When knowing the type of an XML element would be beneficial to evaluating an XQuery program, MarkLogic Server will look for the relevant schema document (based on that element's namespace) using this strategy:

1. If the XQuery program explicitly references a schema for the namespace in question, MarkLogic Server uses this reference.
2. Otherwise, MarkLogic Server searches the schema database for an XML schema document whose target namespace is the same as the namespace of the element that MarkLogic Server is trying to type.
3. If no matching schema document is found in the database, MarkLogic Server looks in its `Config` directory for a matching schema document.
4. If no matching schema document is found in the `Config` directory, no schema is found.

Problems can arise in [Step 2](#) above when there are multiple schema documents in the schema database whose target namespace matches the namespace of the element that MarkLogic Server is trying to type. In this case, it is convenient to be able to use the Admin Interface to specify a default mapping.

Schema mappings can be specified for the HTTP, ODBC, or XDBC servers individually or for the group to apply to all HTTP, ODBC, or XDBC servers in the group. If the schema mapping defined for an HTTP, ODBC, or XDBC server conflicts with the schema mapping defined for the group, the former mapping is used.

When you specify a schema mapping in the Admin Interface, MarkLogic Server uses this strategy to locate the schema:

1. First, MarkLogic Server searches the schema database for a document with the exact URI you specified in the schema mapping.



NOTE

If the schema mapping for the HTTP, ODBC, or XDBC server conflicts with the schema mapping for the group, the former mapping is used.

2. If no matching schema document is found in the schema database, MarkLogic Server looks in its `Config` directory for a schema document whose filename matches the filename portion of the URI you specified.
3. If no matching schema document is found in the `Config` directory, no schema is found.

If a namespace is invoked by one or more data elements stored in a particular database, and the schema for that namespace is defined for the group or HTTP, ODBC, or XDBC server, MarkLogic Server applies the schema to the storage, indexing, and retrieval of that data.

**NOTE**

The schema database in this case is the schema database for the database in which the data is located.

28.2. Procedures For Defining Schemas

The procedures described in this section define schemas.

28.2.1. Adding a Schema Definition for a Group

To make a schema available to all HTTP, ODBC, or XDBC servers in a group, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.
2. Click your target group.
3. Under the target group, click **Schemas**.
4. Click the **Add** tab. The **Add Schemas** page appears:
5. Enter a namespace URI into the **Namespace URI** field and the corresponding schema location into the **Schema Location** field.

If you are planning to store the schema in your `Config` directory, the following table lists the default location of the `Config` directory on each platform:

Platform	Schema Directory
Microsoft Windows	C:\Program Files\MarkLogic\Config
Red Hat Linux	/opt/MarkLogic/Config
Mac OS X	~/Library/MarkLogic/Config/

6. To add more schema definitions, click **More Items** and repeat [Step 5](#).
7. Scroll to the top or bottom and click **OK**.

The schema is added to the group.

28.2.2. Adding a Schema Definition for an HTTP, ODBC, or XDBC Server

To make a schema available to a particular HTTP, ODBC, or XDBC server, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.
2. Click your target group.
3. Under the target group, click **App Servers**.
4. Click your target App Server.
5. Under the target App Server, click **Schemas**
6. Click the **Add** tab. The **Add Schemas** page appears:
7. Enter a namespace URI into the **Namespace Uri** field and the corresponding schema location into the **Schema Location** field.

If you are planning to store the schema in your config directory, refer to the following table for the default location of the config directory on your platform:

Platform	Schema Directory
Microsoft Windows	C:\Program Files\MarkLogic\Config
Red Hat Linux	/opt/MarkLogic/Config
Mac OS X	~/Library/MarkLogic/Config/

8. To add more schema definitions, click **More Items** and repeat [Step 7](#) for other schemas as needed.
9. Scroll to the top or bottom and click **OK**.

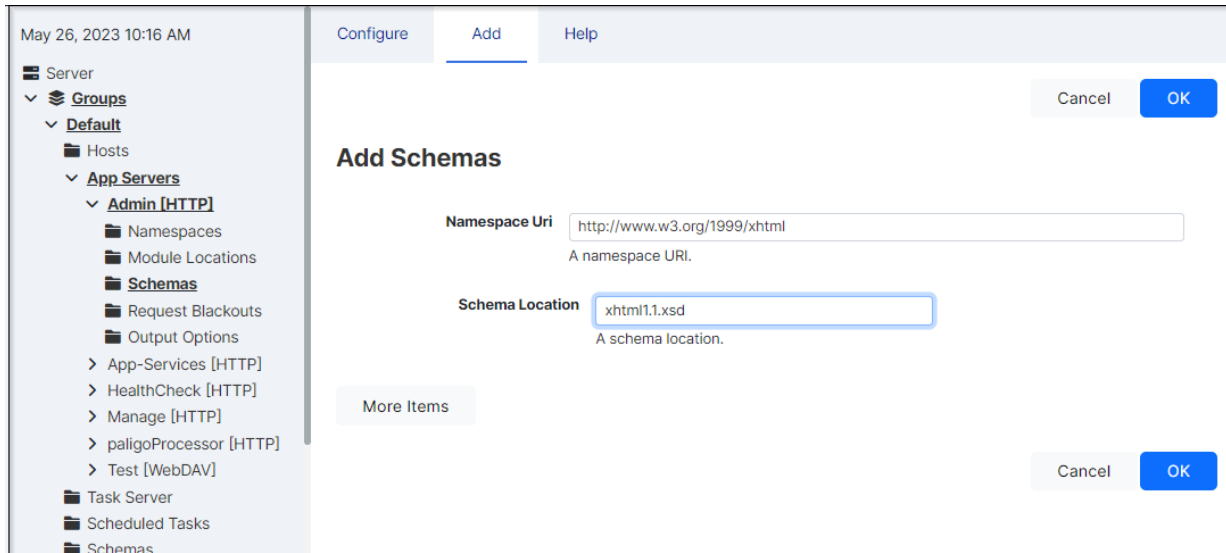
The schema is added to the HTTP, ODBC, or XDBC server.

28.2.3. Viewing Schema Definitions for a Group

To view a schema definition for a group, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.
2. Click your target group.
3. Under the target group, click **Schemas**.

The following example shows just one schema. It specifies that the schema for namespace `http://www.w3.org/1999/xhtml` is found in the file `xhtml.1.xsd`, which is located in the config directory of your MarkLogic Server program directory.

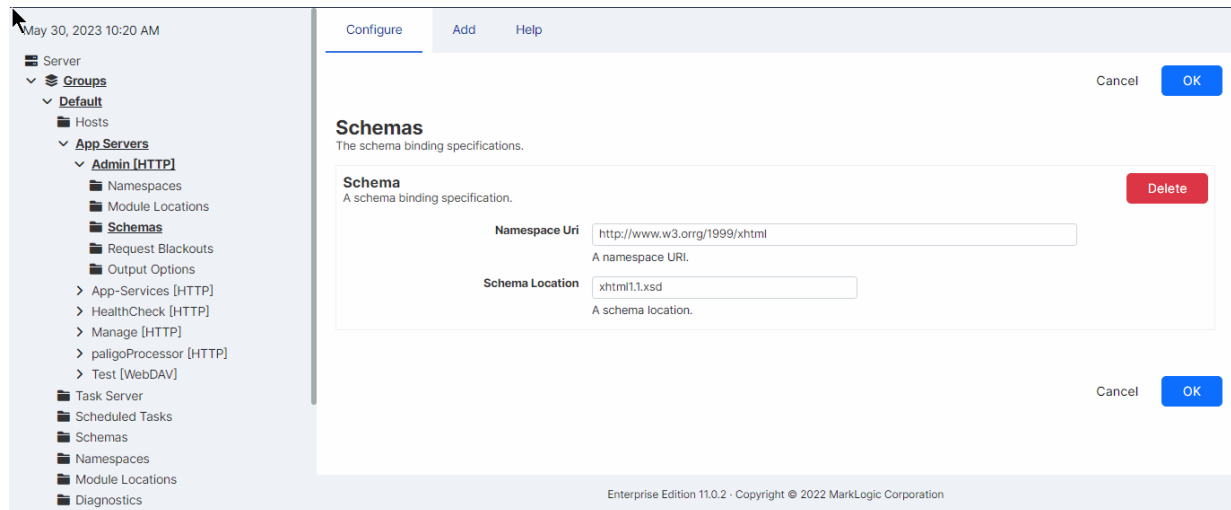


28.2.4. Viewing Schema Definitions for an HTTP, ODBC, or XDBC Server

To view a schema definition for an HTTP or XDBC Server, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.
2. Click your target group.
3. Under the target group, click **App Servers**.
4. Click your target App Server.
5. Under the target App Server, click **Schemas**

The following example shows just one schema. It specifies that the schema for namespace **`http://www.w3.org/1999/xhtml`** is found in the file **`xhtml.1.xsd`**, which is located in the config directory of your MarkLogic Server program directory.



28.2.5. Deleting a Schema Definition for a Group

To delete a schema definition for a group, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.
2. Click your target group.
3. Under the target group, click **Schemas**.
4. Locate the schema definition to be deleted from the system and click **Delete**. A confirmation message appears.
5. Confirm the delete and click **OK**.

The schema is deleted from the group.

28.2.6. Deleting a Schema Definition for an HTTP, ODBC, or XDBC Server

To delete a schema definition for an HTTP, ODBC, or XDBC server, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.
2. Click your target group.
3. Under the target group, click **App Servers**.
4. Click your target App Server.
5. Under the target App Server, click **Schemas**
6. Locate the schema definition to be deleted from the system and click Delete. A confirmation message appears.
7. Confirm the delete and click **OK**.

The schema is deleted from the App Server.

29. Log Files

This section describes the log files. For information on the audit log files, see [Section 10, “Auditing Events” \[71\]](#).

29.1. Application and System Log Files

There are separate log files for application-generated messages and for system-generated messages. This allows for separation of personally identifiable information (such as social security numbers, for example) and system messages (such as merge notices and other system activity). The application log files are configured on a per-App Server basis, and the system log files are configured at the group level. Each host has its own set of log files (both application and system log files). Things like uncaught application errors, which might contain data from an application, are sent to the application logs. Things like MarkLogic Server system activity are sent to the system log files.

29.2. Understanding the Log Levels

MarkLogic Server sends log messages to both the operating system log and the MarkLogic Server system file log. Additionally, application log messages (messages generated from application code) are sent to the application logs. Depending on how you configure your logging functions, both operating system and file logs may or may not receive the equivalent number of messages. To enhance performance, the system log should receive fewer messages than the MarkLogic Server file log.

MarkLogic Server uses the following log settings, where Finest is the most verbose while Emergency is the least verbose:

Log Level	Description
Finest	Extremely detailed debug level messages.
Finer	Very detailed debug level messages.
Fine	Detailed debug level messages.
Debug	Debug level messages.
Config	Configuration messages.
Info	Informational messages. This is the default setting.
Notice	Normal but significant conditions.
Warning	Warning conditions.
Error	Error conditions.
Critical	Critical conditions.
Alert	Immediate action required.
Emergency	System is unusable.

Log file settings are applied on a per-group basis.

By default, the system log for a group is set to Notice while the file log is set to Info. As such, the system log receives fewer log messages than the file log. You may change these settings to suit your needs. It is more efficient to write to the file log than to the system log. It is good practice to run in production with the Debug file log level to get a more detailed record of operations. Log levels more verbose than Debug may result in very large log files and are not recommended for extended periods of time.

29.3. Configuring System Log Files

To configure how log information is generated, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.

2. Click your target group.
3. Scroll to the **System Log Level** field and change the value if needed.
4. If needed, change the **File Log Level** field to change the logging level of the MarkLogic Server private log file (`ErrorLog.txt`).
5. In the **Rotate Log Files** field, select when MarkLogic Server should start a new private log file for this group.

The following table describes each time frame:

Time Frame	Description
Never	The log file grows without bound.
Daily	A new log file is started every day at 12:00 A.M.
Sunday	A new log file is started every week on Sunday at 12:00 A.M.
Saturday	A new log file is started every week on Saturday at 12:00 A.M.
Friday	A new log file is started every week on Friday at 12:00 A.M.
Thursday	A new log file is started every week on Thursday at 12:00 A.M.
Wednesday	A new log file is started every week on Wednesday at 12:00 A.M.
Tuesday	A new log file is started every week on Tuesday at 12:00 A.M.
Monday	A new log file is started every week on Monday at 12:00 A.M.
Monthly	A new log file is started at 12:00 AM on the first day of each month.

6. In the **Keep Log Files** field, enter the number of private log files to keep. The private log files are kept in an aging archive. After the number of log files grows to the value specified in the **Keep Log Files** setting, when a new log file is started, the oldest log file archive is automatically deleted.
7. Scroll to the top or bottom and click **OK**.

29.4. Configuring Application Log Files

To configure how log information is generated for an App Server, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.
2. Click your target group.
3. Under the target group, click **App Servers**.
4. Click your target App Server.
5. Scroll down to the **File Log Level** field and change the logging level of the application log file (for example, `8543_ErrorLog.txt` for the App Server on port 8543) if needed.
6. In the **Log Errors** field, click **true** if you want uncaught application errors to go to the log file, otherwise click **false**.
7. Scroll to the top or bottom and click **OK**.



NOTE

The log rotation of application log files follows the same rules as the system log file for that group, as described in the procedure for [Section 29.3, “Configuring System Log Files” \[259\]](#).

29.5. Viewing the System Log

The system log messages that MarkLogic Server generates are viewable using the standard system log viewing tools available for your platform. On Windows platforms, the seven levels of logging messages are collapsed into three broad categories and the system log messages are registered as `MarkLogic`. On UNIX platforms, the system logs use the `LOG_DAEMON` facility, which typically sends

system log messages to a file such as `/var/log/messages`, although this can vary according to the configuration of your system.

29.6. Viewing the Application and System File Logs

The private system file log is maintained as a simple text file, and the application logs are also maintained as simple text files. You may view the current or any archived file log at any time using standard text file viewing tools. Additionally, you can access the log files from the Log tab on the main page of the Admin Interface.

The files are stored in the `Logs` directory under the MarkLogic Server data directory for your platform. You may have overridden the default location for this directory at installation time. The following table lists the default location of the file logs on your platform:

Platform	Private Log Files
Microsoft Windows	<code>C:\Program Files\MarkLogic\Data\Logs\ErrorLog.txt</code> <code>C:\Program Files\MarkLogic\Data\Logs<port>_ErrorLog.txt</code>
Red Hat Enterprise Linux	<code>/var/opt/MarkLogic/Logs/ErrorLog.txt</code> <code>/var/opt/MarkLogic/Logs/<port>_ErrorLog.txt</code>
Mac OS X	<code>~/Library/Application Support/MarkLogic/Data/Logs/ErrorLog.txt</code> <code>~/Library/Application Support/MarkLogic/Data/Logs/<port>_ErrorLog.txt</code>

The application log files are prefixed with the port number of the App Server corresponding the log file. These files contain a set of log messages ordered chronologically. The number of messages depends on the system activity and on the log level that you set. For example, a file log set to Debug would contain many lines of messages whereas a file log set to Emergency would contain the minimum set of messages.

Any trace events are also written to the MarkLogic Server `ErrorLog.txt` file. Trace events are used to debug applications. You can enable and set trace events in the Admin Interface on the Diagnostics page for a group. You can also generate your own trace events with the `xdmp:trace` function, which will go to the application log.



NOTE

When the log files are being written to a different device than the data files, it is possible to run out of space for logging. This does not appear to impact the servers ability to start, nor the availability of the forests, databases or app servers. Log entries will also still be written to `/var/log/messages` based on the configured OS logging level.

29.7. Viewing Access Log Files

MarkLogic Server also produces access log files for each App Server. The access logs are in the NCSA combined log format, and show the requests made against each App Server. The access log files are in the same directory as the `ErrorLog.txt` logs, and have the port number encoded into their name. For example, the access log files for the Admin Interface is named `8001_AccessLog.txt`. You may view the current or any archived file log at any time using standard text file viewing tools. Additionally, you can access the log files from the Log tab on the main page of the Admin Interface. Older versions of the access logs are aged from the system according to the settings configured at the group level, as described in [Section 29.3, “Configuring System Log Files” \[259\]](#).

29.8. Viewing Crash Log Files

MarkLogic Server produces a crash log file named `CrashLog.txt` for each App Server in the event of segmentation fault or similar events. This file will contain useful information that MarkLogic support could use to further investigate the cause of failure.

After node recovery, MarkLogic will copy the contents of this file to `ErrorLog.txt`. It will then clear the contents of this crash log.

30. Scheduling Tasks

This section describes how to use the Admin Interface to manage scheduled tasks. It describes how to schedule tasks that execute XQuery main modules at a predefined date/time or interval. For details on how to manage scheduled tasks programmatically, see [Group Maintenance Operations](#) in the *Scripting Administrative Tasks Guide*.

30.1. Understanding Scheduled Tasks

MarkLogic Server allows you to schedule the execution of XQuery main modules. The ability to schedule module execution is useful for

- Loading content. For example, periodically checking for new content from an external data source, such as a web site, web service, etc.
- Synchronizing content. For example, when MarkLogic is used as a metadata repository, you might want to periodically check for changed data.
- Delivering batches of content: For example, initiate an RSS feed, hourly or daily.
- Delivering aggregated alerts, either hourly or daily.
- Delivering reports, either daily, weekly, or monthly.
- Polling for the completion of an asynchronous process, such as the creation of a PDF file.

Tasks can be scheduled to run at a particular time on a particular date, or at a specified interval. MarkLogic Server attempts to place the task on the task server's queue at the specified time, but the actual execution of the task might not start at this time. If the queue is full, the task fails and will not be re-tried until the next scheduled interval.

30.2. Scheduling a Module for Invocation

To schedule a module for invocation at a particular date/time or interval, follow these steps:

1. Click **Groups** in the left tree menu. A list of groups appears.
2. Click your target group.
3. Click **Scheduled Tasks** on the left tree menu.
4. Click the **Create** tab to bring up the **Schedule A Task** page.
5. Specify the URI for the module to invoke in the **Task Path** field. The task path must begin with a forward slash (/) and cannot contain a question mark (?), colon (:), or pound (#) character.
6. In the **Task Root** field, specify the root directory (file system) or URI root (database) that contains the module.
7. In the **Task Type** field, select one of the task types described in [Section 30.3, "Selecting a Task Type" \[264\]](#).
8. In the **Task Database** field, select the database on which to invoke the module.
9. In the **Task Modules** field, select either the file system or the database that contains the module specified in the **Task Path** field.
If **Task Modules** is set to (file system), then place the module in the directory specified by **Task Root**. The file will be located on disk with a path that is the **Task Root** combined with the **Task Path**. For example, if **Task Root** is /marklogic/modules and **Task Path** is /tasks/test.xqy, then test.xqy should be located at /marklogic/modules/tasks/test.xqy on the file system.
If **Task Modules** is set to a database, then load the module into that database under the URI root specified by **Task Root**. For example, for the configuration shown in [Step 10](#), test.xqy should be inserted into the Modules database at URI /marklogic/modules/test.xqy.
10. In the **Task User** and **Task Host** fields, specify the user with permission to invoke the task and the host computer on which the task is to be invoked. If no host is specified, then the task runs on all hosts.



NOTE

The user specified in the **Task User** field must have the privileges required to execute the functions used in the module. See [Section 32, “Appendix B: Pre-defined Execute Privileges” \[271\]](#) for the full list of execute privileges.

Example of a Scheduled Task configuration:

30.3. Selecting a Task Type

You can select one of the date/time or interval scheduling options described in this section as your task type.



NOTE

The date/time options are scheduled in terms of the local time designated by the server’s clock. This means that, in regions that recognize daylight savings time, a scheduling interval of 24 hours is not the same as a once-per-day at a particular time scheduling interval.

30.3.1. Scheduling per Minute

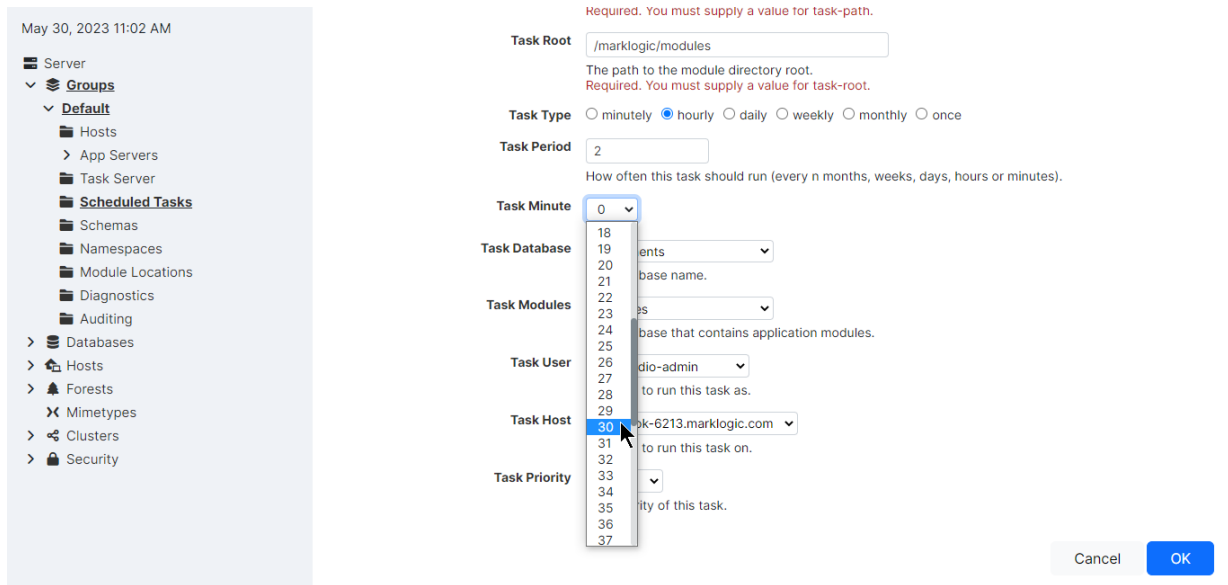
If you select minutely as the task type, in the **Task Period** field, specify how many minutes are to elapse between each invocation of the module.

30.3.2. Scheduling per Hour

If you select the hourly task type, in the **Task Period** field, specify how many hours should elapse between each invocation of the module. The **Task Minute** field specifies how many minutes after the hour the module is to be invoked. Note that the Task Minute setting does not add to the interval.

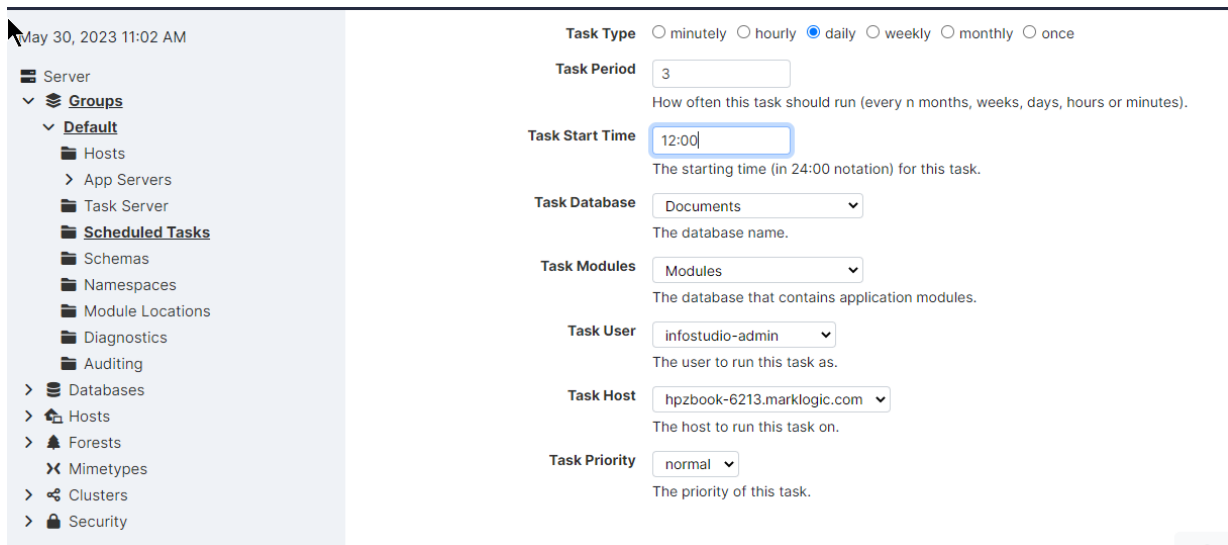
This image shows how to invoke the module every 2 hours at 30 minutes past the hour (or as soon as possible thereafter, if the server is overloaded):

If the current time is 2:15 pm, the task will run at 2:30 pm, 4:30 pm, 6:30 pm, 8:30 pm, and so on.



30.3.3. Scheduling per Day and Time

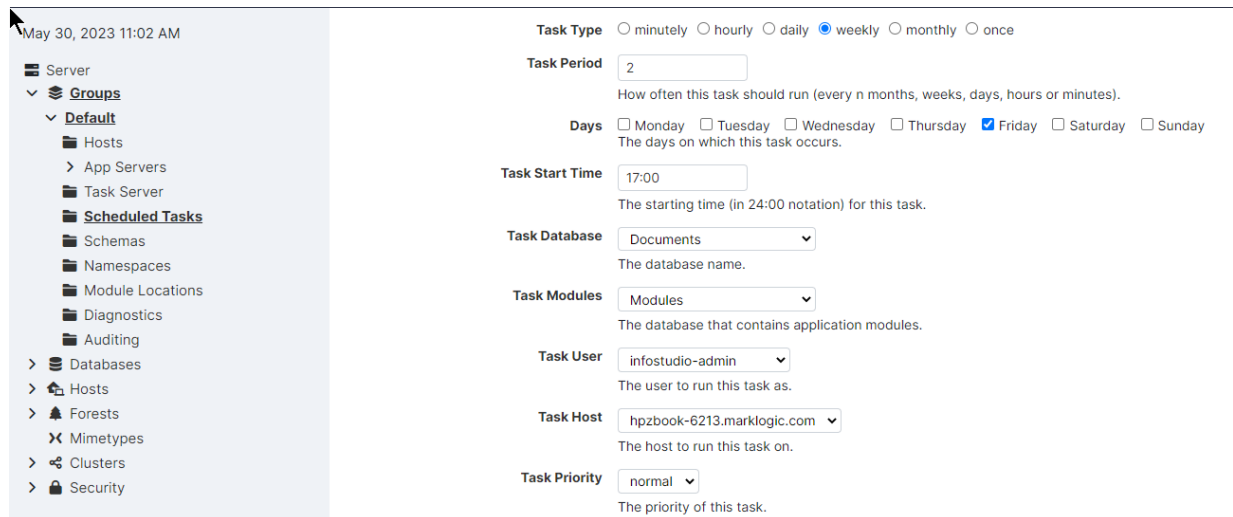
If you select the daily task type, in the **Task Period** field, specify how many days are to elapse between each invocation of the module and the time of day (in 24:00 notation) of the invocation. This image shows how to invoke the module every three days at 12:00 pm:



30.3.4. Scheduling per Week, Day, and Time

If you select the weekly task type, in the **Task Period** field, specify how many weeks are to elapse between each invocation of the module, as well as one or more days of the week and time (in 24:00 notation) of the invocation.

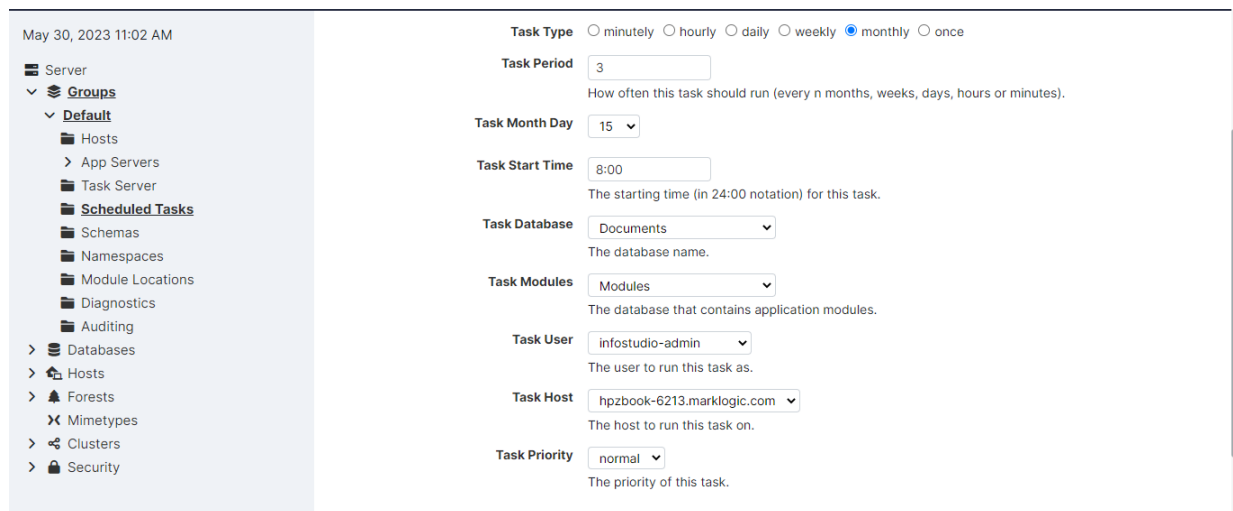
For example, to invoke the module every other week, on Friday, at 5:00 pm, enter this:



30.3.5. Scheduling per Month, Day, and Time

If you select the monthly task type, in the **Task Period** field, specify how many months are to elapse between each invocation of the module, as well as the day of the month and time (in 24:00 notation) of the invocation.

This image shows invoking the module every three months, on the 15th day of the month:



30.3.6. Scheduling Once Invocation on a Calendar Date and Time

If you select the once task type, in the **Task Period** field, specify the calendar day (month/day/year) and time (in 24:00 notation) of the invocation. This image shows how to invoke the module on May 2, 2009 at 6:00 pm:

May 30, 2023 11:02 AM

- Server
 - Groups
 - Default
 - Hosts
 - App Servers
 - Task Server
 - Scheduled Tasks**
 - Schemas
 - Namespaces
 - Module Locations
 - Diagnostics
 - Auditing
 - Databases
 - Hosts
 - Forests
 - Mimetypes
 - Clusters
 - Security

Task Type minutely hourly daily weekly monthly once

Task Start Date
The starting date (in MM/DD/YYYY notation) for this task.

Task Start Time
The starting time (in 24:00 notation) for this task.

Task Database
The database name.

Task Modules
The database that contains application modules.

Task User
The user to run this task as.

Task Host
The host to run this task on.

Task Priority
The priority of this task.

Cancel

31. Appendix A: 'Hot' versus 'Cold' Admin Tasks

"Hot" admin tasks are defined as tasks that take effect immediately and do not require the server to restart. "Cold" admin tasks are defined as tasks that require one or more instances of the server to restart to reflect the changes. Cold tasks have an asterisk (*) next to the setting in the Admin Interface.

In a clustered deployment, "cold" tasks will require one or more hosts in the cluster to restart their instance of MarkLogic in order to reflect the changes. In an single-server deployments, "cold" tasks will cause MarkLogic to restart in order to reflect the changes.

This section shows the "hot" or "cold" status for adding objects, changing configuration parameters, and dropping objects.

31.1. Groups

Add Object	Change Configuration Parameters	Delete Object
Hot	The following group parameters are hot: <ul style="list-style-type: none"> > group name > system log level > file log level > rotate log files > keep log files > namespaces > schemas The following group parameters are cold for the hosts in the group: <ul style="list-style-type: none"> > list cache size > compressed tree cache size > expanded tree cache size Adding and dropping hosts from groups is cold for that host.	Hot

31.2. HTTP, ODBC, XDBC, and WebDAV Servers

Add Object	Change Configuration Parameters	Delete Object
Hot	<p>The following App Server parameters are hot:</p> <ul style="list-style-type: none"> > server name > root > database > request timeout > keep alive timeout > session timeout > time limit > realm > security mode > namespaces > schemas > ssl certificate template > ssl hostname > ssl ciphers <p>The following App Server parameters are cold for all hosts in the group defining the HTTP, ODBC, XDBC, or WebDAV Server:</p> <ul style="list-style-type: none"> > port > address > backlog > threads > ssl enabled 	Cold

31.3. Databases

Add Object	Change Configuration Parameters	Delete Object
Hot	Parameters changes are hot	Hot

31.4. Hosts

Add Object	Change Configuration Parameters	Delete Object
Only the added host needs to restart	Only the host whose parameters change requires a restart. The rest of the hosts remain hot.	Hot for the remaining hosts.

31.5. Forests

Add Object	Change Configuration Parameters	Delete Object
Hot	Parameter changes are hot. Backup is hot. Restore, clear, and drop are hot	Hot

31.6. Mimetypes

Add Object	Change Configuration Parameters	Delete Object
Cold	Parameter changes are cold.	Cold

31.7. Security

Add Object	Change Configuration Parameters	Delete Object
Hot	Parameter changes are hot.	Hot

32. Appendix B: Pre-defined Execute Privileges

The pre-defined execute privileges listed below are included with every installation of MarkLogic Server:

Name	Action URI	Description	Protects Function
add-query-rolesets	http://marklogic.com/xdmp/privileges/add-query-rolesets	privilege to add query rolesets	sec:add-query-rolesets
admin-database	http://marklogic.com/xdmp/privileges/admin/database	privilege to administer databases	admin built-ins
admin-default-read	http://marklogic.com/xdmp/privileges/admin-default-read	internal privilege to use the Admin API for reading configuration information	admin built-ins
admin-forest	http://marklogic.com/xdmp/privileges/admin/forest	privilege to administer forests	admin built-ins
admin-host	http://marklogic.com/xdmp/privileges/admin/host	privilege to administer hosts	admin built-ins
admin-app-server	http://marklogic.com/xdmp/privileges/admin/app-server	privilege to administer app-servers	admin built-ins
admin-app-server-security	http://marklogic.com/xdmp/privileges/admin/app-server-security	privilege to administer app-servers' security	admin built-ins
admin-group	http://marklogic.com/xdmp/privileges/admin/group	privilege to administer groups	admin built-ins
admin-group-security	http://marklogic.com/xdmp/privileges/admin/group-security	privilege to administer groups' security	admin built-ins
admin-cluster	http://marklogic.com/xdmp/privileges/admin/cluster	privilege to administer clusters	admin built-ins
admin-mimetype	http://marklogic.com/xdmp/privileges/admin/mimetypes	privilege to administer mimetypes	admin built-ins
admin-module-read	http://marklogic.com/xdmp/privileges/admin-module-read	privilege to use the Admin API for reading configuration information	admin built-ins
admin-module-write	http://marklogic.com/xdmp/privileges/admin-module-write	privilege to use the Admin API for writing configuration information	admin built-ins
admin-ui	http://marklogic.com/xdmp/privileges/admin-ui	privilege to view the Admin Interface, but not to make changes	admin built-ins
amp-add-roles	http://marklogic.com/xdmp/privileges/amp-add-roles	privilege to assign additional roles to the amp	sec:amp-add-roles
amp-change-database	http://marklogic.com/xdmp/privileges/amp-change-database	privilege to assign additional roles to the amp	sec:amps-change-modules-database
amp-get-roles	http://marklogic.com/xdmp/privileges/amp-get-roles	privilege to get the roles associated with the amp	sec:amp-get-roles
amp-remove-roles	http://marklogic.com/xdmp/privileges/amp-remove-roles	privilege to remove roles assigned to the amp	sec:amp-remove-roles
amp-set-roles	http://marklogic.com/xdmp/privileges/amp-set-roles	privilege to set the roles associated with the amp	sec:amp-set-roles
any-collection	http://marklogic.com/xdmp/privileges/any-collection	privilege to add to or remove from any collection, regardless of whether it is protected	xdmp:document-add-collections, xdmp:document-remove-collections, xdmp:document-set-collections

Name	Action URI	Description	Protects Function
any-transaction-locks	http://marklogic.com/xdmp/privileges/any-transaction-locks	privilege to see URIs currently locked for read or write by a transaction.	xdmp:transaction-locks
any-uri	http://marklogic.com/xdmp/privileges/any-uri	privilege to create a document with any uri, regardless of whether the uri is protected	xdmp:document-insert, xdm:document-load, xdm:load
app-builder	http://marklogic.com/xdmp/privileges/app-builder	privilege to use the Application Builder UI App Builder is no longer part of MarkLogic	
appservices-cache-server-fields	http://marklogic.com/xdmp/privileges/appservices-cache-server-fields		
cancel-any-requests	http://marklogic.com/xdmp/privileges/cancel-any-requests	privilege to cancel requests issued by any user attempting to cancel a request	admin built-ins
cancel-my-requests	http://marklogic.com/xdmp/privileges/cancel-my-requests	privilege to cancel requests issued by the user attempting to cancel a request	admin built-ins
clang:read	http://marklogic.com/xdmp/privileges/custom-language-read	privilege to read custom language configuration specifications	clang:language-config-read
clang:write	http://marklogic.com/xdmp/privileges/custom-language-write	privilege to write custom language configuration specifications	clang:language-config-write
collection-add-permissions	http://marklogic.com/xdmp/privileges/collection-add-permissions	privilege to add permissions to a collection	sec:get-collections, sec:collection-add-permissions
collection-get-permissions	http://marklogic.com/xdmp/privileges/collection-get-permissions	privilege to get permissions on a collection	sec:collection-get-permissions
collection-remove-permissions	http://marklogic.com/xdmp/privileges/collection-remove-permissions	privilege to remove permissions from a collection	sec:get-collections, sec:collection-remove-permissions
collection-set-permissions	http://marklogic.com/xdmp/privileges/collection-set-permissions	privilege to set permissions on a collection	sec:get-collections, sec:collection-set-permissions
compartment-get-roles	http://marklogic.com/xdmp/privileges/compartment-get-roles	privilege to get roles on a compartment	sec:compartment-get-roles
complete-any-transactions	http://marklogic.com/xdmp/privileges/complete-any-transactions	privilege to use transaction built-ins for any transactions	xdmp:transaction-commit, xdm:xa-complete
complete-my-transactions	http://marklogic.com/xdmp/privileges/complete-my-transactions	privilege to use transaction built-ins for transactions started by the current user	xdmp:transaction-commit, xdm:xa-complete
count-builtins	http://marklogic.com/xdmp/privileges/counts	privilege to run xdm:forest-counts	xdmp:forest-counts
create-amp	http://marklogic.com/xdmp/privileges/create-amp	privilege to create an amp	sec:create-amp
create-credential	http://marklogic.com/xdmp/privileges/create-credential	privilege to create security credentials	sec:create-credential
create-data-role	http://marklogic.com/xdmp/privileges/create-data-role	privilege for non-admin users (with <code>manage</code> role) to create and manage roles	
create-data-user	http://marklogic.com/xdmp/privileges/create-data-user	privilege for non-admin users (with <code>manage</code> role) to create and manage users	

Name	Action URI	Description	Protects Function
create-domain	http://marklogic.com/xdmp/privileges/create-domain	privilege to create domains	dom:create
create-external-security	http://marklogic.com/xdmp/privileges/create-external-security	privilege to create an external authentication configuration	sec:create-external-security
create-pipeline	http://marklogic.com/xdmp/privileges/create-pipeline	privilege to create a pipeline	p:insert, p:create
create-privilege	http://marklogic.com/xdmp/privileges/create-privilege	privilege to create a privilege	sec:create-role
create-role	http://marklogic.com/xdmp/privileges/create-role	privilege to create a role	sec:create-role
create-trigger	http://marklogic.com/xdmp/privileges/create-trigger	privilege to create a trigger	trgr:create-trigger
create-user	http://marklogic.com/xdmp/privileges/create-user	privilege to create a user	sec:create-user
credential-get-certificate	http://marklogic.com/xdmp/privileges/credential-get-certificate	privilege to return the certificate for a credential	sec:credential-get-certificate
credential-get-description	http://marklogic.com/xdmp/privileges/credential-get-description	privilege to return the description of a credential	sec:credential-get-description
credential-get-id	http://marklogic.com/xdmp/privileges/credential-get-id	privilege to return the id of a credential	sec:credential-get-id
credential-get-password	http://marklogic.com/xdmp/privileges/credential-get-password	privilege to return the password for a credential	sec:credential-get-password
credential-get-permissions	http://marklogic.com/xdmp/privileges/credential-get-permissions	privilege to return the permissions for a credential	sec:credential-get-permissions
credential-get-private-key	http://marklogic.com/xdmp/privileges/credential-get-private-key	privilege to return the private key for a credential	sec:credential-get-private-key
credential-get-signing	http://marklogic.com/xdmp/privileges/credential-get-signing	privilege to return the signing flag for a credential	sec:credential-get-signing
credential-get-targets	http://marklogic.com/xdmp/privileges/credential-get-targets	privilege to return the targets for a credential	sec:credential-get-targets
credential-get-username	http://marklogic.com/xdmp/privileges/credential-get-username	privilege to return the user name for a credential	sec:credential-get-username
credential-set-certificate	http://marklogic.com/xdmp/privileges/credential-set-certificate	privilege to update the certificate for a credential	sec:credential-set-certificate
credential-set-description	http://marklogic.com/xdmp/privileges/credential-set-description	privilege to update the description for a credential	sec:credential-set-description
credential-set-name	http://marklogic.com/xdmp/privileges/credential-set-name	privilege to update the name for a credential	sec:credential-set-name
credential-set-password	http://marklogic.com/xdmp/privileges/credential-set-password	privilege to update the password for a credential	sec:credential-set-password
credential-set-permissions	http://marklogic.com/xdmp/privileges/credential-set-permissions	privilege to update the permissions for a credential	sec:credential-set-permissions
credential-set-signing	http://marklogic.com/xdmp/privileges/credential-set-signing	privilege to update the signing flag for a credential	sec:credential-set-signing
credential-set-targets	http://marklogic.com/xdmp/privileges/credential-set-targets	privilege to update the targets for a credential	sec:credential-set-targets

Name	Action URI	Description	Protects Function
credential-set-username	http://marklogic.com/xdmp/privileges/credential-set-username	privilege to update the user name for a credential	sec:credential-set-username
credentials-get-aws	http://marklogic.com/xdmp/privileges/credentials-get-aws	privilege to return the Amazon Web Services access key, secret key, and session token	sec:credentials-get-aws
credentials-set-aws	http://marklogic.com/xdmp/privileges/credentials-set-aws	privilege to set the Amazon Web Services access key, secret key, and session token	sec:credentials-set-aws
cts-write-dictionary	http://marklogic.com/xdmp/privileges/cts-write-dictionary		
data-role-edit-<ROLEID>	http://marklogic.com/xdmp/privileges/data-role-edit-<ROLEID>	internal privilege to edit a role created by non-admin user through create-data-role privilege	
data-role-inherit-<ROLEID>	http://marklogic.com/xdmp/privileges/data-role-inherit-<ROLEID>	internal privilege to inherit a role created by non-admin user through create-data-role privilege	
data-user-edit-<USERID>	http://marklogic.com/xdmp/privileges/data-user-edit-<USERID>	internal privilege to track and manage a user created by non-admin user through create-data-user privilege	
database-node-query-rolesets	http://marklogic.com/xdmp/privileges/database-node-query-rolesets	privilege to return a sequence of query-rolesets	xdmp:database-node-query-rolesets
debug-any-requests	http://marklogic.com/xdmp/privileges/debug-any-requests	privilege to debug all requests from any user	debug built-ins
debug-my-requests	http://marklogic.com/xdmp/privileges/debug-my-requests	privilege to debug your own requests	debug built-ins
dls-admin	http://marklogic.com/xdmp/privileges/dls-admin	privilege to configure the Library Services	dls:break-checkout, dls:retention-rule, dls:retention-rule-insert, dls:retention-rule-remove

Name	Action URI	Description	Protects Function
dls-user	http://marklogic.com/xdmp/privileges/dls-user	privilege to use the Library Services	dls:as-of-query, dls:author-query, dls:document-add-collections, dls:document-add-permissions, dls:document-add-properties, dls:document-checkin, dls:document-checkout, dls:document-checkout-status, dls:document-delete, dls:document-extract-part, dls:document-get-permissions, dls:document-history, dls:document-include-query, dls:document-insert-and-manage, dls:document-is-managed, dls:document-manage, dls:document-purge, dls:document-remove-collections, dls:document-remove-permissions, dls:document-remove-properties, dls:document-retention-rules, dls:document-set-collections, dls:document-set-permissions, dls:document-set-properties, dls:document-set-property, dls:document-set-quality, dls:document-unmanage, dls:document-update, dls:document-version, dls:document-version-as-of, dls:document-version-delete, dls:document-version-query, dls:document-version-uri, dls:document-versions-query, dls:documents-query, dls:link-expand, dls:link-references, dls:node-expand, dls:purge, dls:retention-rules
ec2-http-protected	http://marklogic.com/xdmp/privileges/ec2-http-protected		
environment-ui	http://marklogic.com/xdmp/privileges/environment-ui		
external-security-clear-cache	http://marklogic.com/xdmp/privileges/external-security-clear-cache	privilege to clear the login cache in an external authorization configuration object	sec:external-security-clear-cache
external-security-get-authentication	http://marklogic.com/xdmp/privileges/external-security-get-authentication	privilege to return the authentication protocol set in an external authorization configuration object	sec:external-security-get-authentication
external-security-get-authorization	http://marklogic.com/xdmp/privileges/external-security-get-authorization	privilege to return the authorization scheme set in an external authorization configuration object	sec:external-security-get-authorization
external-security-get-cache-timeout	http://marklogic.com/xdmp/privileges/external-security-get-cache-timeout	privilege to return the login cache timeout set in an external authorization configuration object	sec:external-security-get-cache-timeout

Name	Action URI	Description	Protects Function
external-security-get-description	http://marklogic.com/xdmp/privileges/external-security-get-description	privilege to return the description set in an external authorization configuration object	sec:external-security-get-description
external-security-get-http-option	http://marklogic.com/xdmp/privileges/external-security-get-http-option	privilege to return the http options set in an external authorization configuration object	sec:external-security-get-http-options
external-security-get-ldap-attribute	http://marklogic.com/xdmp/privileges/external-security-get-ldap-attribute	privilege to return the LDAP attribute for user lookup set in an external authorization configuration object	sec:external-security-get-ldap-attribute
external-security-get-ldap-base	http://marklogic.com/xdmp/privileges/external-security-get-ldap-base	privilege to return the LDAP base for user lookup set in an external authorization configuration object	sec:external-security-get-ldap-base
external-security-get-ldap-bind-method	http://marklogic.com/xdmp/privileges/external-security-get-ldap-bind-method	privilege to return the bind method set in an external authorization configuration object	sec:external-security-get-ldap-bind-method
external-security-get-ldap-default-user	http://marklogic.com/xdmp/privileges/external-security-get-ldap-default-user	privilege to return the default LDAP user name set in an external authorization configuration object	sec:external-security-get-ldap-default-user
external-security-get-ldap-member-attribute	http://marklogic.com/xdmp/privileges/external-security-get-ldap-member-attribute	privilege to return the member attribute set in an external authorization configuration object	sec:external-security-get-ldap-member-attribute
external-security-get-ldap-memberof-attribute	http://marklogic.com/xdmp/privileges/external-security-get-ldap-memberof-attribute	privilege to return the memberof attribute set in an external authorization configuration object	sec:external-security-get-ldap-memberof-attribute
external-security-get-ldap-server-uri	http://marklogic.com/xdmp/privileges/external-security-get-ldap-server-uri	privilege to return the LDAP server uri set in an external authorization configuration object	sec:external-security-get-ldap-server-uri
external-security-get-oauth-client-id	http://marklogic.com/xdmp/privileges/external-security-get-oauth-client-id	Gets the OAuth client-id of an external security object	sec:external-security-get-oauth-client-id
external-security-get-oauth-flow-type	http://marklogic.com/xdmp/privileges/external-security-get-oauth-flow-type	Gets the OAuth flow-type of an external security object	sec:external-security-get-oauth-flow-type
external-security-get-oauth-introspection-server-uri	http://marklogic.com/xdmp/privileges/external-security-get-oauth-introspection-server-uri	Gets the OAuth introspection server URI of an external security object	sec:external-security-get-oauth-introspection-server-uri
external-security-get-oauth-jwks-uri [v11.2.0 and up]	http://marklogic.com/xdmp/privileges/external-security-get-oauth-jwks-uri	Gets the OAuth JWKS URI of an external security object	sec:external-security-get-oauth-jwks-uri
external-security-get-oauth-jwt-alg [v11.2.0 and up]	http://marklogic.com/xdmp/privileges/external-security-get-oauth-jwt-alg	Gets the OAuth JWT algorithm of an external security object	sec:external-security-get-oauth-jwt-alg
external-security-get-oauth-jwt-issuer-uri [v11.2.0 and up]	http://marklogic.com/xdmp/privileges/external-security-get-oauth-jwt-issuer-uri	Gets the OAuth JWT issuer URI of an external security object	sec:external-security-get-oauth-jwt-issuer-uri
external-security-get-oauth-jwt-secrets [v11.2.0 and up]	http://marklogic.com/xdmp/privileges/external-security-get-oauth-jwt-secrets	Gets the OAuth JWT secrets of an external security object	sec:external-security-get-oauth-jwt-secrets
external-security-get-oauth-privilege-attribute	http://marklogic.com/xdmp/privileges/external-security-get-oauth-privilege-attribute	Gets the OAuth privilege-attribute of an external security object	sec:external-security-get-oauth-privilege-attribute

Name	Action URI	Description	Protects Function
external-security-get-oauth-role-attribute	http://marklogic.com/xdmp/privileges/external-security-get-oauth-role-attribute	Gets the OAuth role-attribute of an external security object	sec:external-security-get-oauth-role-attribute
external-security-get-oauth-token-type	http://marklogic.com/xdmp/privileges/external-security-get-oauth-token-type	Gets the OAuth token-type of an external security object	sec:external-security-get-oauth-token-type
external-security-get-oauth-username-attribute	http://marklogic.com/xdmp/privileges/external-security-get-oauth-username-attribute	Gets the OAuth username-attribute of an external security object	sec:external-security-get-oauth-username-attribute
external-security-get-oauth-vendor	http://marklogic.com/xdmp/privileges/external-security-get-oauth-vendor	Gets the OAuth vendor of an external security object	sec:external-security-get-oauth-vendor
external-security-get-saml-attribute-names	http://marklogic.com/xdmp/privileges/external-security-get-saml-attribute-names	privilege to return the SAML attribute names set in an external authorization configuration object	sec:external-security-get-saml-attribute-names
external-security-get-saml-entity-id	http://marklogic.com/xdmp/privileges/external-security-get-saml-entity-id	privilege to return the SAML entity id set in an external authorization configuration object	sec:external-security-get-saml-entity-id
external-security-get-saml-privilege-attribute-name	http://marklogic.com/xdmp/privileges/external-security-get-saml-privilege-attribute-name	privilege to return the SAML privilege attribute name set in an external authorization configuration object	sec:external-security-get-saml-privilege-attribute-name
external-security-get-ssl-client-certificate-authorities	http://marklogic.com/xdmp/privileges/external-security-get-ssl-client-certificate-authorities	privilege to return the external security's SSL client certificate authorities set in an external authorization configuration object	sec:external-security-get-ssl-client-certificate-authorities
external-security-get-ssl-require-client-certificate	http://marklogic.com/xdmp/privileges/external-security-get-ssl-require-client-certificate	privilege to return the external security's SSL require client certificate flag set in an external authorization configuration object	sec:external-security-get-ssl-require-client-certificate
external-security-set-authentication	http://marklogic.com/xdmp/privileges/external-security-set-authentication	privilege to set the authentication protocol in an external authorization configuration object	sec:external-security-set-authentication
external-security-set-authorization	authorization http://marklogic.com/xdmp/privileges/external-security-set-authorization	privilege to set the authorization scheme in an external authorization configuration object	sec:external-security-set-authorization
external-security-set-cache-timeout	http://marklogic.com/xdmp/privileges/external-security-set-cache-timeout	privilege to set the login cache timeout in an external authorization configuration object	sec:external-security-set-cache-timeout
external-security-set-description	http://marklogic.com/xdmp/privileges/external-security-set-description	privilege to set the description in an external authorization configuration object	sec:external-security-set-description
external-security-set-http-options	http://marklogic.com/xdmp/privileges/external-security-set-http-options	privilege to set the http options in an external authorization configuration object	sec:external-security-set-http-options
external-security-set-ldap-attribute	http://marklogic.com/xdmp/privileges/external-security-set-ldap-attribute	privilege to set the LDAP attribute for user lookup in an external authorization configuration object	sec:external-security-set-ldap-attribute
external-security-set-ldap-base	http://marklogic.com/xdmp/privileges/external-security-set-ldap-base	privilege to set the LDAP base for user lookup in an external authorization configuration object	sec:external-security-set-ldap-base

Name	Action URI	Description	Protects Function
external-security-set-ldap-bind-method	http://marklogic.com/xdmp/privileges/external-security-set-ldap-bind-method	privilege to set the bind method in an external authorization configuration object	sec:external-security-set-ldap-bind-method
external-security-set-ldap-default-user	http://marklogic.com/xdmp/privileges/external-security-set-ldap-default-user	privilege to set the default user name in an external authorization configuration object	sec:external-security-set-ldap-default-user
external-security-set-ldap-member-attribute	http://marklogic.com/xdmp/privileges/external-security-set-ldap-member-attribute	privilege to set the member LDAP attribute in an external authorization configuration object	sec:external-security-set-ldap-member-attribute
external-security-set-ldap-memberof-attribute	http://marklogic.com/xdmp/privileges/external-security-set-ldap-memberof-attribute	privilege to set the memberof LDAP attribute in an external authorization configuration object	sec:external-security-set-ldap-memberof-attribute
external-security-set-ldap-password	http://marklogic.com/xdmp/privileges/external-security-set-ldap-password	privilege to set the default user password in an external authorization configuration object	sec:external-security-set-ldap-password
external-security-set-ldap-server-uri	http://marklogic.com/xdmp/privileges/external-security-set-ldap-server-uri	privilege to set the LDAP server uri in an external authorization configuration object	sec:external-security-set-ldap-server-uri
external-security-set-name	http://marklogic.com/xdmp/privileges/external-security-set-name	privilege to set the name of an external authorization configuration object	sec:external-security-set-name
external-security-set-oauth-client-id	http://marklogic.com/xdmp/privileges/external-security-set-oauth-client-id	Sets the OAuth client-id of an external security object	sec:external-security-set-oauth-client-id
external-security-set-oauth-flow-type	http://marklogic.com/xdmp/privileges/external-security-set-oauth-flow-type	Sets the OAuth flow-type of an external security object	sec:external-security-set-oauth-flow-type
external-security-set-oauth-introspection-server-uri	http://marklogic.com/xdmp/privileges/external-security-set-oauth-introspection-server-uri	Sets the OAuth introspection server URI of an external security object	sec:external-security-set-oauth-introspection-server-uri
external-security-set-oauth-jwks-uri [v11.2.0 and up]	http://marklogic.com/xdmp/privileges/external-security-set-oauth-jwks-uri	Sets the OAuth JWKS URI of an external security object	sec:external-security-set-oauth-jwks-uri
external-security-set-oauth-jwt-alg [v11.2.0 and up]	http://marklogic.com/xdmp/privileges/external-security-set-oauth-jwt-alg	Sets the OAuth JWT algorithm of an external security object	sec:external-security-set-oauth-jwt-alg
external-security-set-oauth-jwt-issuer-uri [v11.2.0 and up]	http://marklogic.com/xdmp/privileges/external-security-set-oauth-jwt-issuer-uri	Sets the OAuth JWT issuer URI of an external security object	sec:external-security-set-oauth-jwt-issuer-uri
external-security-set-oauth-jwt-secrets [v11.2.0 and up]	http://marklogic.com/xdmp/privileges/external-security-set-oauth-jwt-secrets	Sets the OAuth JWT secrets of an external security object	sec:external-security-set-oauth-jwt-secrets
external-security-set-oauth-privilege-attribute	http://marklogic.com/xdmp/privileges/external-security-set-oauth-privilege-attribute	Sets the OAuth privilege-attribute of an external security object	sec:external-security-set-oauth-privilege-attribute
external-security-set-oauth-role-attribute	http://marklogic.com/xdmp/privileges/external-security-set-oauth-role-attribute	Sets the OAuth role-attribute of an external security object	sec:external-security-set-oauth-role-attribute
external-security-set-oauth-token-type	http://marklogic.com/xdmp/privileges/external-security-set-oauth-token-type	Sets the OAuth token-type of an external security object	sec:external-security-set-oauth-token-type

Name	Action URI	Description	Protects Function
external-security-set-oauth-username-attribute	http://marklogic.com/xdmp/privileges/external-security-set-oauth-username-attribute	Sets the OAuth username-attribute of an external security object	sec:external-security-set-oauth-username-attribute
external-security-set-oauth-vendor	http://marklogic.com/xdmp/privileges/external-security-set-oauth-vendor	Sets the OAuth vendor of an external security object	sec:external-security-set-oauth-vendor
external-security-set-saml-attribute-names	http://marklogic.com/xdmp/privileges/external-security-set-saml-attribute-names	privilege to set SAML attribute names used by other security objects to identify a SAML configuration	sec:external-security-set-saml-attribute-names
external-security-set-saml-entity-id	http://marklogic.com/xdmp/privileges/external-security-set-saml-entity-id	privilege to set the SAML entity ID used by other security objects to identify a SAML configuration	sec:external-security-set-saml-entity-id
external-security-set-saml-privilege-attribute-name	http://marklogic.com/xdmp/privileges/external-security-set-saml-privilege-attribute-name	privilege to set the SAML privilege attribute name in a SAML configuration	sec:external-security-set-saml-privilege-attribute-name
external-security-set-ssl-client-certificate-authorities	http://marklogic.com/xdmp/privileges/external-security-set-ssl-client-certificate-authorities	privilege to set the SSL client certificate authorities in an external authorization configuration object	sec:external-security-set-ssl-client-certificate-authorities
external-security-set-ssl-require-client-certificate	http://marklogic.com/xdmp/privileges/external-security-set-ssl-require-client-certificate	privilege to set the SSL require client certificate flag in an external authorization configuration object	sec:external-security-set-ssl-require-client-certificate
flexrep-admin	http://marklogic.com/xdmp/privileges/flexrep-admin	privilege to administer flexible replication	flexible replication functions
flexrep-internal	http://marklogic.com/xdmp/privileges/flexrep-internal	used for amping flexible replication functions	flexible-internal
flexrep-user	http://marklogic.com/xdmp/privileges/flexrep-user	privilege to use flexible replication	flexible replication user functions
forget-any-xa-transactions	http://marklogic.com/xdmp/privileges/forget-any-xa-transactions	privilege to run built-in to forget XA transactions for any transactions	xdmp:xa-forget, xdmp:xq-forget-xid
forget-my-xa-transactions	http://marklogic.com/xdmp/privileges/forget-my-xa-transactions	privilege to run built-in to forget XA transactions for the user's transactions	xdmp:xa-forget, xdmp:xq-forget-xid
get-amp	http://marklogic.com/xdmp/privileges/get-amp	privilege to get an amp	sec:get-amp
get-an-admin-user-id	http://marklogic.com/xdmp/privileges/get-an-admin-user-id	privilege to get an admin user id	
get-appserver-logs	http://marklogic.com/xdmp/privileges/logs/appserver	privilege to get App Server logs	
get-compartments	http://marklogic.com/xdmp/privileges/get-compartments	privilege to get a the compartments	sec:get-compartments
get-credential	http://marklogic.com/xdmp/privileges/get-credential	privilege to get a PEM encoded X509 certificate	sec:get-credential
get-credential-by-id	http://marklogic.com/xdmp/privileges/get-credential-by-id	privilege to get a PEM encoded X509 certificate	
get-credential-ids	http://marklogic.com/xdmp/privileges/get-credential-ids	privilege to get all of the credential IDs in the security database	
get-credential-names	http://marklogic.com/xdmp/privileges/get-credential-names	privilege to get all of the credential names in the security database	
get-credentials-encoded-kek	http://marklogic.com/xdmp/privileges/get-credentials-encoded-kek		

Name	Action URI	Description	Protects Function
get-logs	http://marklogic.com/xdmp/privileges/logs	privilege to get logs	
get-privilege	http://marklogic.com/xdmp/privileges/get-privilege	privilege to get a privilege from action uri and type	sec:get-privilege
get-role-ids	http://marklogic.com/xdmp/privileges/get-role-ids	privilege to get role ids	internal functions
get-role-names	http://marklogic.com/xdmp/privileges/get-role-names	privilege to get role names	internal functions
get-saml-entity-ids	http://marklogic.com/xdmp/privileges/get-saml-entity-ids	privilege to get the SAML entity ids stored in the Security database	sec:get-saml-entity-ids
get-system-logs	http://marklogic.com/xdmp/privileges/logs/system	privilege to get system logs	
get-taskserver-logs	http://marklogic.com/xdmp/privileges/logs/taskserver	privilege to get taskserver logs	
get-user-names	http://marklogic.com/xdmp/privileges/get-user-names	privilege to get user names	sec:get-user-names
grant-all-roles	http://marklogic.com/xdmp/privileges/grant-all-roles	privilege to grant a user all roles. Either grant-all-roles or grant-my-roles would be needed by functions that assign roles.	sec:create-user, sec:user-set-roles, sec:user-add-roles, sec:user-remove-roles, sec:create-role, sec:role-set-roles, sec:role-add-roles, sec:role-remove-roles, sec:remove-role-from-roles, sec:remove-role-from-privileges, sec:remove-role-from-amps, sec:create-role, sec:privilege-set-roles, sec:privilege-add-roles, sec:privilege-remove-roles, sec:create-amp, sec:amp-set-roles, sec:amp-add-roles, sec:amp-remove-roles
grant-my-roles	http://marklogic.com/xdmp/privileges/grant-my-roles	privilege to grant a user my roles. Either grant-all-roles or grant-my-roles would be needed by functions that assign roles.	sec:create-user, sec:user-set-roles, sec:user-add-roles, sec:user-remove-roles, sec:create-role, sec:role-set-roles, sec:role-add-roles, sec:role-remove-roles, sec:remove-role-from-roles, sec:remove-role-from-privileges, sec:remove-role-from-amps, sec:create-role, sec:privilege-set-roles, sec:privilege-add-roles, sec:privilege-remove-roles, sec:create-amp, sec:amp-set-roles, sec:amp-add-roles, sec:amp-remove-roles
hadoop-user-read	http://marklogic.com/xdmp/privileges/hadoop-user-read	privilege to use MarkLogic Server as an input for a Hadoop MapReduce job that reads data from MarkLogic Server.	Java APIs in the Hadoop package.
hadoop-user-write	http://marklogic.com/xdmp/privileges/hadoop-user-write	privilege to use MarkLogic Server as an input for a Hadoop MapReduce job that writes data from MarkLogic Server	Java APIs in the Hadoop package.
healthcheck	http://marklogic.com/xdmp/privileges/healthcheck	privilege to use the HealthCheck App Server	

Name	Action URI	Description	Protects Function
infostudio	http://marklogic.com/xdmp/privileges/infostudio	privilege to use Information Studio Information Studio is no longer part of MarkLogic	Information Studio functions
java	http://marklogic.com/xdmp/privileges/java		
manage	http://marklogic.com/xdmp/privileges/manage	privilege to run the Management API	For example, non-admin users can use manage role plus create-data-role/ create-data-user granular privilege to manage roles/ users created by data users or granted. package:add-database, package:add-appserver, All of the resource addresses in the Management API
manage-admin	http://marklogic.com/xdmp/privileges/manage-admin	privilege to use the manage REST APIs	
my-transaction-locks	http://marklogic.com/xdmp/privileges/my-transaction-locks	privilege to return URIs currently locked for read or write by a transaction	xdmp:transaction-locks
native-plugin	http://marklogic.com/xdmp/privileges/native-plugin		
node-query-rolesets	http://marklogic.com/xdmp/privileges/node-query-rolesets	privilege to return query-rolesets	xdmp:node-query-rolesets
odbc:eval	http://marklogic.com/xdmp/privileges/odbc-eval	privilege to execute eval statements from odbc	xdmp:eval
odbc:eval-in	http://marklogic.com/xdmp/privileges/odbc-eval-in	privilege to execute eval-in statements from odbc	xdmp:eval-in
odbc:eval-modules-change	http://marklogic.com/xdmp/privileges/odbc-eval-modules-change	privilege to execute eval statements that change a modules database from odbc	xdmp:eval
odbc:eval-modules-change-file	http://marklogic.com/xdmp/privileges/odbc-eval-modules-change-file	privilege to execute eval statements that change a filesystem root from odbc	xdmp:eval
odbc:insert	http://marklogic.com/xdmp/privileges/odbc-insert	privilege to execute insert statements from odbc	odbc inserts
odbc:insert-in	http://marklogic.com/xdmp/privileges/odbc-insert-in	privilege to execute insert statements from odbc	odbc inserts into another database
odbc:invoke	http://marklogic.com/xdmp/privileges/odbc-invoke	privilege to execute invoke statements from odbc	odbc invokes
odbc:invoke-in	http://marklogic.com/xdmp/privileges/odbc-invoke-in	privilege to execute invoke statements from odbc	odbc invokes into another database
odbc:invoke-modules-change	http://marklogic.com/xdmp/privileges/odbc-invoke-modules-change	privilege to execute invoke statements that change a modules database from odbc	odbc invokes that change the modules database
odbc:invoke-modules-change-file	http://marklogic.com/xdmp/privileges/odbc-invoke-modules-change-file	privilege to execute invoke statements that change a filesystem root from odbc	odbc invokes that change the filesystem root
odbc:spawn	http://marklogic.com/xdmp/privileges/odbc-spawn	privilege to execute spawn statements from odbc	odbc spawns
odbc:spawn-in	http://marklogic.com/xdmp/privileges/odbc-spawn-in	privilege to execute spawn statements from odbc	odbc spawns into another database
odbc:spawn-modules-change	http://marklogic.com/xdmp/privileges/odbc-spawn-modules-change	privilege to execute spawn statements that change a modules database from odbc	odbc spawn that change the modules database

Name	Action URI	Description	Protects Function
odbc:spawn-modules-change-file	http://marklogic.com/xdmp/privileges/odbc-spawn-modules-change-file	privilege to execute spawn statements that change a filesystem root from odbc	odbc spawn that change the filesystem root
path-add-permissions	http://marklogic.com/xdmp/privileges/path-add-permissions	privilege to add permissions for a protected path	sec:path-add-permissions
path-get-permissions	http://marklogic.com/xdmp/privileges/path-get-permissions	privilege to return permissions for a protected path	sec:path-get-permissions
path-remove-permissions	http://marklogic.com/xdmp/privileges/path-remove-permissions	privilege to remove permissions for a protected path	sec:path-remove-permissions
path-set-permissions	http://marklogic.com/xdmp/privileges/path-set-permissions	privilege to set permissions for a protected path	sec:path-set-permissions
pki	http://marklogic.com/xdmp/privileges/pki	privilege to use the PKI functions.	pki:create-template, pki:delete-certificate, pki:delete-template, pki:generate-certificate-request, pki:generate-template-certificate-authority, pki:generate-temporary-certificate, pki:generate-temporary-certificate-if-necessary, pki:get-certificate, pki:get-certificate-pem, pki:get-certificate-xml, pki:get-certificates, pki:get-certificates-for-template, pki:get-certificates-for-template-xml, pki:get-pending-certificate-request, pki:get-pending-certificate-requests-pem, pki:get-pending-certificate-requests-xml, pki:get-template, pki:get-template-by-name, pki:get-template-certificate-authority, pki:get-template-ids, pki:get-trusted-certificate-ids, pki:insert-certificate-revocation-list, pki:insert-signed-certificates, pki:insert-template, pki:insert-trusted-certificates, pki:is-temporary, pki:need-certificate, pki:template-get-description, pki:template-get-id, pki:template-get-key-options, pki:template-get-key-type, pki:template-get-name, pki:template-get-request, pki:template-get-version, pki:template-in-use, pki:template-set-description, pki:template-set-key-options, pki:template-set-key-type, pki:template-set-name, pki:template-set-request
plugin-register	http://marklogic.com/xdmp/privileges/plugin-register	privilege to use the plugin API	plugin:register
plugin-server-fields	http://marklogic.com/xdmp/privileges/plugin-server-fields	privilege to use the plugin API	Used by the plugin API

Name	Action URI	Description	Protects Function
prepare-any-xa-transactions	http://marklogic.com/xdmp/privileges/prepare-any-xa-transactions	privilege to run built-in to prepare XA transactions for any transactions	xdmp:xa-prepare
prepare-my-xa-transactions	http://marklogic.com/xdmp/privileges/prepare-my-xa-transactions	privilege to run built-in to prepare XA transactions for the user's transactions	xdmp:xa-prepare
privilege-add-roles	http://marklogic.com/xdmp/privileges/privilege-add-roles	privilege to assign the privilege to additional roles	sec:privilege-add-roles
privilege-get-roles	http://marklogic.com/xdmp/privileges/privilege-get-roles	privilege to get all roles associated with a privilege	sec:privilege-get-roles
privilege-remove-roles	http://marklogic.com/xdmp/privileges/privilege-remove-roles	privilege to remove privilege from roles to which it is assigned	sec:privilege-remove-roles
privilege-set-name	http://marklogic.com/xdmp/privileges/privilege-set-name	privilege to set a privilege's name	sec:privilege-set-name
privilege-set-roles	http://marklogic.com/xdmp/privileges/privilege-set-roles	privilege to set roles associated with a privilege	sec:privilege-set-roles
profile-any-requests	http://marklogic.com/xdmp/privileges/profile-any-requests	privilege to profile requests initiated by any user	prof:enable and other profile APIs
profile-my-requests	http://marklogic.com/xdmp/privileges/profile-my-requests	privilege to profile requests initiated by the user running the request from which profiling is called	prof:enable and other profile APIs
protect-collection	http://marklogic.com/xdmp/privileges/protect-collection	privilege to make a new or existing collection protected	sec:protect-collection
protect-path	http://marklogic.com/xdmp/privileges/protect-path	privilege to protect a path	sec:protect-path
qconsole	http://marklogic.com/xdmp/privileges/qconsole	privilege to run Query Console	
redaction-user	http://marklogic.com/xdmp/privileges/redaction-user	privilege to validate and set redaction rules	rdt:rule-validate rdt:redact
remove-amp	http://marklogic.com/xdmp/privileges/remove-amp	privilege to remove an amp from the security database	sec:remove-amp
remove-credential	http://marklogic.com/xdmp/privileges/remove-credential	privilege to remove credentials	sec:remove-credential
remove-credential-by-id	http://marklogic.com/xdmp/privileges/remove-credential-by-id	privilege to remove credentials	sec:remove-credential-by-id
remove-external-security	http://marklogic.com/xdmp/privileges/remove-external-security	privilege to remove external authentication configuration objects	sec:remove-external-security
remove-path	http://marklogic.com/xdmp/privileges/remove-path	privilege to remove protection from protected paths	sec:remove-path
remove-privilege	http://marklogic.com/xdmp/privileges/remove-privilege	privilege to remove a privilege from the security database	sec:remove-privilege
remove-query-rolesets	http://marklogic.com/xdmp/privileges/remove-query-rolesets	privilege to remove query rolesets from the Security database	sec:remove-query-rolesets
remove-role	http://marklogic.com/xdmp/privileges/remove-role	privilege to remove a role from the security database	sec:remove-role
remove-role-from-amps	http://marklogic.com/xdmp/privileges/remove-role-from-amps	privilege to remove a role from all amps in the security database	sec:remove-role-from-amps
remove-role-from-privileges	http://marklogic.com/xdmp/privileges/remove-role-from-privileges	privilege to remove a role from all privileges in the security database	sec:remove-role-from-privileges

Name	Action URI	Description	Protects Function
remove-role-from-roles	http://marklogic.com/xdmp/privileges/remove-role-from-roles	privilege to remove a role from all roles in the security database	sec:remove-role-from-roles
remove-role-from-users	http://marklogic.com/xdmp/privileges/remove-role-from-users	privilege to remove a role from all users in the security database	sec:remove-role-from-users
remove-user	http://marklogic.com/xdmp/privileges/remove-user	privilege to remove a user from the security database	sec:remove-user
rest-admin	http://marklogic.com/xdmp/privileges/rest-admin	privilege to perform administrative tasks using the REST API	REST APIs
rest-reader	http://marklogic.com/xdmp/privileges/rest-reader	privilege to perform read operations using the REST API	REST APIs
rest-tracer	http://marklogic.com/xdmp/privileges/rest-tracer		
rest-writer	http://marklogic.com/xdmp/privileges/rest-writer	privilege to perform update tasks using the REST API	REST APIs
role-add-roles	http://marklogic.com/xdmp/privileges/role-add-roles	privilege to add roles to the roles of a specified role	sec:role-add-roles
role-exists	http://marklogic.com/xdmp/privileges/get-role	privilege to find out if a role exists	sec:role-exists
role-get-compartment	http://marklogic.com/xdmp/privileges/role-get-compartment	privilege to get a role's compartment	sec:role-get-compartment
role-get-default-collections	http://marklogic.com/xdmp/privileges/role-get-default-collections	privilege to get a role's default collections	sec:role-get-default-collections
role-get-default-permissions	http://marklogic.com/xdmp/privileges/role-get-default-permissions	privilege to get a role's default permissions	sec:role-get-default-permissions
role-get-description	http://marklogic.com/xdmp/privileges/role-get-description	privilege to get a role's description	sec:role-get-description
role-get-external-names	http://marklogic.com/xdmp/privileges/role-get-external-names	privilege to get a role's external LDAP group names	sec:role-get-external-names
role-get-roles	http://marklogic.com/xdmp/privileges/role-get-roles	privilege to get all the roles included in the specified role	sec:role-get-roles
role-privileges	http://marklogic.com/xdmp/privileges/role-privileges	privilege to get all the privileges for a given role	sec:role-privileges
role-remove-roles	http://marklogic.com/xdmp/privileges/role-remove-roles	privilege to remove roles from the roles of a specified role	sec:role-remove-roles
role-set-default-collections	http://marklogic.com/xdmp/privileges/role-set-default-collections	privilege to set a role's default collections	sec:role-set-default-collections
role-set-default-permissions	http://marklogic.com/xdmp/privileges/role-set-default-permissions	privilege to set a role's default permissions	sec:role-set-default-permissions
role-set-description	http://marklogic.com/xdmp/privileges/role-set-description	privilege to set a role's name	sec:role-set-description
role-set-external-names	http://marklogic.com/xdmp/privileges/role-set-external-names	privilege to set external LDAP distinguished names for a role	sec:role-set-external-names
role-set-name	http://marklogic.com/xdmp/privileges/role-set-name	privilege to change a role's name	sec:role-set-name
role-set-roles	http://marklogic.com/xdmp/privileges/role-set-roles	privilege to change all the roles in the specified role	sec:role-set-roles

Name	Action URI	Description	Protects Function
saml-entity-delete	http://marklogic.com/xdmp/privileges/saml-entity-delete	privilege to delete a SAML entity	sec:saml-entity-delete
saml-entity-insert	http://marklogic.com/xdmp/privileges/saml-entity-insert	privilege to insert a SAML entity into the Security database	sec:saml-entity-insert
sem:sparql	http://marklogic.com/xdmp/privileges/sem-sparql	privilege to run a sparql query	sem:sparql
sem:sparql-update	http://marklogic.com/xdmp/privileges/sem-sparql-update	privilege to run a sparql update	sem:sparql-update
set-any-time-limit	http://marklogic.com/xdmp/privileges/xdmp-set-request-time-limit-any	privilege to change the request time limit	xdmp:set-request-time-limit
set-any-transaction-name	http://marklogic.com/xdmp/privileges/xdmp-set-transaction-name-any	privilege to set a name for any transaction	xdmp:set-transaction-name
set-any-transaction-time-limit	http://marklogic.com/xdmp/privileges/xdmp-set-transaction-time-limit-any	privilege to set a time limit for any transaction	xdmp:set-transaction-time-limit
set-my-time-limit	http://marklogic.com/xdmp/privileges/xdmp-set-request-time-limit-my	privilege to change the request time limit	xdmp:set-request-time-limit
set-my-transaction-name	http://marklogic.com/xdmp/privileges/xdmp-set-transaction-name-my	privilege to set a name for the user's transactions	xdmp:set-transaction-name
set-my-transaction-time-limit	http://marklogic.com/xdmp/privileges/xdmp-set-transaction-time-limit-my	privilege to set a time limit for the user's transactions	xdmp:set-transaction-time-limit
status-builtins	http://marklogic.com/xdmp/privileges/status	privilege to access the status built-ins	status built-ins
switch-task-user	http://marklogic.com/xdmp/privileges/switch-task-user	privilege for setting the task-user if it is not the current user	
temporal-admin	http://marklogic.com/xdmp/privileges/temporal-admin	privilege to execute temporal admin functions	All temporal admin functions
temporal-internal	http://marklogic.com/xdmp/privileges/temporal-internal	internal temporal privilege	
temporal-document-protect	http://marklogic.com/xdmp/privileges/temporal-document-protect	privilege to protect a temporal document from certain temporal operations for a period of time	temporal:document-protect
temporal:document-wipe	http://marklogic.com/xdmp/privileges/temporal-document-wipe	privilege to delete all versions of a temporal document	temporal:document-wipe
temporal:set-lsqt-automation	http://marklogic.com/xdmp/privileges/temporal-set-lsqt-automation	privilege to set Last Stable Query Time (LSQT) management to automatic	temporal:set-lsqt-automation
temporal:set-use-lsqt	http://marklogic.com/xdmp/privileges/temporal-set-use-lsqt	privilege to enable or disable the use of LSQT (Last Stable Query Time) on temporal collections	temporal:set-use-lsqt
temporal:statement-set-system-time	http://marklogic.com/xdmp/privileges/temporal-statement-set-system-time	privilege to set the system start time on temporal documents	temporal:statement-set-system-time
term-query	http://marklogic.com/xdmp/privileges/term-query		cts:term-query
database-create-sub-database	http://marklogic.com/xdmp/privileges/database-create-sub-database	privilege to create sub databases	tieredstorage:database-create-sub-database
database-create-super-database	http://marklogic.com/xdmp/privileges/database-create-super-database	privilege to create super databases	tieredstorage:database-create-super-database

Name	Action URI	Description	Protects Function
database-delete-sub-database	http://marklogic.com/xdmp/privileges/database-delete-sub-database	privilege to delete sub databases	tieredstorage:database-delete-sub-database
database-delete-super-database	http://marklogic.com/xdmp/privileges/database-delete-super-database	privilege to delete super databases	tieredstorage:database-delete-super-database
database-partition-numbers	http://marklogic.com/xdmp/privileges/database-partition-numbers	privilege to return the partition numbers of the forests in a database	tieredstorage:database-partition-numbers
database-partitions	http://marklogic.com/xdmp/privileges/database-partitions	privilege to return the names of the partitions in a database	tieredstorage:database-partitions
forest-combine	http://marklogic.com/xdmp/privileges/forest-combine	privilege to combine data in multiple forests into one new forest	tieredstorage:forest-combine
forest-migrate	http://marklogic.com/xdmp/privileges/forest-migrate	privilege to move data in a forest to new data directories	tieredstorage:forest-migrate
partition-create	http://marklogic.com/xdmp/privileges/partition-create	privilege to create a query partition	tieredstorage:query-partition-create
partition-delete	http://marklogic.com/xdmp/privileges/partition-delete	privilege to delete a query partition	tieredstorage:partition-delete
partition-delete-query	http://marklogic.com/xdmp/privileges/partition-delete-query	privilege to delete a query from a partition	tieredstorage:partition-delete-query
partition-forests	http://marklogic.com/xdmp/privileges/partition-forests	privilege to returns ids of the forests in a query partition	tieredstorage:partition-forests
partition-get-exclusion-enabled	http://marklogic.com/xdmp/privileges/partition-get-exclusion-enabled	privilege to return the safe-to-exclude setting for a database	tieredstorage:partition-get-exclusion-enabled
partition-get-query	http://marklogic.com/xdmp/privileges/partition-get-query	privilege to return the query of a partition	tieredstorage:partition-get-query
partition-migrate	http://marklogic.com/xdmp/privileges/partition-migrate	privilege to migrate forests in a partition to a data directory and hosts	tieredstorage:partition-migrate
partition-number-forests	http://marklogic.com/xdmp/privileges/partition-number-forests	privilege to return the IDs of the forests associated with a partition	tieredstorage:partition-number-forests
partition-queries	http://marklogic.com/xdmp/privileges/partition-queries	privilege to return the queries in a schema database	tieredstorage:partition-queries
partition-resize	http://marklogic.com/xdmp/privileges/partition-resize	privilege to create or combine forests in a partition	tieredstorage:partition-resize
partition-set-availability	http://marklogic.com/xdmp/privileges/partition-set-availability	privilege to set the availability of the partition	tieredstorage:partition-set-availability
partition-set-exclusion-enabled	http://marklogic.com/xdmp/privileges/partition-set-exclusion-enabled	privilege to exclude a query partition from being searched if the search query does not match the query assignment policy set for the partition	tieredstorage:partition-set-exclusion-enabled
partition-set-query	http://marklogic.com/xdmp/privileges/partition-set-query	privilege to set the query for a partition	tieredstorage:partition-set-query
partition-set-updates-allowed	http://marklogic.com/xdmp/privileges/partition-set-updates-allowed	privilege to set update-allowed state for the forests in a partition	tieredstorage:partition-set-updates-allowed
partition-transfer	http://marklogic.com/xdmp/privileges/partition-transfer	privilege to transfer a partition from one database to another	tieredstorage:partition-transfer

Name	Action URI	Description	Protects Function
unprotect-collection	http://marklogic.com/xdmp/privileges/unprotect-collection	privilege to change roles for a collection	xdmp:document-add-collections, xdmp:document-remove-collections, xdmp:document-set-collections
unprotect-path	http://marklogic.com/xdmp/privileges/unprotect-path	privilege to remove a protection from a protected path	sec:unprotect-path
unprotected-collections	http://marklogic.com/xdmp/privileges/unprotected-collections	privilege to add to or remove from collections that are unprotected	xdmp:document-add-collections, xdmp:document-remove-collections, xdmp:document-set-collections
unprotected-uri	http://marklogic.com/xdmp/privileges/unprotected-uri	privilege to create document with uri's that are unprotected	xdmp:document-insert, xdmp:load
user-add-roles	http://marklogic.com/xdmp/privileges/user-add-roles	privilege to add roles to a user	sec:user-add-roles
user-exists	http://marklogic.com/xdmp/privileges/get-user	privilege to check if a user exists in the security database	sec:user-exists
user-get-default-collections	http://marklogic.com/xdmp/privileges/user-gt-default-collections	privilege to get a user's default collections	sec:user-get-default-collections
user-get-default-permissions	http://marklogic.com/xdmp/privileges/user-get-default-permissions	privilege to get user's default permissions	sec:user-get-default-permissions
user-get-description	http://marklogic.com/xdmp/privileges/user-get-description	privilege to get user's description	sec:user-get-description (if not logged in as user)
user-get-external-names	http://marklogic.com/xdmp/privileges/user-get-external-names	privilege to get the external LDAP group names assigned to a user	sec:user-get-external-names
user-get-password-extra	http://marklogic.com/xdmp/privileges/user-get-password-extra	privilege to get the password-extra element from the user document	sec:user-get-password-extra
user-get-roles	http://marklogic.com/xdmp/privileges/user-get-roles	privilege to get user's roles	sec:user-get-roles (if not logged in as user)
user-privileges	http://marklogic.com/xdmp/privileges/user-privileges	privilege to get a user's complete privileges	sec:user-privileges (if not logged in as user)
user-remove-roles	http://marklogic.com/xdmp/privileges/user-remove-roles	privilege to remove roles from a user	sec:user-remove-roles
user-set-default-collections	http://marklogic.com/xdmp/privileges/user-set-default-collections	privilege to set a user's default collections	sec:user-set-default-collections
user-set-default-permissions	http://marklogic.com/xdmp/privileges/user-set-default-permissions	privilege to set a user's default permissions	sec:user-set-default-permissions
user-set-description	http://marklogic.com/xdmp/privileges/user-set-description	privilege to set a user's description	sec:user-set-description (if not logged in as user)
user-set-external-names	http://marklogic.com/xdmp/privileges/user-set-external-names	privilege to set the external names for a user	sec:user-set-external-names
user-set-name	http://marklogic.com/xdmp/privileges/user-set-name	privilege to set a user's name	sec:user-set-name (if not logged in as user)
user-set-password	http://marklogic.com/xdmp/privileges/user-set-password	privilege to set user's password	sec:user-set-password (if not logged in as user)

Name	Action URI	Description	Protects Function
user-set-password-extra	http://marklogic.com/xdmp/privileges/user-set-password-extra	privilege to set the password-extra element in the user document	sec:user-set-password-extra
user-set-roles	http://marklogic.com/xdmp/privileges/user-set-roles	privilege to set a user's role	sec:user-set-roles
view-create	http://marklogic.com/xdmp/privileges/create-view	privilege to create a view	view:create
view-schema-create	http://marklogic.com/xdmp/privileges/create-schema	privilege to create a relational schema	view:schema-create
xdbc-eval	http://marklogic.com/xdmp/privileges/xdbc-eval	privilege to execute eval statements from xcc or xdbc	xdmp:eval
xdbc-eval-in	http://marklogic.com/xdmp/privileges/xdbc-eval-in	privilege to execute eval-in statements from xcc or xdbc	xdmp:eval-in
xdbc-eval-modules-change	http://marklogic.com/xdmp/privileges/xdbc-eval-modules-change	privilege to execute eval statements that change a modules database from xcc or xdbc	xdmp:eval
xdbc-eval-modules-change-file	http://marklogic.com/xdmp/privileges/xdbc-eval-modules-change-file	privilege to execute eval statements that change a filesystem root from xcc or xdbc	xdmp:eval
xdbc-insert	http://marklogic.com/xdmp/privileges/xdbc-insert-in	privilege to execute insert statements from xcc or xdbc	xcc or xdbc inserts
xdbc-insert-in	http://marklogic.com/xdmp/privileges/xdbc-insert-in	privilege to execute insert statements from xcc or xdbc	xdbc or xcc inserts into another database
xdbc-invoke	http://marklogic.com/xdmp/privileges/xdbc-invoke	privilege to execute invoke statements from xcc or xdbc	xdbc or xcc invokes
xdbc-invoke-in	http://marklogic.com/xdmp/privileges/xdbc-invoke-in	privilege to execute invoke statements from xcc or xdbc	xdbc or xcc invokes into another database
xdbc-invoke-modules-change	http://marklogic.com/xdmp/privileges/xdbc-invoke-modules-change	privilege to execute invoke statements that change a modules database from xcc or xdbc	xdbc or xcc invokes that change the modules database
xdbc-invoke-modules-change-file	http://marklogic.com/xdmp/privileges/xdbc-invoke-modules-change-file	privilege to execute invoke statements that change a filesystem root from xcc or xdbc	xdbc or xcc invokes that change the filesystem root
xdbc-spawn	http://marklogic.com/xdmp/privileges/xdbc-spawn	privilege to execute spawn statements from xcc or xdbc	xdbc or xcc spawns
xdbc-spawn-in	http://marklogic.com/xdmp/privileges/xdbc-spawn-in	privilege to execute spawn statements from xcc or xdbc	xdbc or xcc spawns into another database
xdbc-spawn-modules-change	http://marklogic.com/xdmp/privileges/xdbc-spawn-modules-change	privilege to execute spawn statements that change a modules database from xcc or xdbc	xdbc or xcc spawn that change the modules database
xdbc-spawn-modules-change-file	http://marklogic.com/xdmp/privileges/xdbc-spawn-modules-change-file	privilege to execute spawn statements that change a filesystem root from xcc or xdbc	xdbc or xcc spawn that change the filesystem root
xdmp-add-response-header	http://marklogic.com/xdmp/privileges/xdmp-add-response-header	privilege to use the function that adds a response header to a request functions.	admin built-ins, alert-user
xdmp-address-bindable	http://marklogic.com/xdmp/privileges/xdmp-address-bindable	privilege to perform admin functions.	admin built-ins
xdmp-alert-admin	http://marklogic.com/xdmp/privileges/xdmp-alert-admin	privilege to perform alerting admin functions.	xdmp:alert-admin
xdmp-alert-internal	http://marklogic.com/xdmp/privileges/xdmp-alert-internal	privilege used by the Alerting API functions.	xdmp:alert-internal

Name	Action URI	Description	Protects Function
xdmp-alert-user	http://marklogic.com/xdmp/privileges/xdmp-alert-user	privilege to perform user-level Alerting functions.	xdmp:alert-user, xdmp:alert-admin
xdmp-amp-roles	http://marklogic.com/xdmp/privileges/xdmp-amp-roles	privilege to get an amp's roles	xdmp:amp-roles
xdmp-binary-join	http://marklogic.com/xdmp/privileges/xdmp-binary-join	privilege to run the binary-join built-n	xdmp:binary-join
xdmp-compressed-tree-cache-partitions	http://marklogic.com/xdmp/privileges/xdmp-compressed-tree-cache-partitions	privilege to perform admin functions	admin built-ins
xdmp-compressed-tree-cache-size	http://marklogic.com/xdmp/privileges/xdmp-compressed-tree-cache-size	privilege to perform admin functions	admin built-ins
xdmp-data-directory	http://marklogic.com/xdmp/privileges/xdmp-data-directory	privilege to access the data directory	admin built-ins
xdmp-database-backup	http://marklogic.com/xdmp/privileges/xdmp-database-backup	privilege to perform a database backup	admin built-ins
xdmp-database-backup-cancel	http://marklogic.com/xdmp/privileges/xdmp-database-backup-cancel	privilege to cancel a database backup	admin built-ins
xdmp-database-backup-purge	http://marklogic.com/xdmp/privileges/xdmp-database-backup-purge	privilege to get purge a database backup	admin built-ins
xdmp-database-backup-status	http://marklogic.com/xdmp/privileges/xdmp-database-backup-status	privilege to get status for a database backup	admin built-ins
xdmp-database-backup-validate	http://marklogic.com/xdmp/privileges/xdmp-database-backup-validate	privilege to validate a database backup	admin built-ins
xdmp-database-create-sub-database	http://marklogic.com/xdmp/privileges/xdmp-database-create-sub-database	privilege to create a sub database	tieredstorage:database-create-sub-database
xdmp-database-create-super-database	http://marklogic.com/xdmp/privileges/xdmp-database-create-super-database	privilege to create a super database	tieredstorage:database-create-super-database
xdmp-database-delete-sub-database	http://marklogic.com/xdmp/privileges/xdmp-database-delete-sub-database	privilege to delete a sub database	tieredstorage:database-delete-sub-database
xdmp-database-delete-super-database	http://marklogic.com/xdmp/privileges/xdmp-database-delete-super-database	privilege to delete a super database	tieredstorage:database-delete-super-database
xdmp-database-incremental-backup	http://marklogic.com/xdmp/privileges/xdmp-database-incremental-backup	privilege to validate if forests can be incrementally backed up	xdmp:database-incremental-backup
xdmp-database-incremental-backup-validate	http://marklogic.com/xdmp/privileges/xdmp-database-incremental-backup-validate	privilege to start an incremental backup of forests	xdmp:database-incremental-backup-validate
xdmp-database-restore	http://marklogic.com/xdmp/privileges/xdmp-database-restore	privilege to perform a database restore	admin built-ins
xdmp-database-restore-cancel	http://marklogic.com/xdmp/privileges/xdmp-database-backup	privilege to cancel a database restore	admin built-ins
xdmp-database-restore-status	http://marklogic.com/xdmp/privileges/xdmp-database-restore-status	privilege to get status for a database restore	admin built-ins
xdmp-database-restore-validate	http://marklogic.com/xdmp/privileges/xdmp-database-restore-validate	privilege to validate a database restore	admin built-ins

Name	Action URI	Description	Protects Function
xdmp-default-in-memory-geospatial-region-index-size	http://marklogic.com/xdmp/privileges/xdmp-default-in-memory-geospatial-region-index-size	privilege to perform admin functions.	admin built-ins
xdmp-default-in-memory-limit	http://marklogic.com/xdmp/privileges/xdmp-default-in-memory-limit	privilege to perform admin functions.	admin built-ins
xdmp-default-in-memory-list-size	http://marklogic.com/xdmp/privileges/xdmp-default-in-memory-list-size	privilege to perform admin functions.	admin built-ins
xdmp-default-in-memory-range-index-size	http://marklogic.com/xdmp/privileges/xdmp-default-in-memory-range-index-size	privilege to perform admin functions	admin built-ins
xdmp-default-in-memory-reverse-index-size	http://marklogic.com/xdmp/privileges/xdmp-default-in-memory-reverse-index-size	privilege to perform admin functions	admin built-ins
xdmp-default-in-memory-tree-size	http://marklogic.com/xdmp/privileges/xdmp-default-in-memory-tree-size	privilege to perform admin functions	admin built-ins
xdmp-default-in-memory-triple-index-size	http://marklogic.com/xdmp/privileges/xdmp-default-in-memory-triple-index-size	privilege to perform admin functions.	admin built-ins
xdmp-default-journal-count	http://marklogic.com/xdmp/privileges/xdmp-default-journal-count	privilege to perform admin functions.	admin built-ins
xdmp-default-journal-size	http://marklogic.com/xdmp/privileges/xdmp-default-journal-size	privilege to perform admin functions.	admin built-ins
xdmp-default-preallocate-journals	http://marklogic.com/xdmp/privileges/xdmp-default-preallocate-journals	privilege to perform admin functions.	admin built-ins
xdmp-default-s3-domain	http://marklogic.com/xdmp/privileges/xdmp-default-s3-domain	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file	privilege to perform admin functions	admin built-ins
xdmp-delete-cluster-config-file-assignments	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/assignments.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file-ca-bundle	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/ca-bundle.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file-calendars	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/calendars.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file-clusters	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/clusters.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file-countries	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/countries.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file-databases	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/databases.xml	privilege to perform admin functions.	admin built-ins

Name	Action URI	Description	Protects Function
xdmp-delete-cluster-config-file-dtfmt-languages	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/dtfmt-languages.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file-groups	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/groups.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file-hosts	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/hosts.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file-keystore	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/keystore.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file-languages	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/languages.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file-mimetypes	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/mimetypes.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file-security	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/security.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file-server	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/server.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file-tokenizer	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/tokenizer.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-cluster-config-file-user-languages	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file/user-languages.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file	privilege to perform admin functions	admin built-ins
xdmp-delete-host-config-file-assignments	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/assignments.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file-ca-bundle	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/ca-bundle.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file-calendars	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/calendars.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file-clusters	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/clusters.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file-countries	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/countries.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file-databases	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/databases.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file-dtfmt-languages	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/dtfmt-languages.xml	privilege to perform admin functions.	admin built-ins

Name	Action URI	Description	Protects Function
xdmp-delete-host-config-file-groups	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/groups.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file-hosts	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/hosts.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file-keystore	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/keystore.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file-languages	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/languages.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file-mimetypes	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/mimetypes.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file-security	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/security.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file-server	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/server.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file-tokenizer	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/tokenizer.xml	privilege to perform admin functions.	admin built-ins
xdmp-delete-host-config-file-user-languages	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file/user-languages.xml	privilege to perform admin functions.	admin built-ins
xdmp-disable-event	http://marklogic.com/xdmp/privileges/xdmp-disable-event	privilege to perform admin functions	admin built-ins
xdmp-document-get	http://marklogic.com/xdmp/privileges/xdmp-document-get	privilege to execute function	xdmp:document-get
xdmp-document-load	http://marklogic.com/xdmp/privileges/xdmp-document-load	privilege to execute function	xdmp:document-load
xdmp-email	http://marklogic.com/xdmp/privileges/xdmp-email	privilege to email	xdmp:email
xdmp-email-address	http://marklogic.com/xdmp/privileges/xdmp-email-address	privilege to perform admin functions	admin built-ins
xdmp-enable-event	http://marklogic.com/xdmp/privileges/xdmp-enable-event	privilege to perform admin functions	admin built-ins
xdmp-eval	http://marklogic.com/xdmp/privileges/xdmp-eval	privilege to perform eval functions	xdmp:eval
xdmp-eval-in	http://marklogic.com/xdmp/privileges/xdmp-eval-in	privilege to perform eval-in functions	xdmp:eval-in
xdmp-eval-modules-change	http://marklogic.com/xdmp/privileges/xdmp-eval-modules-change	privilege to execute eval statements that change a modules database	xdmp:eval statements that change the modules database
xdmp-eval-modules-change-file	http://marklogic.com/xdmp/privileges/xdmp-eval-modules-change-file	privilege to execute eval statements that change a filesystem root	xdmp:eval statements that change the filesystem root
xdmp-eval-transaction	http://marklogic.com/xdmp/privileges/xdmp-eval-transaction	privilege to run eval statements with the transaction option	xdmp:eval statements that start a new transaction
xdmp-expanded-tree-cache-partitions	http://marklogic.com/xdmp/privileges/xdmp-expanded-tree-cache-partitions	privilege to perform admin functions	admin built-ins

Name	Action URI	Description	Protects Function
xdmp-expanded-tree-cache-size	http://marklogic.com/xdmp/privileges/xdmp-expanded-tree-cache-size	privilege to perform admin functions	admin built-ins
xdmp-external-binary	http://marklogic.com/xdmp/privileges/xdmp-external-binary	privilege to access external binary function	xdmp:external-binary
xdmp-filesystem-directory	http://marklogic.com/xdmp/privileges/xdmp-filesystem-directory	privilege to run the built-in	xdmp:filesystem-directory
xdmp-filesystem-directory-create	http://marklogic.com/xdmp/privileges/xdmp-filesystem-directory-create	privilege to perform admin functions	admin built-ins
xdmp-filesystem-directory-delete	http://marklogic.com/xdmp/privileges/xdmp-filesystem-directory-delete	privilege to perform admin functions.	admin built-ins
xdmp-filesystem-file	http://marklogic.com/xdmp/privileges/xdmp-filesystem-file	privilege to perform admin functions	xdmp:filesystem-file
xdmp-filesystem-file-delete	http://marklogic.com/xdmp/privileges/xdmp-filesystem-file-delete	privilege to perform admin functions.	admin built-ins
xdmp-filesystem-file-exists	http://marklogic.com/xdmp/privileges/xdmp-filesystem-file-exists	privilege to run the built-in	xdmp:filesystem-file-exists
xdmp-filesystem-file-get-time	http://marklogic.com/xdmp/privileges/xdmp-filesystem-file-get-time	privilege to perform admin functions.	xdmp:filesystem-file-get-time
xdmp-filesystem-file-length	http://marklogic.com/xdmp/privileges/xdmp-filesystem-file-length	privilege to run the built-in	xdmp:filesystem-file-length
xdmp-forest-backup	http://marklogic.com/xdmp/privileges/xdmp-forest-backup	privilege to perform admin functions	admin built-ins
xdmp-forest-clear	http://marklogic.com/xdmp/privileges/xdmp-forest-clear	privilege to perform admin functions	admin built-ins
xdmp-forest-combine	http://marklogic.com/xdmp/privileges/xdmp-forest-combine	privilege to perform admin functions	admin built-in
xdmp-forest-copy	http://marklogic.com/xdmp/privileges/xdmp-forest-copy	privilege to perform admin functions	admin built-in
xdmp-forest-delete	http://marklogic.com/xdmp/privileges/xdmp-forest-delete	privilege to perform admin functions	admin built-ins
xdmp-forest-directory-delete	http://marklogic.com/xdmp/privileges/xdmp-forest-directory-delete	privilege to perform admin functions	admin built-in
xdmp-forest-directory-exists	http://marklogic.com/xdmp/privileges/xdmp-forest-directory-exists	privilege to perform admin functions	admin built-in
xdmp-forest-get-readonly	http://marklogic.com/xdmp/privileges/xdmp-forest-get-readonly	privilege to perform admin functions	admin built-in
xdmp-forest-rename	http://marklogic.com/xdmp/privileges/xdmp-forest-rename	privilege to perform admin functions	admin built-in
xdmp-forest-restart	http://marklogic.com/xdmp/privileges/xdmp-forest-restart	privilege to perform admin functions	admin built-ins
xdmp-forest-restore	http://marklogic.com/xdmp/privileges/xdmp-forest-restore	privilege to perform admin functions	admin built-ins

Name	Action URI	Description	Protects Function
xdmp-forest-rollback	http://marklogic.com/xdmp/privileges/xdmp-forest-rollback	privilege to perform admin functions	admin built-ins
xdmp-forest-set-readonly	http://marklogic.com/xdmp/privileges/xdmp-forest-set-readonly	privilege to perform admin functions	admin built-in
xdmp-get	http://marklogic.com/xdmp/privileges/xdmp-get	privilege to get a document into memory	xdmp:get
xdmp-get-forest-keys	http://marklogic.com/xdmp/privileges/xdmp-get-forest-keys	privilege to perform admin functions	admin built-ins
xdmp-get-hot-updates	http://marklogic.com/xdmp/privileges/xdmp-get-hot-updates	privilege to perform admin functions	admin built-ins
xdmp-get-orphaned-binaries	http://marklogic.com/xdmp/privileges/xdmp-get-orphaned-binaries	privilege to run the built-in	xdmp:get-orphaned-binaries
xdmp-get-server-field	http://marklogic.com/xdmp/privileges/xdmp-get-server-field	privilege to get server fields	xdmp:get-server-field
xdmp-get-server-field-names	http://marklogic.com/xdmp/privileges/xdmp-get-server-field-names	privilege to get server fields names	xdmp:get-server-field-names
xdmp-get-session-field	http://marklogic.com/xdmp/privileges/xdmp-get-session-field	privilege to get session fields	xdmp:get-session-field
xdmp-get-session-field-names	http://marklogic.com/xdmp/privileges/xdmp-get-session-field-names	privilege to get session field names	xdmp:get-session-field-names
xdmp-getenv	http://marklogic.com/xdmp/privileges/xdmp-getenv	privilege to perform admin function	admin built-ins
xdmp-host-cores	http://marklogic.com/xdmp/privileges/xdmp-host-cores	privilege to perform admin functions	admin built-ins
xdmp-host-cpus	http://marklogic.com/xdmp/privileges/xdmp-host-cpus	privilege to perform admin functions	admin built-ins
xdmp-host-size	http://marklogic.com/xdmp/privileges/xdmp-host-size	privilege to perform admin functions	admin built-ins
xdmp-hostname	http://marklogic.com/xdmp/privileges/xdmp-hostname	privilege to perform admin functions	admin built-ins
xdmp-http-get	http://marklogic.com/xdmp/privileges/xdmp-http-get	privilege to perform http function	xdmp:http-get
xdmp-http-head	http://marklogic.com/xdmp/privileges/xdmp-http-head	privilege to perform http function	xdmp:http-head
xdmp-http-options	http://marklogic.com/xdmp/privileges/xdmp-http-options	privilege to perform http function	xdmp:http-options
xdmp-http-delete	http://marklogic.com/xdmp/privileges/xdmp-http-delete	privilege to perform http function	xdmp:http-delete
xdmp-http-post	http://marklogic.com/xdmp/privileges/xdmp-http-post	privilege to perform http function	xdmp:http-post
xdmp-http-put	http://marklogic.com/xdmp/privileges/xdmp-http-put	privilege to perform http function	xdmp:http-put
xdmp-install-directory	http://marklogic.com/xdmp/privileges/xdmp-install-directory	privilege to access the installation directory	admin built-ins
xdmp-invoke	http://marklogic.com/xdmp/privileges/xdmp-invoke	privilege to perform invoke functions	xdmp:invoke
xdmp-invoke-in	http://marklogic.com/xdmp/privileges/xdmp-invoke-in	privilege to perform invoke-in functions	xdmp:invoke-in
xdmp-invoke-modules-change	http://marklogic.com/xdmp/privileges/xdmp-invoke-modules-change	privilege to execute invoke statements that change a modules database	xdmp:invoke statements that change the modules database

Name	Action URI	Description	Protects Function
xdmp-invoke-modules-change-file	http://marklogic.com/xdmp/privileges/xdmp-invoke-modules-change-file	privilege to execute invoke statements that change a filesystem root	xdmp:invoke statements that change the filesystem root
xdmp-invoke-transaction	http://marklogic.com/xdmp/privileges/xdmp-invoke-transaction	privilege to execute invoke statements that have the <transaction-id> option	xdmp:invoke
xdmp-license-accepted	http://marklogic.com/xdmp/privileges/xdmp-license-accepted	privilege to perform admin functions	admin built-ins
xdmp-license-fee	http://marklogic.com/xdmp/privileges/xdmp-license-fee	privilege to perform admin functions	admin built-ins
xdmp-license-key	http://marklogic.com/xdmp/privileges/xdmp-license-key	privilege to perform admin functions	admin built-ins
xdmp-license-key-agreement	http://marklogic.com/xdmp/privileges/xdmp-license-key-agreement	privilege to perform admin functions	admin built-ins
xdmp-license-key-cores	http://marklogic.com/xdmp/privileges/xdmp-license-key-cores	privilege to perform admin functions	admin built-ins
xdmp-license-key-cpus	http://marklogic.com/xdmp/privileges/xdmp-license-key-cpus	privilege to perform admin functions	admin built-ins
xdmp-license-key-decode	http://marklogic.com/xdmp/privileges/xdmp-license-key-decode	privilege to perform admin functions	admin built-ins
xdmp-license-key-encode	http://marklogic.com/xdmp/privileges/xdmp-license-key-encode	privilege to perform admin functions	admin built-ins
xdmp-license-key-expires	http://marklogic.com/xdmp/privileges/xdmp-license-key-expires	privilege to perform admin functions	admin built-ins
xdmp-license-key-options	http://marklogic.com/xdmp/privileges/xdmp-license-key-options	privilege to perform admin functions	admin built-ins
xdmp-license-key-size	http://marklogic.com/xdmp/privileges/xdmp-license-key-size	privilege to perform admin functions	admin built-ins
xdmp-license-key-valid	http://marklogic.com/xdmp/privileges/xdmp-license-key-valid	privilege to perform admin functions	admin built-ins
xdmp-licensee	http://marklogic.com/xdmp/privileges/xdmp-licensee	privilege to perform admin functions	admin built-ins
xdmp-list-cache-partitions	http://marklogic.com/xdmp/privileges/xdmp-list-cache-partitions	privilege to perform admin functions	admin built-ins
xdmp-list-cache-size	http://marklogic.com/xdmp/privileges/xdmp-list-cache-size	privilege to perform admin functions	admin built-ins
xdmp-load	http://marklogic.com/xdmp/privileges/xdmp-load	privilege needed to load a document from the file system	xdmp:load
xdmp-login	http://marklogic.com/xdmp/privileges/xdmp-login	privilege to log in a user without the corresponding password	xdmp-login
xdmp-merge	http://marklogic.com/xdmp/privileges/xdmp-merge	privilege to start merging the forests	xdmp-merge
xdmp-merging	http://marklogic.com/xdmp/privileges/xdmp-merging	privilege to get forest ids of forests currently merging	xdmp:merging
xdmp-missing-directories	http://marklogic.com/xdmp/privileges/xdmp-missing-directories	privilege to perform admin functions	admin built-ins

Name	Action URI	Description	Protects Function
xdmp-plan	http://marklogic.com/xdmp/privileges/xdmp-plan	privilege to perform admin functions	admin built-ins
xdmp-pre-release-expires	http://marklogic.com/xdmp/privileges/xdmp-pre-release-expires	privilege to perform admin functions	admin built-ins
xdmp-privilege-roles	http://marklogic.com/xdmp/privileges/xdmp-privilege-roles	privilege needed to get a role's privileges	xdmp:privilege-roles
xdmp-read-cluster-config-file	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-assignments	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/assignments.xml	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-assignments-schema	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/assignments.xsd	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-ca-bundle	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/ca-bundle.xml	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-calendars	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/calendars.xml	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-clusters	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/clusters.xml	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-clusters-schema	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/clusters.xsd	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-countries	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/countries.xml	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-database-schema	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/database.xsd	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-databases	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/databases.xml	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-dtfmt-langauges	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/dtfmt-languages.xml	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-group-schema	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/group.xsd	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-groups	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/groups.xml	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-host-schema	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/hosts.xsd	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-hosts	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/hosts.xml	privilege to perform admin functions	admin built-ins

Name	Action URI	Description	Protects Function
xdmp-read-cluster-config-file-keystore	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/keystore.xml	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-keystore-schema	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/keystore.xsd	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-languages	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/languages.xml	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-mimetypes	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/mimetypes.xml	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-mimetypes-schema	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/mimetypes.xsd	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-security	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/security.xml	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-server	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/server.xml	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-tokenizer	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/tokenizer.xml	privilege to perform admin functions	admin built-ins
xdmp-read-cluster-config-file-user-languages	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file/user-languages.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-assignments	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/assignments.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-ca-bundle	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/ca-bundle.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-calendars	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/calendars.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-clusters	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/clusters.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-countries	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/countries.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-databases	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/databases.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-dtfmt-languages	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/dtfmt-languages.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-groups	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/groups.xml	privilege to perform admin functions	admin built-ins

Name	Action URI	Description	Protects Function
xdmp-read-host-config-file-hosts	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/hosts.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-keystore	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/keystore.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-languages	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/languages.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-mimetypes	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/mimetypes.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-security	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/security.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-server	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/server.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-tokenizer	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/tokenizer.xml	privilege to perform admin functions	admin built-ins
xdmp-read-host-config-file-user-languages	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file/user-languages.xml	privilege to perform admin functions	admin built-ins
xdmp-remove-orphaned-binary	http://marklogic.com/xdmp/privileges/xdmp-remove-orphaned-binary	privilege to run the built-in	xdmp:remove-orphaned-binary
xdmp-restart	http://marklogic.com/xdmp/privileges/xdmp-restart	privilege to perform admin functions	admin built-ins
xdmp-role-roles	http://marklogic.com/xdmp/privileges/xdmp-role-roles	privilege to get a role's roles	xdmp:role-roles
xdmp-rotate-log-files	http://marklogic.com/xdmp/privileges/xdmp-rotate-log-files	privilege to perform admin functions	admin built-ins
xdmp-save	http://marklogic.com/xdmp/privileges/xdmp-save	privilege needed to save a document to the file system	xdmp:save
xdmp-server-backup	http://marklogic.com/xdmp/privileges/xdmp-server-backup	privilege to perform admin functions	admin built-ins
xdmp-server-import-qualities	http://marklogic.com/xdmp/privileges/xdmp-server-import-qualities	privilege to perform admin functions	admin built-ins
xdmp-server-restore	http://marklogic.com/xdmp/privileges/xdmp-server-restore	privilege to perform admin functions	admin built-ins
xdmp-set-current-transaction	http://marklogic.com/xdmp/privileges/set-current-transaction	privilege to perform the multi-statement transaction function	xdmp:set-current-transaction
xdmp-set-hot-updates	http://marklogic.com/xdmp/privileges/xdmp-set-hot-updates	privilege to perform admin functions	admin built-ins
xdmp-set-request-time-limit	http://marklogic.com/xdmp/privileges/xdmp-set-request-time-limit	privilege to set time limits for a request	xdmp:set-request-time-limit
xdmp-set-server-field	http://marklogic.com/xdmp/privileges/xdmp-set-server-field	privilege to set a server fields	xdmp:set-server-field
xdmp-set-server-field-privilege	http://marklogic.com/xdmp/privileges/xdmp-set-server-field-privilege	privilege to set a specific privilege on a server field	xdmp:set-server-field-privilege

Name	Action URI	Description	Protects Function
xdmp-set-session-field	http://marklogic.com/xdmp/privileges/xdmp-set-session-field	privilege to run the built-in	xdmp:set-session-field
xdmp-shutdown	http://marklogic.com/xdmp/privileges/xdmp-shutdown	privilege to perform admin functions	admin built-ins
xdmp-sleep	http://marklogic.com/xdmp/privileges/xdmp-sleep	privilege to perform admin functions	admin built-ins
xdmp-smtp-relay	http://marklogic.com/xdmp/privileges/xdmp-smtp-relay	privilege to perform admin functions	admin built-ins
xdmp-spawn	http://marklogic.com/xdmp/privileges/xdmp-spawn	privilege to perform spawn functions	xdmp:spawn
xdmp-spawn-in	http://marklogic.com/xdmp/privileges/xdmp-spawn-in	privilege to perform spawn-in functions	xdmp:spawn-in
xdmp-spawn-modules-change	http://marklogic.com/xdmp/privileges/xdmp-spawn-modules-change	privilege to execute spawn statements that change a modules database	xdmp:spawn statements that change the modules database
xdmp-spawn-modules-change-file	http://marklogic.com/xdmp/privileges/xdmp-spawn-modules-change-file	privilege to execute spawn statements that change a filesystem root	xdmp:spawn statements that change the filesystem root
xdmp-spawn-transaction	http://marklogic.com/xdmp/privileges/xdmp-spawn-transaction	privilege to execute spawn statements that have the <transaction-id> option	xsmp:spawn
xdmp-sql	http://marklogic.com/xdmp/privileges/xdmp-sql	privilege to perform SQL queries	xdmp:sql
xdmp-timestamp	http://marklogic.com/xdmp/privileges/xdmp-timestamp	privilege to perform point-in-time queries	xdmp:eval, xdmp:invoke (timestamp option)
xdmp-transaction-create	http://marklogic.com/xdmp/privileges/xdmp-transaction-create	privilege to run the built-in	xdmp:transaction-create
xdmp-transaction-create-xid	http://marklogic.com/xdmp/privileges/xdmp-transaction-create-xid	privilege to run the built-in	xdmp:transaction-create-xid
xdmp-triple-cache-partitions	http://marklogic.com/xdmp/privileges/xdmp-triple-cache-partitions	privilege to run the built-in	admin built-ins
xdmp-triple-cache-size	http://marklogic.com/xdmp/privileges/xdmp-triple-cache-size	privilege to run the built-in	admin built-ins
xdmp-triple-value-cache-partitions	http://marklogic.com/xdmp/privileges/xdmp-triple-value-cache-partitions	privilege to run the built-in	admin built-ins
xdmp-triple-value-cache-size	http://marklogic.com/xdmp/privileges/xdmp-triple-value-cache-size	privilege to run the built-in	admin built-ins
xdmp-user-last-login	http://marklogic.com/xdmp/privileges/xdmp-user-last-login	privilege to get run the built-in	xdmp:user-last-login
xdmp-user-roles	http://marklogic.com/xdmp/privileges/xdmp-user-roles	privilege to get a user's roles	xdmp:user-roles
xdmp-username	http://marklogic.com/xdmp/privileges/xdmp-username	privilege to perform admin functions	admin built-ins
xdmp-value	http://marklogic.com/xdmp/privileges/xdmp-value	privilege to use the "evaluate an expression" function	xdmp:value
xdmp-with-namespace	http://marklogic.com/xdmp/privileges/xdmp-with-namespace	privilege to use the "evaluate an expression preserving the namespace" function	xdmp:with-namespace
xdmp-write-cluster-config-file	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file	privilege to perform admin functions	admin built-ins

Name	Action URI	Description	Protects Function
xdmp-write-cluster-config-file-assignments	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/assignments.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-ca-bundle	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/ca-bundle.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-calendars	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/calendars.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-clusters	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/clusters.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-countries	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/countries.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-databases	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/databases.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-dtfmt-langauges	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/dtfmt-languages.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-groups	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/groups.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-hosts	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/hosts.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-keystore	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/keystore.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-languages	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/languages.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-mimetypes	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/mimetypes.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-security	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/security.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-server	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/server.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-tokenizer	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/tokenizer.xml	privilege to perform admin functions	admin built-ins
xdmp-write-cluster-config-file-user-languages	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file/user-languages.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file	privilege to perform admin functions	admin built-ins

Name	Action URI	Description	Protects Function
xdmp-write-host-config-file-assignments	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/assignments.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-ca-bundle	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/ca-bundle.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-calendars	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/calendars.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-clusters	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/clusters.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-countries	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/countries.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-databases	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/databases.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-dtfmt-langauges	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/dtfmt-languages.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-groups	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/groups.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-hosts	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/hosts.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-keystore	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/keystore.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-languages	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/languages.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-mimetypes	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/mimetypes.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-security	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/security.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-server	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/server.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-tokenizer	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/tokenizer.xml	privilege to perform admin functions	admin built-ins
xdmp-write-host-config-file-user-languages	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file/user-languages.xml	privilege to perform admin functions	admin built-ins
xdmp-xslt-eval	http://marklogic.com/xdmp/privileges/xslt-eval	privilege to use xdmp:xslt-eval	xdmp:xslt-eval
xdmp-xslt-eval-in	http://marklogic.com/xdmp/privileges/xslt-eval-in	privilege to use xdmp:xslt-eval-in	xdmp:xslt-eval-in
xdmp-xslt-eval-modules-change	http://marklogic.com/xdmp/privileges/xslt-eval-modules-change	privilege to change the modules database for xdmp:xslt-eval	xdmp:xslt-eval
xdmp-xslt-eval-modules-change-file	http://marklogic.com/xdmp/privileges/xslt-eval-modules-change-file	privilege to change the filesystem root for xdmp:xslt-eval	<xdmp:xslt-eval

Name	Action URI	Description	Protects Function
xdmp-xslt-eval-transaction	http://marklogic.com/xdmp/privileges/xslt-eval-transaction	privilege to execute xdmp:xslt-eval statements that have the <transaction-id> option	xdmp:xslt-eval
xdmp-xslt-invoke	http://marklogic.com/xdmp/privileges/xslt-invoke	privilege to use xdmp:xslt-invoke	xdmp:xslt-invoke
xdmp-xslt-invoke-in	http://marklogic.com/xdmp/privileges/xslt-invoke-in	privilege to use xdmp:xslt-invoke-in	xdmp:xslt-invoke-in
xdmp-xslt-invoke-modules-change	http://marklogic.com/xdmp/privileges/xslt-invoke-modules-change	privilege to use xdmp:xslt-invoke and change the modules database	xdmp:xslt-invoke
xdmp-xslt-invoke-modules-change-file	http://marklogic.com/xdmp/privileges/xslt-invoke-modules-change-file	privilege to use xdmp:xslt-invoke and change the App Server root	xdmp:xslt-invoke
xdmp-xslt-invoke-transaction	http://marklogic.com/xdmp/privileges/xslt-invoke-transaction	privilege to execute xdmp:xslt-invoke statements that have the <transaction-id> option	xdmp:xslt-invoke

33. Appendix C: Pre-defined Roles

The roles in this section are pre-defined in every installation of MarkLogic Server. To give a user execute privileges listed for each pre-defined role, you may either add the execute privileges individually to an existing role for the user or add the pre-defined role to the user's set of roles.

33.1. admin

The `admin` role is given all privileges and permissions to perform any action in the system. There are no default permissions associated with the `admin` role. Users with the `admin` role are considered authorized administrators; they are trusted personnel and are assumed to be non-hostile, appropriately trained, and following proper administrative procedures.

33.2. admin-builtins

The `admin-builtins` role has the execute privileges to call the admin built-in functions. These are the execute privileges given to the `admin-builtins` role:

Name	Action URI
cancel-any-request	http://marklogic.com/xdmp/privileges/cancel-any-request
cancel-my-request	http://marklogic.com/xdmp/privileges/cancel-my-request
count-builtins	http://marklogic.com/xdmp/privileges/counts
xdmp:address-bindable	http://marklogic.com/xdmp/privileges/xdmp-address-bindable
xdmp:amp-roles	http://marklogic.com/xdmp/privileges/xdmp-amp-roles
xdmp:castable-as	http://marklogic.com/xdmp/privileges/xdmp-castable-as
xdmp:compressed-tree-cache-size	http://marklogic.com/xdmp/privileges/xdmp-compressed-tree-cache-size
xdmp:compressed-tree-cache-partitions	http://marklogic.com/xdmp/privileges/xdmp-compressed-tree-cache-partitions
xdmp:default-in-memory-limit	http://marklogic.com/xdmp/privileges/xdmp-default-in-memory-limit
xdmp:default-in-memory-list-size	http://marklogic.com/xdmp/privileges/xdmp-default-in-memory-list-size
xdmp:default-in-memory-range-index-size	http://marklogic.com/xdmp/privileges/xdmp-default-in-memory-range-index-size
xdmp:in-memory-tree-size	http://marklogic.com/xdmp/privileges/xdmp-in-memory-tree-size
xdmp:delete-cluster-config-file	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file
xdmp:delete-host-config-file	http://marklogic.com/xdmp/privileges/xdmp-delete-host-config-file
xdmp:directory	http://marklogic.com/xdmp/privileges/xdmp-directory
xdmp:disable-event	http://marklogic.com/xdmp/privileges/xdmp-disable-event
xdmp:email	http://marklogic.com/xdmp/privileges/xdmp-email
xdmp:email-address	http://marklogic.com/xdmp/privileges/xdmp-email-address
xdmp:enable-event	http://marklogic.com/xdmp/privileges/xdmp-enable-event
xdmp:expanded-tree-cache-size	http://marklogic.com/xdmp/privileges/xdmp-expanded-tree-cache-size
xdmp:expanded-tree-cache-partitions	http://marklogic.com/xdmp/privileges/xdmp-expanded-tree-cache-partitions
xdmp:forest-backup	http://marklogic.com/xdmp/privileges/xdmp-forest-backup
xdmp:forest-clear	http://marklogic.com/xdmp/privileges/xdmp-forest-clear
xdmp:forest-delete	http://marklogic.com/xdmp/privileges/xdmp-forest-delete
xdmp:forest-restore	http://marklogic.com/xdmp/privileges/xdmp-forest-restore
xdmp:forest-status	http://marklogic.com/xdmp/privileges/xdmp-forest-status
xdmp:forest-keys	http://marklogic.com/xdmp/privileges/xdmp-forest-keys
xdmp:get-hot-updates	http://marklogic.com/xdmp/privileges/xdmp-get-hot-updates
xdmp:host-name	http://marklogic.com/xdmp/privileges/xdmp-hostname
xdmp:license-accepted	http://marklogic.com/xdmp/privileges/xdmp-license-accepted
xdmp:list-cache-size	http://marklogic.com/xdmp/privileges/xdmp-list-cache-size
xdmp:list-cache-partitions	http://marklogic.com/xdmp/privileges/xdmp-list-cache-partitions

Name	Action URI
xdmp:pre-release-expires	http://marklogic.com/xdmp/privileges/xdmp-pre-release-expires
xdmp:read-cluster-config-file	http://marklogic.com/xdmp/privileges/xdmp-read-cluster-config-file
xdmp:read-host-config-file	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file
xdmp:restart	http://marklogic.com/xdmp/privileges/xdmp-restart
xdmp:server-backup	http://marklogic.com/xdmp/privileges/xdmp-server-backup
xdmp:server-import-qualities	http://marklogic.com/xdmp/privileges/xdmp-server-import-qualities
xdmp:server-restore	http://marklogic.com/xdmp/privileges/xdmp-server-restore
xdmp:set-hot-updates	http://marklogic.com/xdmp/privileges/xdmp-set-hot-updates
xdmp:shutdown	http://marklogic.com/xdmp/privileges/xdmp-shutdown
xdmp:smtp-relay	http://marklogic.com/xdmp/privileges/xdmp-smtp-relay
xdmp:user-last-login	http://marklogic.com/xdmp/privileges/xdmp-user-last-login
xdmp:username	http://marklogic.com/xdmp/privileges/xdmp-username
xdmp:write-cluster-config-file	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file
xdmp:write-host-config-file	http://marklogic.com/xdmp/privileges/xdmp-write-host-config-file

There are no default permissions associated with the `admin-builtins` role.

33.3. admin-configuration-delete

The `admin-configuration-delete` role enables administrator users to delete configuration information.

33.4. admin-configuration-read

The `admin-configuration-read` role enables administrator users to read configuration information.

33.5. admin-configuration-write

The `admin-configuration-write` role enables administrator users to write configuration information.

33.6. admin-default

The `admin-default` role enables administrator users to evaluate administration default expressions.

33.7. admin-default-internal

The `admin-default-internal` role enables administrator users to invoke administration default expressions.

33.8. admin-module-internal

The `admin-module-read-internal` role is used internally by the Admin Library Module. Do not assign this role to any user. For details, see [Scripting Administrative Tasks in MarkLogic Server](#) in the *Scripting Administrative Tasks Guide*.

33.9. admin-module-read-internal

The `admin-module-read-internal` role is used internally by the Admin Library Module for reading. Do not assign this role to any user. For details, see [Scripting Administrative Tasks in MarkLogic Server](#) in the *Scripting Administrative Tasks Guide*.

33.10. admin-module-read-internal

The `admin-module-read-internal` role is used internally by the Admin Library Module for invoking functions with granular privileges. Do not assign this role to any user. For details, see [Scripting Administrative Tasks in MarkLogic Server](#) in the *Scripting Administrative Tasks Guide*.

33.11. admin-transform

The `admin-transform` role enables administrator users to evaluate transformations within the Admin API.

33.12. admin-ui-user

The `admin-ui-user` role enables users to have a read-only view of the Admin Interface, without providing access to data, to security configuration, or to write access to server configuration.

33.13. alert-admin

The `alert-admin` role is used for administrators of an alerting application. For details, see the [Creating Alerting Applications](#) in the *Search Developer's Guide*.

33.14. alert-execution

The `alert-execution` role is used internally by the Alerting API to amp privileges in a protected way. Do not give this role to any individual users. For details, see the [Creating Alerting Applications](#) in the *Search Developer's Guide*.

33.15. alert-internal

The `alert-internal` role is used internally by the Alerting API to amp privileges in a protected way. You should not give this role to any individual users. For details, see the [Creating Alerting Applications](#) in the *Search Developer's Guide*.

33.16. alert-user

The `alert-user` role is used by users of an alerting application. For details, see the [Creating Alerting Applications](#) in the *Search Developer's Guide*.

33.17. app-builder

The `app-builder` role provides the privileges needed to run Application Builder. Application Builder is no longer a part of MarkLogic. This role exists only for backward compatibility.

33.18. app-builder-internal

Application Builder is no longer a part of MarkLogic. This role exists only for backward compatibility.

33.19. app-user

The `app-user` role is a minimally privileged role that is needed to run any application that Application Builder generates. Application Builder is no longer a part of MarkLogic. This role exists only for backward compatibility.

33.20. application-plugin-registrar

The `application-plugin-registrar` role is used in the plugin API, and has the following execute privileges:

Name	Action URI
plugin-server-fields	http://marklogic.com/xdmp/privileges/plugin-server-fields
plugin-register	http://marklogic.com/xdmp/privileges/plugin-register
xdmp.filesystem-directory	http://marklogic.com/xdmp/privileges/xdmp-filesystem-directory
xdmp:get-server-field	http://marklogic.com/xdmp/privileges/xdmp-get-server-field
xdmp:get-server-field-names	http://marklogic.com/xdmp/privileges/xdmp-get-server-field-names
xdmp:invoke-modules-change-file	http://marklogic.com/xdmp/privileges/xdmp-invoke-modules-change-file
xdmp:set-server-field	http://marklogic.com/xdmp/privileges/xdmp-set-server-field
xdmp:set-server-field-privilege	http://marklogic.com/xdmp/privileges/xdmp-set-server-field-privilege

33.21. appservices-internal

The `appservices-internal` role is used by Application Services to amp certain functions that Application Services performs. You should not explicitly grant the `appservices-internal` role to any user; it is only for internal use by Application Services.

33.22. cpf-restart

The `cpf-restart` role is used by CPF to control access to the CPF restart trigger. The CPF restart user should have the `cpf-restart` role, as well as all of the permissions and privileges that normal users have on the documents.

33.23. custom-dictionary-admin

The `custom-dictionary-admin` role is for performing administrative functions (for writing dictionaries in the configuration) in the custom dictionary API.

33.24. custom-dictionary-user

The `custom-dictionary-user` role is for performing user functions (for reading dictionaries in the configuration) in the custom dictionary API.

33.25. custom-language-admin-read

The `custom-language-admin-read` role enables a user to read custom language configurations. That is, to use functions such as `clang:language-config-read`.

33.26. custom-language-admin-write

The `custom-language-admin-write` role enables a user to modify custom language configurations. That is, to use functions such as `clang:language-config-write` and `clang-language-config-delete`. These operations change the cluster configuration file and cause a cluster-wide restart when used.

33.27. dls-admin

The `dls-admin` role is designed to give administrators of Library Services applications all of the privileges that are needed to use the Library Services API. It has the needed privileges to perform operations such as inserting retention policies and breaking checkouts, so only trusted users (users who are assumed to be non-hostile, appropriately trained, and follow proper administrative procedures) should be granted the `dls-admin` role. Assign the `dls-admin` role to administrators of your Library Services application.

For details, see [Library Services Applications](#) in the *Application Developer's Guide*.

33.28. dls-internal

The `dls-internal` role is a role that is used internally by the Library Services API, but you should not explicitly grant it to any user or role. This role is used to amp special privileges within the context of certain functions of the Library Services API. Assigning this role to users would give them privileges on the system that you typically do not want them to have; do not assign this role to any users.

For details, see [Library Services Applications](#) in the *Application Developer's Guide*.

33.29. dls-user

The `dls-user` role is a minimally privileged role. It is used in the Library Services API to allow regular users of the Library Services application (as opposed to `dls-admin` users) to be able to execute code in the Library Services API. It allows users, with document update permission, to manage, checkout, and checkin managed documents.

The `dls-user` role only has privileges that are needed to run the Library Services API; it does not provide execute privileges to any functions outside the scope of the Library Services API. The

Library Services API uses the `dls-user` role as a mechanism to amp more privileged operations in a controlled way. It is therefore reasonably safe to assign this role to any user whom you trust to use your Library Services application. Assign the `dls-user` role to all users of your Library Services application.

For details, see [Library Services Applications](#) in the *Application Developer's Guide*.

33.30. domain-management

The `domain-management` role has the privileges to create and modify content processing domains. The `domain-management` role has no execute privileges associated with it, but it has the following default permissions:

Role	Capability
<code>domain-management</code>	Read
<code>domain-management</code>	Update

33.31. filesystem-access

The `filesystem-access` role has several execute privileges to access the file system:

Name	Action URI
<code>xdmp:document-get</code>	<code>http://marklogic.com/xdmp/privileges/xdmp-document-get</code>
<code>xdmp:document-load</code>	<code>http://marklogic.com/xdmp/privileges/xdmp-document-load</code>
<code>xdmp:get</code>	<code>http://marklogic.com/xdmp/privileges/xdmp-get</code>
<code>xdmp:load</code>	<code>http://marklogic.com/xdmp/privileges/xdmp-load</code>
<code>xdmp:save</code>	<code>http://marklogic.com/xdmp/privileges/xdmp-save</code>

There are no default permissions associated with the `filesystem-access` role.

33.32. flexrep-admin

The `flexrep-admin` role is required to configure replication.

33.33. flexrep-eval

This is an internal role used by the flexible replication feature.

33.34. flexrep-internal

The `flexrep-internal` role is used by Flexible Replication to amp certain functions that Flexible Replication performs. You should not explicitly grant the `flexrep-internal` role to any user; it is only for internal use by Flexible Replication.

33.35. flexrep-user

The `flexrep-user` role user is required to access the Replica App Server when configured for push replication and the Master App Server when configured for pull replication. The replication user must be given the `flexrep-user` role and have the privileges necessary to update the domain content.

33.36. flexrep-user-change

This is an internal role used by the flexible replication feature that can do an explicit push/pull.

33.37. graphql-internal

GraphQL internal role.

33.38. hadoop-internal

The `hadoop-internal` role is for internal use only. Do not assign this role to any users. This role is used to amp special privileges within the context of certain functions of the Hadoop MapReduce

Connector. Assigning this role to users would give them privileges on the system that you typically do not want them to have.

33.39. hadoop-user-all

The `hadoop-user-all` role combines the privileges of `hadoop-user-read` and `hadoop-user-write`.

33.40. hadoop-user-read

The `hadoop-user-read` role allows use of MarkLogic Server as an input source for a MapReduce job. This role does not grant any other privileges, so the `mapreduce.marklogic.input.user` may still require additional privileges to read content from the target database. The `hadoop-user-read` role has the following execute privileges:

Name	Action URI
<code>hadoop-user-read</code>	<code>http://marklogic.com/xdmp/privileges/hadoop-user-read</code>
<code>xdbc:eval</code>	<code>http://marklogic.com/xdmp/privileges/xdbc-eval</code>
<code>xdbc:eval-in</code>	<code>http://marklogic.com/xdmp/privileges/xdbc-eval-in</code>
<code>xdmp:value</code>	<code>http://marklogic.com/xdmp/privileges/xdmp-value</code>
<code>xdmp:with-namespaces</code>	<code>http://marklogic.com/xdmp/privileges/xdmp-with-namespace</code>

33.41. hadoop-user-write

The `hadoop-user-write` role allows use of MarkLogic Server as an output destination for a MapReduce job. This role does not grant any other privileges, so the `mapreduce.marklogic.output.user` may still require additional privileges to insert or update content in the target database. The `hadoop-user-write` role has the following execute privileges:

Name	Action URI
<code>any-uri</code>	<code>http://marklogic.com/xdmp/privileges/any-uri</code>
<code>hadoop-user-write</code>	<code>http://marklogic.com/xdmp/privileges/hadoop-user-write</code>
<code>unprotected-collections</code>	<code>http://marklogic.com/xdmp/privileges/unprotected-collections</code>
<code>xdbc:eval</code>	<code>http://marklogic.com/xdmp/privileges/xdbc-eval</code>
<code>xdbc:insert-in</code>	<code>http://marklogic.com/xdmp/privileges/xdbc-insert-in</code>
<code>xdmp:with-namespaces</code>	<code>http://marklogic.com/xdmp/privileges/xdmp-with-namespace</code>

33.42. harmonized-reader

A role that by default allows for reading harmonized documents in a data hub.

33.43. harmonized-updater

Provides update permissions to harmonized documents in a data hub.

33.44. healthcheck-user

This is an internal role used for the HealthCheck App Server on port 7997 for application-level authentication. For more details on application-level authentication, please refer to [Application Level](#).

33.45. infostudio-admin-internal

Information Studio is no longer a part of MarkLogic. This role exists only for backward compatibility.

The `infostudio-admin-user` role provides the privileges needed to handle CPF restart and resume unfinished Information Studio tasks in the event of an unexpected shutdown and restart of MarkLogic Server. When MarkLogic Server is restarted, long-running collectors resume loading documents in the database. In this situation, the original user that started the collector is unknown, so the purpose of the `infostudio-admin user` is to resume control of the collector.

33.46. infostudio-internal

Information Studio is no longer a part of MarkLogic. This role exists only for backward compatibility.

The `infostudio-user` role is used by Information Studio to amp certain functions that Information Studio performs. You should not explicitly grant the `infostudio-internal` role to any user; it is only for internal use by Information Studio.

33.47. infostudio-user

Information Studio is no longer a part of MarkLogic. This role exists only for backward compatibility.

The `infostudio-user` role is a minimally privileged role that is needed to use Information Studio. You must grant this role to all users who are allowed to access Information Studio.

The `infostudio-user` role has the following execute privileges:

- `infostudio` (<http://marklogic.com/xdmp/privileges/infostudio>)
- `unprotected-collections`

33.48. manage

The `manage` role has the execute privilege <http://marklogic.com/xdmp/privileges/manage> to run the Management API. For example, non-admin users can use `manage` role plus `create-data-role` or `create-data-user` granular privileges to manage roles and create data users.

Name	Action URI
<code>manage</code>	http://marklogic.com/xdmp/privileges/manage

There are no default permissions associated with the `manage` role.

33.49. manage-admin

The `manage-admin` role has the privileges related to accessing the management API and the tiered storage API for operations that change the configuration. The table provides the execute privileges given to the `manage-admin` role:

Name	Action URI
<code>manage</code>	http://marklogic.com/xdmp/privileges/manage
<code>manage-admin</code>	http://marklogic.com/xdmp/privileges/manage-admin
<code>ts:database-create-sub-database</code>	http://marklogic.com/xdmp/privileges/database-create-sub-database
<code>ts:database-create-super-database</code>	http://marklogic.com/xdmp/privileges/database-create-super-database
<code>ts:database-delete-sub-database</code>	http://marklogic.com/xdmp/privileges/database-delete-sub-database
<code>ts:database-delete-super-database</code>	http://marklogic.com/xdmp/privileges/database-delete-super-database
<code>ts:database-partitions</code>	http://marklogic.com/xdmp/privileges/database-partitions
<code>ts:forest-combine</code>	http://marklogic.com/xdmp/privileges/forest-combine
<code>ts:forest-migrate</code>	http://marklogic.com/xdmp/privileges/forest-migrate
<code>ts:partition-create</code>	http://marklogic.com/xdmp/privileges/partition-create
<code>ts:partition-delete</code>	http://marklogic.com/xdmp/privileges/partition-delete
<code>ts:partition-forests</code>	http://marklogic.com/xdmp/privileges/partition-forests
<code>ts:partition-migrate</code>	http://marklogic.com/xdmp/privileges/partition-migrate
<code>ts:partition-resize</code>	http://marklogic.com/xdmp/privileges/partition-resize
<code>ts:partition-set-availability</code>	http://marklogic.com/xdmp/privileges/partition-set-availability
<code>ts:partition-set-updates-allowed</code>	http://marklogic.com/xdmp/privileges/partition-set-updates-allowed
<code>ts:partition-transfer</code>	http://marklogic.com/xdmp/privileges/partition-transfer

There are no default permissions associated with the `manage-admin` role.

33.50. manage-admin-internal

The `manage-admin-internal` role is used to amp certain functions used by the Management API. You should not explicitly grant the `manage-admin-internal` role to any user. It is only for internal use.

33.51. manage-internal

The `manage-internal` role is used to amp certain functions used by the Management API. You should not explicitly grant the `manage-internal` role to any user. It is only for internal use.

33.52. manage-schematron-user

This role is used to manage [schematron functions](#).

33.53. manage-user

The `manage-user` role has the privileges related to accessing the Management API. These are the execute privileges given to the `manage-user` role:

Name	Action URI
manage	http://marklogic.com/xdmp/privileges/manage

There are no default permissions associated with the `manage-user` role.

33.54. merge

The `merge` role has the privileges related to forest merging. These are the execute privileges given to the `merge` role:

Name	Action URI
xdmp:merge	http://marklogic.com/xdmp/privileges/xdmp-merge
xdmp:merging	http://marklogic.com/xdmp/privileges/xdmp-merging

There are no default permissions associated with the `merge` role.

33.55. network-access

The `network-access` role has the privileges to run the `xdmp:http-*` functions (`xdmp:http-get`, `xdmp:http-post`, and so on). These are the execute privileges given to the `network-access` role:

Name	Action URI
xdmp:http-get	http://marklogic.com/xdmp/privileges/xdmp-http-get
xdmp:http-head	http://marklogic.com/xdmp/privileges/xdmp-http-head
xdmp:http-options	http://marklogic.com/xdmp/privileges/xdmp-http-options
xdmp:http-delete	http://marklogic.com/xdmp/privileges/xdmp-http-delete
xdmp:http-post	http://marklogic.com/xdmp/privileges/xdmp-http-post
xdmp:http-put	http://marklogic.com/xdmp/privileges/xdmp-http-put

33.56. optic-reader-internal

The `optic-reader-internal` role is an internal role. It allows the user to execute Optic queries. It amps the Optic API execute functions. This is mainly used by other (internal) roles such as `rest-reader-internal` which is used by the REST API.

33.57. ort-user

ONNX runtime user role. See [Machine Learning with the ONNX API](#) and [Security](#) for additional information.

33.58. pii-reader

The `pii-reader` role entitles users to see data defined as Personal Identifiable Information (PII). This role is used in conjunction with [Element Level Security](#), which is implemented through either [Protected Paths](#) or [Query Rolesets](#).

This role is also used by MarkLogic Data Hub in its [PII Security Configuration Files](#).

33.59. pipeline-execution

The `pipeline-execution` role is used in the XQuery code to allow any user (who can write a document to the domain) to execute code in the pipeline.

For details, see the [Content Processing Framework Guide](#).

33.60. pipeline-management

The `pipeline-management` role has the privileges to create and modify content processing pipelines. The `pipeline-management` role has no execute privileges associated with it, but it has these default permissions:

Role	Capability
<code>pipeline-management</code>	Read
<code>pipeline-management</code>	Update

33.61. pki

The `pki` role has the privileges to use the PKI Library functions. For details, see [Configuring SSL on App Servers](#) in *Securing MarkLogic Server*.

33.62. plugin-internal

The `plugin-user` role is used to amp certain functions associated with plugins. You should not explicitly grant the `plugin-internal` role to any user; it is only for internal use by the plugin API.

33.63. ps-internal

Internal, base role for provenance records stored in datahub-JOBS database as part of the Data Hub Framework. For more information, see [Provenance and Lineage](#).

33.64. ps-user

Base role for reading provenance records stored in datahub-JOBS database as part of the Data Hub Framework.

33.65. qconsole-internal

The `qconsole-internal` role is used by Query Console to amp certain functions that Query Console performs. You should not explicitly grant the `qconsole-internal` role to any user; it is only for internal use by Query Console.

33.66. qconsole-user

The `qconsole-user` role is a minimally privileged role that is needed to use Query Console. You must grant this role to all users who are allowed to use Query Console.

The `qconsole-user` role has these execute privileges:

- `qconsole` (<http://marklogic.com/xdmp/privileges/qconsole>)

33.67. query-view-admin

Query-based view administration. Base role for creating TDE views.

33.68. redaction-internal

Internal role for the redaction feature.

33.69. redaction-user

User role for the redaction feature. See [rdt functions](#) and [Redacting Document Content](#) for more information.

33.70. rest-admin

The `rest-admin` role has the `rest-writer` and `manage-user` roles and allows those granted the role full access to read and write via the REST API.

33.71. rest-admin-internal

The `rest-admin-internal` role is used internally by the REST Library. You should not explicitly grant it to any user or role.

33.72. rest-extension-user

The `rest-extension-user` role enables access to resource service extension methods.

33.73. rest-internal

The `rest-internal` role is used internally by the REST Library. You should not explicitly grant it to any user or role.

33.74. rest-reader

The `rest-reader` role enables read operations through the MarkLogic REST API, such as retrieving documents and metadata.

33.75. rest-reader-internal

The `rest-writer-internal` role is used internally by the REST Library. You should not explicitly grant it to any user or role.

33.76. rest-writer

The `rest-writer` role enables write operations through the MarkLogic REST API, such as creating documents, metadata, or configuration information.

33.77. rest-writer-internal

The `rest-reader-internal` role is used internally by the REST Library. You should not explicitly grant it to any user or role.

33.78. search-internal

The `search-internal` role is a role that is used internally by the search API. You should not explicitly grant it to any user or role.

33.79. security

The `security` role has the privileges needed to perform security functions. These are execute privileges given to the `security` role:

Name	Action URI
amp-add-roles	http://marklogic.com/xdmp/privileges/amp-add-roles
amp-get-roles	http://marklogic.com/xdmp/privileges/amp-get-roles
amp-remove-roles	http://marklogic.com/xdmp/privileges/amp-remove-roles
amp-set-roles	http://marklogic.com/xdmp/privileges/amp-set-roles
any-collection	http://marklogic.com/xdmp/privileges/any-collection
any-uri	http://marklogic.com/xdmp/privileges/any-uri
collection-add-permissions	http://marklogic.com/xdmp/privileges/collection-add-permissions
collection-get-permissions	http://marklogic.com/xdmp/privileges/collection-get-permissions
collection-remove-permissions	http://marklogic.com/xdmp/privileges/collection-remove-permissions
collection-set-permissions	http://marklogic.com/xdmp/privileges/collection-set-permissions
create-amp	http://marklogic.com/xdmp/privileges/create-amp
create-privilege	http://marklogic.com/xdmp/privileges/create-privilege
create-role	http://marklogic.com/xdmp/privileges/create-role
create-user	http://marklogic.com/xdmp/privileges/create-user
get-amp	http://marklogic.com/xdmp/privileges/get-amp
get-privilege	http://marklogic.com/xdmp/privileges/get-privilege
get-role-ids	http://marklogic.com/xdmp/privileges/get-role-ids
grant-all-roles	http://marklogic.com/xdmp/privileges/grant-all-roles
grant-my-roles	http://marklogic.com/xdmp/privileges/grant-my-roles
permission	http://marklogic.com/xdmp/privileges/permission
privilege-add-roles	http://marklogic.com/xdmp/privileges/privilege-add-roles
privilege-get-roles	http://marklogic.com/xdmp/privileges/privilege-get-roles
privilege-remove-roles	http://marklogic.com/xdmp/privileges/privilege-remove-roles
privilege-set-name	http://marklogic.com/xdmp/privileges/privilege-set-name
privilege-set-roles	http://marklogic.com/xdmp/privileges/privilege-set-roles
protect-collection	http://marklogic.com/xdmp/privileges/protect-collection
remove-amp	http://marklogic.com/xdmp/privileges/remove-amp
remove-privilege	http://marklogic.com/xdmp/privileges/remove-privilege
remove-role	http://marklogic.com/xdmp/privileges/remove-role
remove-role-from-amps	http://marklogic.com/xdmp/privileges/remove-role-from-amps
remove-role-from-privileges	http://marklogic.com/xdmp/privileges/remove-role-from-privileges
remove-role-from-roles	http://marklogic.com/xdmp/privileges/remove-role-from-roles
remove-role-from-users	http://marklogic.com/xdmp/privileges/remove-role-from-users
remove-user	http://marklogic.com/xdmp/privileges/remove-user
role-add-roles	http://marklogic.com/xdmp/privileges/role-add-roles
role-get-default-collections	http://marklogic.com/xdmp/privileges/role-get-default-collections
role-get-default-permissions	http://marklogic.com/xdmp/privileges/role-get-default-permissions
role-get-roles	http://marklogic.com/xdmp/privileges/role-get-roles
role-privileges	http://marklogic.com/xdmp/privileges/role-privileges
role-remove-roles	http://marklogic.com/xdmp/privileges/role-remove-roles
role-set-default-collections	http://marklogic.com/xdmp/privileges/role-set-default-collections
role-set-default-permissions	http://marklogic.com/xdmp/privileges/role-set-default-permissions
role-set-description	http://marklogic.com/xdmp/privileges/role-set-description
role-set-name	http://marklogic.com/xdmp/privileges/role-set-name
role-set-roles	http://marklogic.com/xdmp/privileges/role-set-roles
unprotect-collection	http://marklogic.com/xdmp/privileges/unprotect-collection
user-add-roles	http://marklogic.com/xdmp/privileges/user-add-roles
user-get-default-collections	http://marklogic.com/xdmp/privileges/user-gt-default-collections
user-get-default-permissions	http://marklogic.com/xdmp/privileges/user-get-default-permissions
user-get-description	http://marklogic.com/xdmp/privileges/user-get-description

Name	Action URI
user-get-roles	http://marklogic.com/xdmp/privileges/user-get-roles
user-privileges	http://marklogic.com/xdmp/privileges/user-privileges
user-remove-roles	http://marklogic.com/xdmp/privileges/user-remove-roles
user-set-default-collections	http://marklogic.com/xdmp/privileges/user-set-default-collections
user-set-default-permissions	http://marklogic.com/xdmp/privileges/user-set-default-permissions
user-set-description	http://marklogic.com/xdmp/privileges/user-set-description
user-set-name	http://marklogic.com/xdmp/privileges/user-set-name
user-set-password	http://marklogic.com/xdmp/privileges/user-set-password
user-set-roles	http://marklogic.com/xdmp/privileges/user-set-roles
xdmp:amp-roles	http://marklogic.com/xdmp/privileges/xdmp:amp-roles
xdmp:privilege-roles	http://marklogic.com/xdmp/privileges/xdmp:privilege-roles
xdmp:role-roles	http://marklogic.com/xdmp/privileges/xdmp:role-roles
xdmp:user-roles	http://marklogic.com/xdmp/privileges/xdmp:user-roles

These are the default permissions for the `security` role:

Role	Capability
security	Read
security	Insert
security	Update

33.80. security-internal

[v11.2.0 and up]

The `security-internal` role is a role that is used internally to execute internal security functions needed to configure security configurations and execute necessary security actions. You should not explicitly grant it to any user or role.

33.81. sparql-update-user

This is a base role to manage triples using [sem:sparql-update](#).

33.82. sql-execution

SQL execution role.

33.83. tde-admin

The `tde-admin` role has the privileges to administer extraction templates.

33.84. tde-view

The `tde-view` role has the privileges to view extraction templates.

33.85. temporal-admin

The `temporal-admin` role has the privileges to create and modify temporal data.

33.86. temporal-internal

The `temporal-internal` role is an internal role. Do not assign this role to any user.

33.87. tiered-storage-admin

Base role to configure tiered storage.

33.88. tiered-storage-internal

Tiered storage internal role.

33.89. trigger-management

The `trigger-management` role has the privileges to create and modify triggers. The `trigger-management` role has no execute privileges associated with it. This role has these default permissions:

Role	Capability
<code>trigger-management</code>	Read
<code>trigger-management</code>	Update

33.90. view-admin

The `view-admin` role enables a user to view MarkLogic Server administration.

33.91. view-admin-internal

The `view-admin-internal` role is used internally by the MarkLogic Server. Do not explicitly grant it to any user or role.

33.92. welcome-internal

The `welcome-internal` role is a role that used to be used internally by the MarkLogic Server Welcome Page (now removed). Do not explicitly grant it to any user or role.

33.93. xa

The `xa` user role allows creation and management of one's own XA transaction branches

in MarkLogic Server. The `xa` role is required to participate in XA transactions. For details, see [Participating in XA Transactions](#) in *Developing with XCC*. The `xa` role has these execute privileges:

Name	Action URI
<code>complete-my-transaction</code>	<code>http://marklogic.com/xdmp/privileges/complete-my-transactions</code>
<code>forget-my-xa-transactions</code>	<code>http://marklogic.com/xdmp/privileges/forget-my-xa-transactions</code>
<code>prepare-my-xa-transactions</code>	<code>http://marklogic.com/xdmp/privileges/prepare-my-xa-transactions</code>
<code>status-builtins</code>	<code>http://marklogic.com/xdmp/privileges/status-builtins</code>
<code>xdmp:set-current-transaction</code>	<code>http://marklogic.com/xdmp/privileges/set-current-transaction</code>
<code>xdmp:transaction-create</code>	<code>http://marklogic.com/xdmp/privileges/xdmp-transaction-create</code>
<code>xdmp:transaction-create-xid</code>	<code>http://marklogic.com/xdmp/privileges/xdmp-transaction-create-xid</code>

33.94. xa-admin

The `xa-admin` role allows creation and manage of any user's XA transaction branches in

MarkLogic Server. The `xa-admin` role is intended primarily for Administrators who need to complete or forget XA transactions. The `xa-admin` role has these execute privileges:

Name	Action URI
<code>complete-any-transactions</code>	<code>http://marklogic.com/xdmp/privileges/complete-any-transactions</code>
<code>complete-my-transaction</code>	<code>http://marklogic.com/xdmp/privileges/complete-my-transactions</code>
<code>forget-any-xa-transactions</code>	<code>http://marklogic.com/xdmp/privileges/forget-any-xa-transactions</code>
<code>forget-my-xa-transactions</code>	<code>http://marklogic.com/xdmp/privileges/forget-my-xa-transactions</code>
<code>prepare-any-xa-transactions</code>	<code>http://marklogic.com/xdmp/privileges/prepare-any-xa-transactions</code>
<code>prepare-my-xa-transactions</code>	<code>http://marklogic.com/xdmp/privileges/prepare-my-xa-transactions</code>
<code>status-builtins</code>	<code>http://marklogic.com/xdmp/privileges/status-builtins</code>

Name	Action URI
xdmp:set-current-transaction	http://marklogic.com/xdmp/privileges/set-current-transaction
xdmp:transaction-create	http://marklogic.com/xdmp/privileges/xdmp-transaction-create
xdmp:transaction-create-xid	http://marklogic.com/xdmp/privileges/xdmp-transaction-create-xid

33.95. xinclude

The `xinclude` role provides the privileges to run the XInclude code used in the XInclude CPF application. For details, see [Reusing Content With Modular Document Applications](#) in the *Application Developer's Guide*.

34. Technical support

Progress Software provides technical support according to the terms detailed in your Software License Agreement or End User License Agreement.

We invite you to visit our support website at <http://help.marklogic.com> to access information on known and fixed issues, knowledge base articles, and more. For licensed customers with an active maintenance contract, see the [Support Handbook](#) for instructions on registering support contacts and on working with the MarkLogic Server Technical Support team.

Complete product documentation, the latest product release downloads, and other useful information is available for all developers at <http://developer.marklogic.com>. For technical questions, we encourage you to ask your question on [Stack Overflow](#).

35. Copyright

For copyright information, see [Product Documentation and Copyright Notice](#).