MarkLogic Server

Administrator's Guide

MarkLogic 8
February, 2015

Last Revised: 8.0-4, November, 2015

Copyright © 2015 MarkLogic Corporation. All rights reserved.

Table of Contents

Administrator's Guide

1.0 Introduction 15
1.1 Objectives 15
1.2 Audience 15
1.3 Scope and Requirements 15
1.4 Architecture Overview 16

2.0 Administrative Interface 18
2.1 Overview of the Admin Interface 18
2.2 Accessing the Admin Interface 19
2.3 Logging Off the Admin Interface 19
2.4 Creating and Managing Administrators 19

3.0 Common Administrative Procedures 20
3.1 Installing and Upgrading MarkLogic Server 20
3.2 Starting and Stopping MarkLogic Server 20
3.3 Creating and Configuring Forests and Databases 21
3.4 Creating and Configuring App Servers 21
3.5 Setting up Users, Roles, Privileges, and Permissions 22
3.6 Loading Content into a Database 22
3.7 Running The XQuery Use Cases and Building Simple Applications 22
3.8 Backing up and Restoring Data 22
3.9 Monitoring and Tuning Performance 23
3.10 Scripting and Scheduling Administrative Tasks 24
3.11 Configuring Clusters, Groups and Failover 24

4.0 Starting and Stopping MarkLogic Server 25
4.1 Starting the Server 25
4.2 Stopping the Server 25
4.2.1 Using System Command to Stop MarkLogic Server 26
4.2.2 Using the Admin Interface to Stop MarkLogic Server 26
4.3 Restarting the Server 27
4.4 Example XQuery Scripts 27
4.4.1 Script that Restarts MarkLogic Server 27
4.4.2 Script that Stops MarkLogic Server 27

5.0 Clusters 28
5.1 Overview of Cluster Configuration 28

5.2 OpenSSL FIPS 140-2 Mode 28
5.3 Procedures for Configuring Clusters 28
5.3.1 Configuring OpenSSL FIPS 140-2 Mode 29
5.3.2 Coupling Clusters 29

6.0 Groups 30
6.1 Overview of Groups 30
6.2 Example 31
6.3 Procedures for Configuring and Managing Groups 32
6.3.1 Creating a New Group 32
6.3.2 Viewing Group Settings 33
6.3.3 Deleting a Group 34
6.3.4 Enabling SSL communication over XDQP 34
6.3.5 Configuring an SMTP Server 36
6.3.6 Restarting All Hosts in a Group 36

7.0 HTTP Servers 37
7.1 HTTP Server Overview 37
7.2 Procedures for Creating and Managing HTTP Servers 38
7.2.1 Creating a New HTTP Server 38
7.2.2 Setting Output Options for an HTTP Server 42
7.2.3 Viewing HTTP Server Settings 43
7.2.4 Deleting an HTTP Server 43
7.2.5 Canceling a Request 44

8.0 XDBC Servers 46
8.1 XDBC Server Overview 46
8.2 Procedures for Creating and Managing XDBC Servers 47
8.2.1 Creating a New XDBC Server 47
8.2.2 Setting Output Options for an XDBC Server 51
8.2.3 Viewing XDBC Server Settings 52
8.2.4 Deleting an XDBC Server 52

9.0 WebDAV Servers 53
9.1 WebDAV Server Overview 53
9.1.1 Accesses a Database for Read and Write, Not XQuery Execution 54
9.1.2 WebDAV Server Security 54
9.1.3 Directories 55
9.1.3.1 Automatic Directory Creation in a Database Settings 55
9.1.3.2 Properties and URIs of Directories 56
9.1.4 Server Root Directory 57
9.1.5 Documents in a WebDAV Server 58
9.2 Procedures for Creating and Managing WebDAV Servers 58
9.2.1 Creating a New WebDAV Server 58

9.2.2 Setting Output Options for a WebDAV Server 62
9.2.3 Viewing WebDAV Server Settings 62
9.2.4 Deleting a WebDAV Server 63
9.3 WebDAV Clients 63
9.3.1 Tested WebDAV Clients 64
9.3.2 General Steps to Connect to a Server 65
9.3.3 Steps to Connect to a Web Folder in Windows Explorer 65
9.4 Example: Setting Up a WebDAV Server to Add/Modify Documents Used By Another Server 66

10.0 ODBC Servers 68
10.1 ODBC Server Overview 68
10.2 Procedures for Creating and Managing ODBC Servers 69
10.2.1 Creating a New ODBC Server 70
10.2.2 Setting Output Options for an ODBC Server 74
10.2.3 Viewing ODBC Server Settings 75
10.2.4 Deleting an ODBC Server 75
10.2.5 Canceling a Request 76

11.0 Configuring SSL on App Servers 78
11.1 Understanding SSL 78
11.2 General Procedure for Setting up SSL for an App Server 80
11.3 Procedures for Enabling SSL on App Servers 81
11.3.1 Creating a Certificate Template 81
11.3.2 Enabling SSL for an App Server 83
11.4 Accessing an SSL-Enabled Server from a Browser or WebDAV Client 84
11.4.1 Creating a Security Exception in Internet Explorer 85
11.4.2 Creating a Security Exception in Google Chrome 86
11.4.3 Creating a Security Exception in Mozilla Firefox 88
11.4.3.1 What to do if you don't get an 'Or you can add an exception' Prompt 90
11.4.4 Importing a Self-Signed Certificate Authority into Windows 92
11.4.5 Importing a Self-Signed Certificate Authority into Mozilla Firefox 98
11.5 Procedures for Obtaining a Signed Certificate 99
11.5.1 Generating and Downloading Certificate Requests 99
11.5.2 Importing a Signed Certificate into MarkLogic Server 101
11.6 Viewing Trusted Certificate Authorities 102
11.7 Importing a Certificate Revocation List into MarkLogic Server 104
11.8 Deleting a Certificate Template 105

12.0 Auditing Events 106
12.1 Overview of Auditing 106
12.1.1 Audit Log Files 106
12.1.2 Restricting Audit Events 107
12.1.3 Audit Successful, Unsuccessful, or Both Types of Events 107

12.1.4 Enabled at the Group Level 108
12.2 Auditable Events 108
12.2.1 Audit Log Content 112
12.2.2 Sample Audit Logs 113
12.3 Configuring Auditing for a Group 114
12.3.1 Enabling Auditing for a Group 114
12.3.2 Disabling Auditing for a Group 114
12.3.3 Configuring Auditing to Audit Certain Events and Set Up Certain Restrictions 115

13.0 Managing User Sessions and Monitoring Login Attempts 116
13.1 Managing Concurrent User Sessions 116
13.1.1 Limiting Concurrent Requests with User Session Limits 116
13.1.2 Configuring User Concurrent Session Controls 116
13.2 Setting Request Blackouts on an App Server 117
13.2.1 Configuring Request Blackouts 117
13.2.2 Deleting Request Blackouts 118
13.3 Storing and Monitoring the Last User Login Attempt 118
13.3.1 Storing Last User Login Information in a Last-Login Database 118
13.3.2 Configuring User Login Monitoring 119
13.3.3 Displaying the Last Login Information for an App Server or for the Admin Interface 119

14.0 Databases 120
14.1 Understanding Databases 120
14.1.1 Schemas and Security Databases 121
14.1.2 Modules Database 121
14.1.3 Triggers Database 122
14.1.4 Database Settings 122
14.1.4.1 Basic Administrative Settings 122
14.1.4.2 Index Settings that Affect Documents 123
14.1.4.3 Rebalancer Settings 126
14.1.4.4 Reindexing Settings 126
14.1.4.5 Document and Directory Settings 127
14.1.4.6 Memory and Journal Settings 129
14.1.4.7 Other Settings 131
14.1.4.8 Merge Control Settings 132
14.1.5 Example of Databases in MarkLogic Server 133
14.2 Creating a New Database 134
14.3 Attaching and/or Detaching Forests to/from a Database 135
14.4 Viewing Database Settings 136
14.5 Loading Documents into a Database 136
14.6 Merging a Database 137
14.7 Reindexing a Database 138
14.8 Clearing a Database 139

14.9 Disabling a Database 140
14.10 Deleting a Database 141
14.11 Checking and Setting Permissions for a Document in a Database 142

15.0 Word Query Database Settings 143
15.1 Understanding the Word Query Configuration 143
15.1.1 Overview of Configuration Options 143
15.1.2 Understanding Which Elements are Included and Excluded 144
15.1.3 Adding a Weight to Boost or Lower the Relevance of an Included Element 146
15.1.4 Specifying An Attribute Value for an Included Element 147
15.1.5 Understanding the Index Option Configuration 147
15.2 Configuring Customized Word Query Settings 148

16.0 Fields Database Settings 152
16.1 Overview of Fields 152
16.2 Understanding Field Configurations 153
16.2.1 Overview of Field Configuration Options 153
16.2.2 Understanding What is Included and Excluded in a Field 154
16.2.2.1 Root Fields 154
16.2.2.2 Path Fields 154
16.2.2.3 How Field Settings Determine What is Included and Excluded ... 155
16.2.3 Adding a Weight to Boost or Lower the Relevance of an Included Element or Property 157
16.2.4 Specifying An Attribute Value for an Included or Excluded Element ...158
16.2.5 Understanding the Index Option Configuration 159
16.3 Field Word Lexicons and Field Value Lexicons 159
16.4 Configuring Fields 159
16.4.1 Configuring a New Field 160
16.4.2 Modifying an Existing Field 167
16.4.3 Creating a Range Index on a Field 168

17.0 Understanding and Controlling Database Merges 169
17.1 Overview of Merges: Merges are Good 169
17.1.1 Dynamic and Self-Tuning 169
17.1.2 What Happens During a Merge 170
17.1.3 Dangers of Disabling Merges 170
17.1.4 Merges Will Change Scores 171
17.2 Setting Merge Policy 171
17.2.1 Overview of the Merge Policy Controls 171
17.2.2 Description of Merge Parameters 173
17.3 Blackout Periods for Merges 175
17.3.1 Understanding Merge Blackouts 175
17.3.2 Configuring Merge Blackout Periods 175

17.3.3 Deleting Merge Blackout Periods 176
17.4 Merges and Point-In-Time Queries 177
17.5 Setting a Negative Merge Timestamp to Preserve Fragments For a Rolling Window of Time 177
17.6 Monitoring a Merge 178
17.6.1 Messages in the ErrorLog.txt File 178
17.6.2 Database Status Page 179
17.7 Explicit Merge Commands 179
17.7.1 Manually Initiating a Merge 179
17.7.2 Cancelling a Merge 180
17.8 Configuring Merge Policy Rules 180
17.8.1 Determine the Baseline for Your Merges 181
17.8.2 If You Want to Reduce the Number of 'Large' Merges 181
17.8.3 Other Solutions 184

18.0 Database Rebalancing 185
18.1 Overview of the Database Rebalancer 185
18.2 Rebalancer Trigger Events 187
18.3 Rebalancer Document Assignment Policies 187
18.3.1 Bucket Assignment Policy 187
18.3.2 Statistical Assignment Policy 189
18.3.3 Range Assignment Policy 190
18.3.4 Legacy Assignment Policy 192
18.3.5 Summary of Assignment Policies 193
18.4 How the Rebalancer Moves Documents 193
18.4.1 How Data is Moved when a Forest is Attached to the Database 194
18.4.2 How Data is Moved when a Forest is Retired from the Database 194
18.5 Configuring the Rebalancer on a Database 194
18.6 Configuring the Rebalancer on a Forest 196
18.7 Retiring a Forest from the Database 197
18.8 Checking the Rebalancer Status 199
18.9 How the Rebalancer Interacts with other Database and Forest Settings 200
18.9.1 Database Replication 200
18.9.2 Restoring a Database from a Backup 200
18.9.3 Tiered Storage 200
18.9.4 Fast Locking 201
18.9.5 Delete-only and Read-only Forests 201
18.10 Rebalancer Settings after Upgrading from an Earlier Release 202

19.0 Tiered Storage 203
19.1 Terms Used in this Chapter 204
19.2 Overview of Tiered Storage 205
19.3 Partitions, Partition Keys, and Partition Ranges 207
19.4 Configuring a Database to Participate in Tiered Storage 211
19.5 Defining a Partition Key 211

19.6 Creating Partitions 213
19.6.1 Creating a Partition with New Forests 213
19.6.2 Creating a Partition from Existing Forests 214
19.7 Overview of the Tiered Storage REST API 215
19.7.1 Asynchronous Operations 216
19.7.2 Privileges 216
19.7.3 /manage/v2/databases/{id|name}/partitions 216
19.7.4 /manage/v2/databases/{id|name}/partitions/{name} 217
19.7.5 /manage/v2/databases/{id|name}/partitions/{name}/properties 218
19.7.6 /manage/v2/databases/{id|$name}/sub-databases 218
19.7.7 /manage/v2/databases/{id|name}/sub-databases/{id|name} 219
19.7.8 /manage/v2/databases/{id|name}/super-databases 219
19.7.9 /manage/v2/databases/{id|name}/super-databases/{id|name} 220
19.7.10/manage/v2/forests 220
19.7.11/manage/v2/forests/{id|name} 221
19.7.12/manage/v2/forests/{id|name}/properties 222
19.8 Common Forest and Partition Operations 223
19.8.1 Viewing Partitions 223
19.8.2 Migrating Forests and Partitions 224
19.8.3 Resizing Partitions 226
19.8.4 Transferring Partitions between Databases 226
19.8.5 Combining Forests 227
19.8.6 Retiring Forests 228
19.8.7 Taking Forests and Partitions Online and Offline 228
19.8.8 Setting the Updates-allowed State on Partitions 229
19.8.9 Deleting Partitions 229
19.9 Sub-databases and Super-databases 230
19.9.1 Creating Sub-databases 230
19.9.2 Creating Super-databases 231
19.9.3 Viewing Super-databases and Sub-databases 231
19.10 Partitions with Forest-Level Failover 231

20.0 Backing Up and Restoring a Database 234
20.1 Backup and Restore Overview 234
20.1.1 Consistent, Database-Level Backup 235
20.1.2 Admin Interface 236
20.1.3 Backup and Restore Transactions 236
20.1.4 Backup Directory Structure 236
20.1.5 Phases of Backup or Restore Operation 239
20.1.5.1 Validation Phase 239
20.1.5.2 Copy Phase 240
20.1.5.3 Synchronization Phase 240
20.1.6 Notes about Backup and Restore Operations 240
20.2 Backing Up Databases with Journal Archiving 241
20.3 Incremental Backup 243
20.3.1 Incremental Backup of New Forest 243

20.4 Incremental Backup with Journal Archiving 244
20.5 Backing Up a Database 245
20.5.1 Backing Up a Database Immediately 245
20.5.2 Scheduling a Database Backup 248
20.6 Restoring a Database without Journal Archiving 251
20.7 Restoring Databases with Journal Archiving 254
20.7.1 Restoring to the Safe Timestamp 255
20.7.2 Restoring to a Specific Timestamp 257
20.7.3 Restoring Based on Sample Documents 257
20.8 Restoring from an Incremental Backup with Journal Archiving 258
20.9 Backing up and Restoring a Database Following Local Disk Failover 262

21.0 Hosts 266
21.1 Adding a Host to a Cluster 266
21.2 Changing the Group of the Host 267
21.3 Shutting Down or Restarting a Host 268
21.4 Clearing a Forest on a Host 268
21.5 Deleting a Forest on a Host 269
21.6 Leaving the Cluster 269
21.7 Changing the License Key For a Host 271
21.8 Rolling Back a Transaction 271

22.0 Forests 273
22.1 Understanding Forests 274
22.2 Creating a Forest 275
22.3 Making a Forest Delete-Only 279
22.4 Making a Forest Read-Only 280
22.5 Attaching and Detaching Forests Using the Forest Summary Page 282
22.6 Making Backups of a Forest 283
22.6.1 Backing Up a Forest 283
22.6.2 Scheduling a Forest Backup 285
22.7 Restoring a Forest 286
22.8 Rolling Back a Forest to a Point In Time 287
22.9 Merging a Forest 287
22.10 Clearing a Forest 287
22.11 Disabling a Forest 288
22.12 Deleting a Forest from a Host 289
22.13 Rolling Back a Prepared XA Transaction Branch 290

23.0 Security Administration 292
23.1 Security Entities 292
23.2 Users 295
23.2.1 Creating a User 295
23.2.2 Viewing a User Configuration 298
23.2.3 Modifying a User Configuration 299

23.2.4 Deleting a User 299
23.3 Roles 299
23.3.1 Creating a Role 301
23.3.2 Viewing a Role 303
23.3.3 Modifying a Role Configuration 304
23.3.4 Deleting a Role 304
23.4 Execute Privileges 304
23.4.1 Creating an Execute Privilege 305
23.4.2 Viewing an Execute Privilege 306
23.4.3 Modifying an Execute Privilege 307
23.4.4 Deleting an Execute Privilege 307
23.5 URI Privileges 308
23.5.1 Creating a URI Privilege 309
23.5.2 Viewing a URI Privilege 309
23.5.3 Modifying a URI Privilege 310
23.5.4 Deleting a URI Privilege 310
23.6 Amps 311
23.6.1 Creating an Amp 312
23.6.2 Viewing an Amp 313
23.6.3 Modifying an Amp 314
23.6.4 Deleting an Amp 314
23.7 Protected Collections 315
23.7.1 Creating a Protected Collection 316
23.7.2 Viewing a Protected Collection 317
23.7.3 Removing a Permission from a Protected Collection 317
23.7.4 Deleting a Protected Collection 318
23.8 Certificate Templates 318
23.9 Realm 319
23.9.1 Setting the Realm 319
23.9.2 Changing the Realm 320

24.0 Text Indexing 321
24.1 Text Indexes 321
24.1.1 Understanding the Text Index Settings 322
24.1.2 Viewing Text Index Configuration 328
24.1.3 Configuring Text Indexes 330
24.2 Phrasing and Element-Word-Query Boundary Control 331
24.2.1 Phrasing Control 331
24.2.2 Element Word Query Throughs 333
24.2.3 Procedures 333
24.2.3.1 Viewing Phrasing and Element-Word-Query Settings 333
24.2.3.2 Configuring Phrasing and Element-Word-Query Settings 334
24.2.3.3 Deleting a Phrasing or Element-Word-Query Setting 336
24.3 Query Behavior with Reindex Settings Enabled and Disabled 337
24.3.1 Understanding the Reindexer Enable Settings 337
24.3.2 Query Evaluation According to the Lowest Common Denominator 338

24.3.3 Reindexing Does Not Apply to Point-In-Time Versions of Fragments .338
24.3.4 Example Scenario 339

25.0 Range Indexes and Lexicons 340
25.1 Understanding Range Indexes 341
25.2 Using Range Indexes for Value Lexicons 345
25.3 Understanding Word Lexicons 345
25.4 Understanding Path Range Indexes 346
25.4.1 Examples of Index Path Expressions 346
25.4.2 Limitations on Index Path Expressions 347
25.4.3 Testing the Validity of an Index Path Expression 348
25.4.4 Using Namespace Prefixes in Index Path Expressions 348
25.4.5 Grammar for Index Path Expressions 349
25.5 Viewing Element Range Index Settings 350
25.6 Defining Element Range Indexes 351
25.7 Viewing Attribute Range Index Settings 353
25.8 Defining Attribute Range Indexes 354
25.9 Viewing Path Range Index Settings 356
25.10 Defining Namespace Prefixes Used in Path Range Indexes and Fields 356
25.11 Defining Path Range Indexes 357
25.12 Viewing Element Word Lexicon Settings 359
25.13 Defining Element Word Lexicons 360
25.14 Viewing Attribute Word Lexicon Settings 361
25.15 Defining Attribute Word Lexicons 361
25.16 Defining Value Lexicons 363
25.17 Deleting Range Indexes or Lexicons 364
25.18 Defining Field Range Indexes 364

26.0 Fragments 365
26.1 Choosing a Fragmentation Strategy 366
26.1.1 Fragment Roots 367
26.1.2 Fragment Parents 367
26.2 Defining Fragment Roots 368
26.3 Defining Fragment Parents 369
26.4 Viewing Fragment Rules 370
26.5 Deleting Fragment Rules 371

27.0 Namespaces 372
27.1 Defining Namespaces for a Group 373
27.2 Defining Namespaces for an HTTP, ODBC, or XDBC Server 374
27.3 Viewing Namespace Settings for a Group 375
27.4 Viewing Namespace Settings for an HTTP, ODBC, or XDBC Server 375
27.5 Deleting Namespaces for a Group 376
27.6 Deleting Namespaces for an HTTP, ODBC, or XDBC Server 377

28.0 Understanding and Defining Schemas 378
28.1 Understanding Schemas 378
28.2 Procedures For Defining Schemas 379
28.2.1 Adding a Schema Definition for a Group 379
28.2.2 Adding a Schema Definition for an HTTP, ODBC, or XDBC Server ... 381
28.2.3 Viewing Schema Definitions for a Group 382
28.2.4 Viewing Schema Definitions for an HTTP, ODBC, or XDBC Server ... 384
28.2.5 Deleting a Schema Definition for a Group 385
28.2.6 Deleting a Schema Definition for an HTTP, ODBC, or XDBC Server . 385

29.0 Log Files 386
29.1 Understanding the Log Levels 386
29.2 Configuring Log Files 387
29.3 Viewing the System Log 388
29.4 Viewing the File Log 389
29.5 Access Log Files 389

30.0 Scheduling Tasks 390
30.1 Understanding Scheduled Tasks 390
30.2 Scheduling a Module for Invocation 391
30.3 Selecting a Task Type 393
30.3.1 Scheduling Per Minute 393
30.3.2 Scheduling Per Hour 394
30.3.3 Scheduling Per Day and Time 395
30.3.4 Scheduling Per Week, Day, and Time 395
30.3.5 Scheduling Per Month, Day, and Time 396
30.3.6 Scheduling One Invocation on a Calendar Date and Time 396

31.0 Using the Configuration Manager 397
31.1 Configuration Manager Overview 397
31.2 Security Considerations 398
31.3 Accessing the Configuration Manager 398
31.4 Viewing Configurations 398
31.4.1 Browsing Resource Configurations 399
31.4.2 Searching for a Resource 402
31.5 Searching for a Configuration Setting 402
31.6 Editing Configuration Settings 403
31.7 Exporting and Importing Configurations 405
31.7.1 Exporting a Configuration 406
31.7.2 Importing a Configuration 408
31.7.3 Comparing Imported Configuration with Current Configuration 409
31.8 Applying Imported Configuration Settings 413

32.0 Appendix A: ‘Hot’ versus ‘Cold’ Admin Tasks 414

32.1 Groups 415
32.2 HTTP, ODBC, XDBC, and WebDAV Servers 416
32.3 Databases 416
32.4 Hosts 416
32.5 Forests 417
32.6 Mimetypes 417
32.7 Security 417

33.0 Appendix B: Pre-defined Execute Privileges 418

34.0 Appendix C: Pre-defined Roles 440
34.1 admin 442
34.2 admin-builtins 442
34.3 admin-module-internal 443
34.4 alert-admin 443
34.5 alert-execution 443
34.6 alert-internal 443
34.7 alert-user 444
34.8 app-builder 444
34.9 app-builder-internal 444
34.10 app-user 444
34.11 application-plugin-registrar 444
34.12 appservices-internal 446
34.13 cpf-restart 446
34.14 custom-dictionary-admin 446
34.15 custom-dictionary-user 446
34.16 dls-admin 446
34.17 dls-internal 446
34.18 dls-user 446
34.19 domain-management 447
34.20 filesystem-access 447
34.21 flexrep-admin 447
34.22 flexrep-internal 447
34.23 flexrep-user 448
34.24 hadoop-internal 448
34.25 hadoop-user-all 448
34.26 hadoop-user-read 448
34.27 hadoop-user-write 449
34.28 infostudio-admin-internal 449
34.29 infostudio-internal 449
34.30 infostudio-user 449
34.31 manage-admin 450
34.32 manage-admin-internal 450
34.33 manage-internal 450
34.34 manage-user 451

34.35 merge 451
34.36 network-access 451
34.37 pipeline-execution 451
34.38 pipeline-management 452
34.39 pki 452
34.40 plugin-internal 452
34.41 qconsole-internal 452
34.42 qconsole-user 452
34.43 rest-admin 452
34.44 rest-admin-internal 452
34.45 rest-extension-user 453
34.46 rest-internal 453
34.47 rest-reader 453
34.48 rest-writer-internal 453
34.49 rest-writer 453
34.50 rest-reader-internal 453
34.51 search-internal 453
34.52 security 453
34.53 trigger-management 455
34.54 xa 455
34.55 xa-admin 456
34.56 welcome-internal 456
34.57 xinclude 456

35.0 Technical Support 457

36.0 Copyright 458
36.0 NOTICE 458

1.0 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, specifically the system administrator in charge of MarkLogic .

1.3 Scope and Requirements

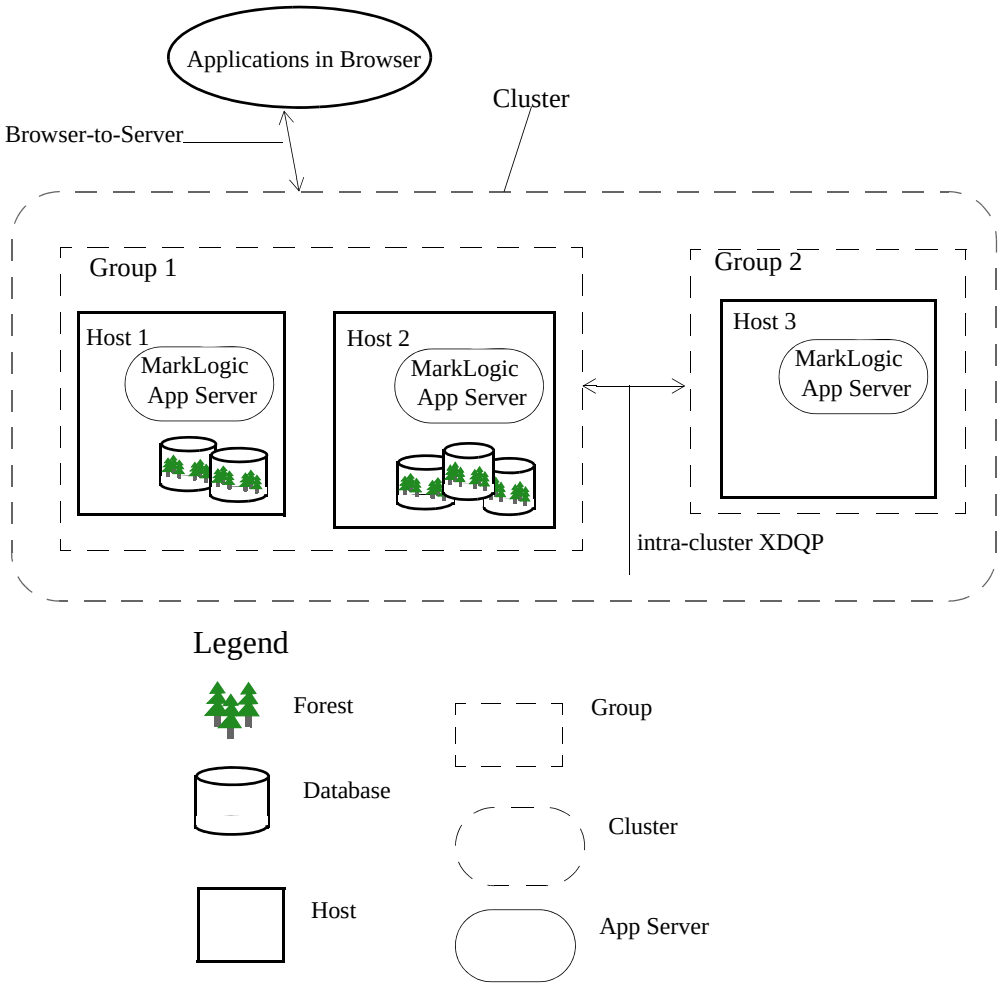
This guide explains administrative tasks for MarkLogic running on all platforms. For details on the supported platforms, see the *Installation Guide* and the *Release Notes*.

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

- *Getting Started With MarkLogic Server*
- *MarkLogic Server Installation Guide*

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 the *Release Notes*.

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 “OpenSSL FIPS 140-2 Mode” on page 28.

For more information, see the following:

- Hosts
- App Servers, see the specific server type:
  - HTTP Servers
  - XDBC Servers
  - WebDAV Servers
  - ODBC Servers
- Groups
- Clusters
- Databases
- Forests

2.0 Administrative Interface

The MarkLogic Server administrative interface (or Admin Interface) is used to configure the MarkLogic Server software on your system. This chapter provides a general overview of the Admin Interface and includes the following sections:

- Overview of the Admin Interface
- Accessing the Admin Interface
- Logging Off the Admin Interface
- Creating and Managing Administrators

2.1 Overview of the Admin Interface

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

- Manage basic software configuration
- Create and configure groups
- Create and manage databases
- Create and manage new forests
- Back up and restore forest content
- Create and manage new web server and Java-language access paths
- Create and manage security configurations
- Tune system performance
- Configure namespaces and schemas
- Check 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 the *Getting Started with MarkLogic Server* manual, then accessing the Admin Interface requires that you enter a user name and password. After you have been authenticated, you should not need to re-enter your user name and password to complete any of the other tasks outlined in this guide during the current session.

Some configurations 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 “Appendix A: ‘Hot’ versus ‘Cold’ Admin Tasks” on page 414.

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. Authorized administrators 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 follow proper administrative procedures.

To access the Admin Interface, complete the following procedure:

1. Open the following URL in a browser:

   ```
   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`.

2. Log in with your admin user name and password. The summary screen for the Admin Interface displays.

   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 used to access the Admin Interface. This action is sufficient to end the current session and force the user to authenticate again starting another session.

2.4 Creating and Managing Administrators

MarkLogic Server administrators are managed by defining which user has 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 follow proper administrative procedures. For the procedures for creating, managing and removing administrators, see “Security Administration” on page 292.

3.0 Common Administrative Procedures

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

The common administrative procedures are:

- Installing and Upgrading MarkLogic Server
- Starting and Stopping MarkLogic Server
- Creating and Configuring Forests and Databases
- Creating and Configuring App Servers
- Setting up Users, Roles, Privileges, and Permissions
- Loading Content into a Database
- Running The XQuery Use Cases and Building Simple Applications
- Backing up and Restoring Data
- Monitoring and Tuning Performance
- Scripting and Scheduling Administrative Tasks
- Configuring Clusters, Groups and Failover

3.1 Installing and Upgrading MarkLogic Server

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

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

- 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 and Stopping MarkLogic Server

The start, stop, and restart operations for MarkLogic Server are described in “Starting and Stopping MarkLogic Server” on page 25.

3.3 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 "Understanding Databases" on page 120. This section outlines the general procedures for creating a database to store your documents.

To create a database to store your documents, do the following:

1. Create one or more forests, as described in "Creating a Forest" on page 275. Depending on your storage, performance, and availability needs, you may want to create multiple forests, each on a separate host. See the *Scalability, Availability, and Failover Guide* for details.

2. Follow the procedure described in "Creating a New Database" on page 134 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 "Attaching and/or Detaching Forests to/from a Database" on page 135.

3.4 Creating and Configuring App Servers

An application is executed on an App Server, which is configured with a specific database, port number, and so on. 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 "Procedures for Creating and Managing HTTP Servers" on page 38.
- XDBC App Servers for Contentbase Connector (XCC) applications that use the Java and .NET XCC libraries. For information on creating and configuring an XDBC App Server, see "Procedures for Creating and Managing XDBC Servers" on page 47.
- WebDAV App Servers for accessing a MarkLogic Server database via a WebDAV client. For information on creating and configuring a WebDAV App Server, see "Procedures for Creating and Managing WebDAV Servers" on page 58.
- ODBC App Servers for accessing a MarkLogic Server database via a SQL client. For information on creating and configuring an ODBC App Server, see "Procedures for Creating and Managing ODBC Servers" on page 69.

To secure your App Server using SSL, see "Configuring SSL on App Servers" on page 78.

3.5 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 the *Understanding and Using Security Guide* and in "Security Administration" on page 292 in this guide.

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

3.6 Loading Content into a Database

You can load documents into the database using the load document functions, as described in the *Loading Content Into MarkLogic Server Guide*.

You can also set up a WebDAV server and client, such as Windows Explorer, to load your documents. 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.

Documents can also be loaded into the database by an XCC application, as described in Using the Sample Applications in the *XCC Developer's Guide*.

3.7 Running The XQuery Use Cases and Building Simple Applications

To test your MarkLogic Server configuration, Follow the procedure in *Getting Started with MarkLogic Server* for Exploring the Use Cases. 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 or .NET application that communicates with MarkLogic Server through the XCC API, see the *XCC Developer's Guide*

3.8 Backing up and Restoring Data

You can make backups of a database, as described in "Backing Up a Database" on page 245, 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 "Making Backups of a Forest" on page 283.

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 “Backing Up and Restoring a Database” on page 234.

You can restore an entire database from a database backup, as described in “Restoring a Database without Journal Archiving” on page 251. You can restore an individual forest from either a database backup, as described in “Restoring a Database without Journal Archiving” on page 251, or from an individual forest backup, as described in “Restoring a Forest” on page 286.

3.9 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:

- The configuration of MarkLogic Servers, as described in the Scalability Considerations in MarkLogic Server chapter in the *Scalability, Availability, and Failover Guide*.
- Merges, as described in “Overview of Merges: Merges are Good” on page 169.
- Fragment size, as described in “Fragments” on page 365.
- Index configuration, as described in “Text Indexing” on page 321.
- Range indexes, as described in “Range Indexes and Lexicons” on page 340.
- Reindexing your database, as described in “Reindexing a Database” on page 138.
- Database memory and journal settings, as described in “Memory and Journal Settings” on page 129.
- Database field configuration, as described in “Fields Database Settings” on page 152.
- Log levels, as described in “Understanding the Log Levels” on page 386.
- 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*.

3.10 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 *MarkLogic REST API Reference*.

You can schedule administrative scripts to be invoked at specific intervals or times, as described in “Scheduling Tasks” on page 390.

3.11 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 “Groups” on page 30. Different configurations of grouped hosts are useful when different groups of hosts perform different tasks or have different system capabilities.

Should a host go 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*.

4.0 Starting and Stopping MarkLogic Server

Use the following procedures to start and stop MarkLogic Server:

- Starting the Server
- Stopping the Server
- Restarting the Server
- Example XQuery Scripts

4.1 Starting the Server

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

Platform	Command
Microsoft Windows	Select Start > Programs > MarkLogic Server > Start MarkLogic Server **Note:** When you start MarkLogic Server from the Start menu, the Windows service configuration for MarkLogic Server is set to start automatically. Also, if you are using Windows Vista or Windows 7, to start the service you must right-click the Start MarkLogic Server link in the Start menu and choose Run as Administrator, then choose to allow the action.
Red Hat Linux	`/etc/init.d/MarkLogic start`
Sun Solaris	`/etc/init.d/MarkLogic start`

4.2 Stopping the Server

There are two ways to perform a clean shutdown of MarkLogic Server:

- Using System Command to Stop MarkLogic Server
- Using the Admin Interface to Stop MarkLogic Server

4.2.1 Using System Command to Stop MarkLogic Server

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 **Note:** If you are using Windows Vista or Windows 7, 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.
Red Hat Linux	`/etc/init.d/MarkLogic stop`
Sun Solaris	`/etc/init.d/MarkLogic stop`

4.2.2 Using the Admin Interface to Stop MarkLogic Server

To stop the server from the Admin Interface, complete the following procedure:

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 on the top right.
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 “Starting the Server” on page 25.

4.3 Restarting the Server

To restart the server from the Admin Interface, complete the following procedure:

1. Click the Hosts icon on the left tree menu.
2. Click the Status tab on the top right.
3. Click Restart.
4. A confirmation message displays while restarting. Click OK to restart MarkLogic Server.

You may also manually stop and start the server as described above.

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.

4.4 Example XQuery Scripts

This section provides the following XQuery scripts:

- Script that Restarts MarkLogic Server
- Script that Stops MarkLogic Server

4.4.1 Script that Restarts MarkLogic Server

The following script restarts MarkLogic Server:

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

4.4.2 Script that Stops MarkLogic Server

The following script stops MarkLogic Server:

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

5.0 Clusters

This chapter describes cluster configuration using the Admin Interface. A *cluster* is a set of hosts that work together. This chapter includes the following sections:

- Overview of Cluster Configuration
- OpenSSL FIPS 140-2 Mode
- Procedures for Configuring Clusters

5.1 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*.

5.2 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: http://www.openssl.org/docs/fips/fipsvalidation.html.

5.3 Procedures for Configuring Clusters

The following procedures describe how to configure clusters in MarkLogic Server:

- Configuring OpenSSL FIPS 140-2 Mode
- Coupling Clusters

5.3.1 Configuring OpenSSL FIPS 140-2 Mode

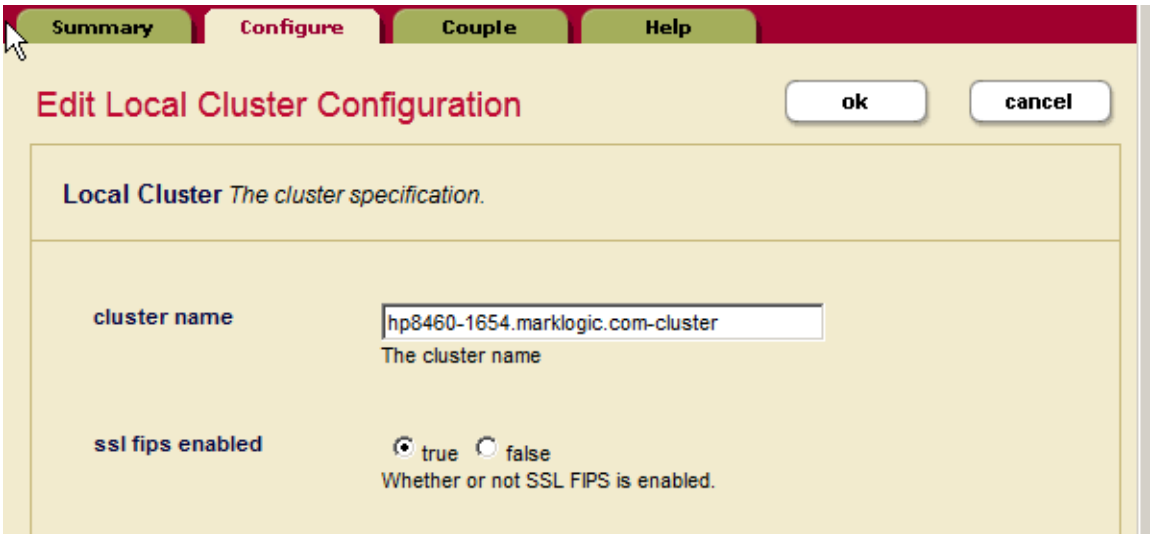
When FIPS 140-2 mode is enabled, the OpenSSL library is initialized into FIPS 140-2 mode at system startup. Note that this is the default behavior of MarkLogic Server. If FIPS mode is enabled or disabled on a running system, the OpenSSL library is reconfigured appropriately without requiring a server 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, perform the following steps:

1. Log into the Admin Interface.

2. Click the Clusters icon on the left tree menu.

3. Select the local cluster. The Edit Local Cluster Configuration page appears.



4. To configure FIPS 140-2 mode, select `true` or `false` as needed. For SSL FIPS Enabled, select `true`.

5. Click OK to save the change.

5.3.2 Coupling Clusters

You can use the Admin Interface to couple local and foreign clusters to enable inter-cluster communication. To replicate a database from one cluster to a database in another cluster, the two clusters must be coupled. For information on how inter-cluster communication relates to database replication, see the *Database Replication Guide*. For details, on coupling clusters specifically, see Coupling the Local and Foreign Clusters in the *Database Replication Guide*.

6.0 Groups

This chapter describes groups in MarkLogic Server, and includes the following sections:

- Overview of Groups
- Example
- Procedures for Configuring and Managing Groups

This chapter 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*.

6.1 Overview of Groups

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

- A g*roup* 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.

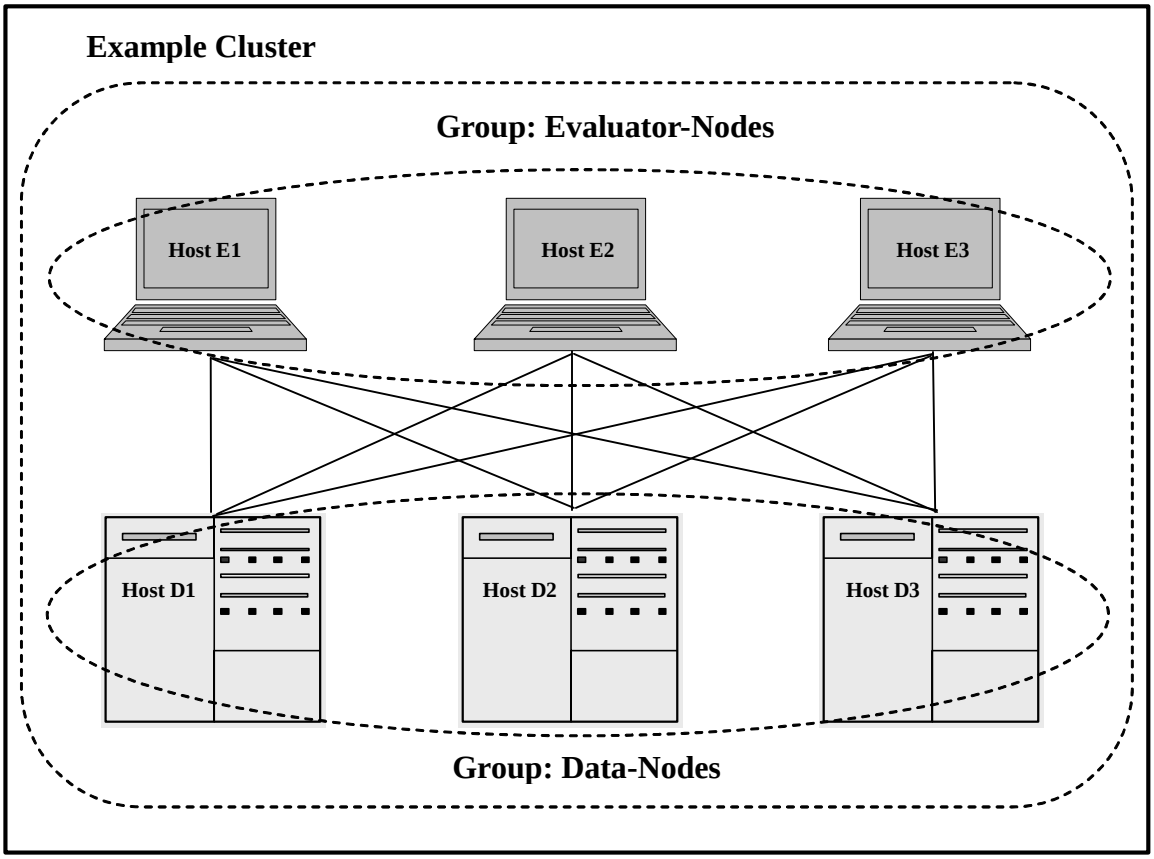
The Configure tab of the Group Administration section of the Admin Interface enables you to define configuration information for memory settings, SMTP server settings, and other configuration settings. The values for the settings are set at installation time based on your system memory configuration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

6.2 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 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.

6.3 Procedures for Configuring and Managing Groups

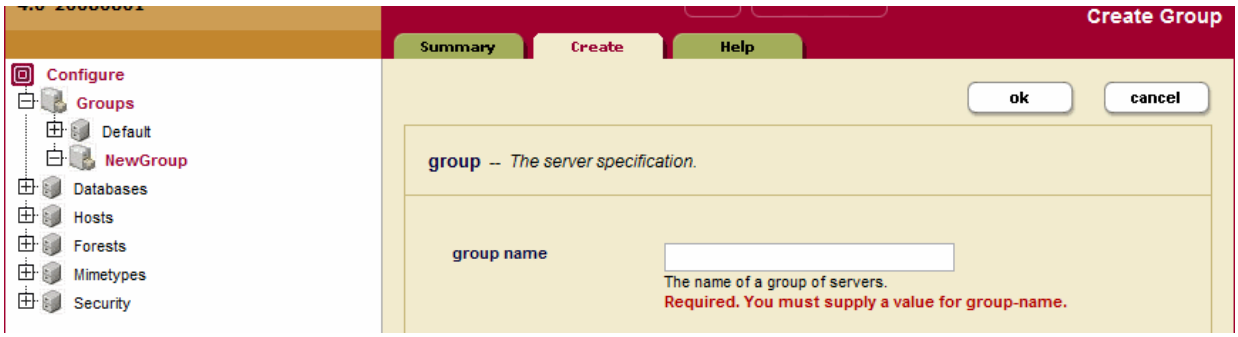
The following procedures describe how to create and manage groups in MarkLogic Server:

- Creating a New Group
- Viewing Group Settings
- Deleting a Group
- Enabling SSL communication over XDQP
- Configuring an SMTP Server
- Restarting All Hosts in a Group

6.3.1 Creating a New Group

To create a new group, perform the following steps:

1. Log into the Admin Interface.
2. Click the Groups icon on the left tree menu.
3. Click the Create tab on the Group Summary page. The Create Group page displays.



4. Go to the Group Name field and enter a short hand name for the group.

   MarkLogic Server will use this name to refer to the group.

5. You can change the value of List Cache Size, Compressed Tree Cache Size and Expanded Tree Cache Size or leave the defaults. They specify the amount of memory dedicated to caching term list, tree data in compressed form and tree data in expanded form.

6. 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.

7. 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.

8. The Rotate Log Files field specifies how often to start a new log file. You may change this field or use the default value provided.

9. The Keep Log Files field specifies how many log files are kept. You may change this field or use the default value provided.

10. 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.

11. The SSL Enabled option and XDQP SSL Ciphers field are to enable SSL for XDQP.

12. Click OK.

    Note: For information about auditing, including how to configure various audit events, see “Auditing Events” on page 106.

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

6.3.2 Viewing Group Settings

To view the settings for a particular group, perform the following steps:

1. Log into the Admin Interface.

2. Click the Groups icon on the left tree menu.

3. Click the Configure tab at the top right.

4. Locate the group for which you want to view settings.

5. Click the icon for this group.

6. View the settings.

6.3.3 Deleting a Group

You must drop all hosts assigned to a group before you can delete a group. To delete a group, perform the following steps:

1. Log into the Admin Interface.
2. Click the Groups icon on the left tree menu.
3. Click the Configure tab at the top right.
4. Locate the Group to be deleted.
5. Click on Hosts to check that there is no host assigned to the group. 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.
6. Click the icon for this group again.
7. Click Delete. Deleting a group deletes it from the system.
8. 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.

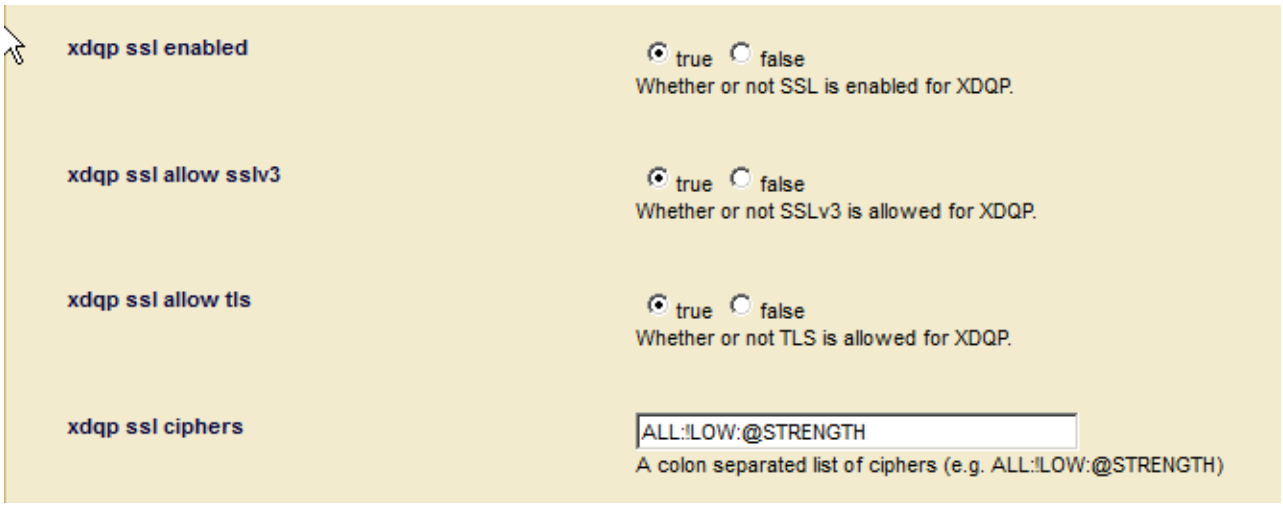
6.3.4 Enabling SSL communication over XDQP

To enable encrypted SSL communication between hosts in the group, 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 is used to sign certificates used between servers communicating over the internal XDQP connections in a cluster. Such certificates are self-signed and trusted by each server in the cluster.

For details on configuring SSL communication between web browsers and App Servers, see “Configuring SSL on App Servers” on page 78. For details on configuring FIPS 140-2 mode for SSL communication, see “OpenSSL FIPS 140-2 Mode” on page 28.

The following screen capture shows the options related to configuring SSL for intra-cluster XDQP communication.


xdqp ssl enabled
true false
Whether or not SSL is enabled for XDQP.
xdqp ssl allow sslv3
true false
Whether or not SSLv3 is allowed for XDQP.
xdqp ssl allow tls
true false
Whether or not TLS is allowed for XDQP.
xdqp ssl ciphers
ALL:!LOW:@STRENGTH
A colon separated list of ciphers (e.g. ALL:!LOW:@STRENGTH)

6.3.5 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 the `xdmp:email` function.

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

1. Log into the Admin Interface.
2. Click the Groups icon on the left tree menu.
3. Click the Configure tab at the top right.
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.

6.3.6 Restarting All Hosts in a Group

Perform the following steps to restart all the hosts in a group from the Admin Interface:

1. Click the Groups icon on the left tree menu.
2. Click the name of the group you want to restart, either from the menu tree of from the Group Summary page.
3. Click the Status tab on the top right.
4. Click Restart.
5. A confirmation message displays while restarting. 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.

7.0 HTTP Servers

This chapter describes HTTP servers and provides procedures for configuring them. The following sections are included:

- HTTP Server Overview
- Procedures for Creating and Managing HTTP Servers

This chapter describes how to use the Admin Interface to create and configure HTTP servers. For details on how to create and configure HTTP servers programmatically, see Creating and Configuring App Servers in the *Scripting Administrative Tasks Guide*.

7.1 HTTP Server Overview

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 “Security Administration” on page 292. For conceptual information on the MarkLogic Server security model, see *Understanding and Using Security Guide*.

7.2 Procedures for Creating and Managing HTTP Servers

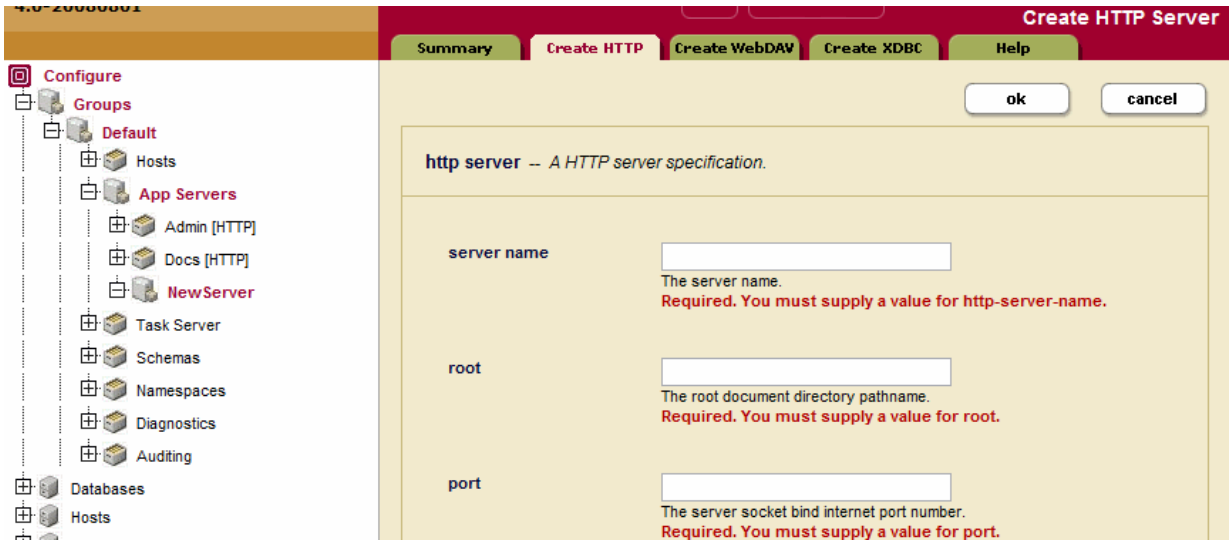
Use the following procedures to create and manage HTTP servers:

- Creating a New HTTP Server
- Setting Output Options for an HTTP Server
- Viewing HTTP Server Settings
- Deleting an HTTP Server
- Canceling a Request

7.2.1 Creating a New HTTP Server

To create a new server, complete the following steps:

1. Click the Groups icon in the left frame.
2. Click the group in which you want to define the HTTP server (for example, Default).
3. Click the App Servers icon on the left tree menu.
4. Click the Create HTTP tab at the top right. The Create HTTP Server page will display:



5. 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.

6. In the Root directory 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`
Sun Solaris	`/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.

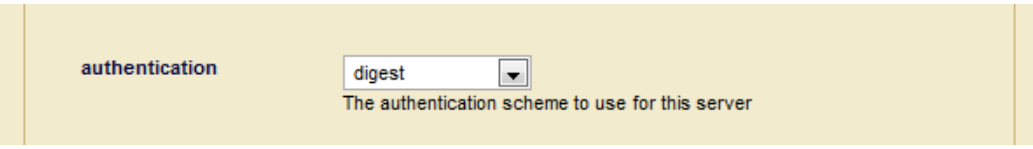
7. In the Port field, enter the port number through which you want to make this HTTP server available.

   The port number must not be assigned to any other HTTP, ODBC, XDBC, or WebDAV server.

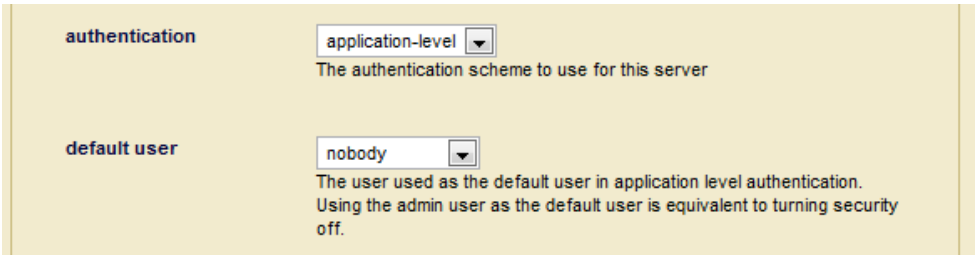
8. In the Modules field, select the database to use as the modules database for your documents, or leave it at the default of storing your modules on the file system. For information on what a modules database is, see “Modules Database” on page 121.

9. In the Database field and select the database to be accessed by this HTTP server. Multiple HTTP, ODBC, XDBC, and WebDAV servers can access the same database.

10. Scroll to the Authentication field. Select an authentication scheme: digest, basic, digestbasic, or application-level. 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.

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 HTTP server must have the execute privilege selected in order to access the HTTP 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 “Storing and Monitoring the Last User Login Attempt” on page 118.
- Address specifies the IP address for the App Server.
- 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 Sessions" on page 116.
- Log Errors specifes 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" on page 78.

13. 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 "Managing User Sessions and Monitoring Login Attempts" on page 116.

7.2.2 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, complete the following steps:

1. Click the Groups icon in the left frame.
2. Click the group which contains the HTTP 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*.

7.2.3 Viewing HTTP Server Settings

To view the settings for a particular HTTP server, complete the following steps:

1. Click the Groups icon in the left frame.
2. Click the group which contains the HTTP server you want to view (for example, Default).
3. Click the App Servers icon on the left tree menu.
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 icon for the HTTP server.
6. View the settings.

7.2.4 Deleting an HTTP Server

To delete the settings for an HTTP server, complete the following steps:

1. Click the Groups icon in the left frame.
2. Click the group which contains the HTTP server you want to delete (for example, Default).
3. Click the App Servers icon on the left tree menu.

4. Locate the HTTP server you want to delete, either in the tree menu or on the summary page.

5. Click the icon for the HTTP server.

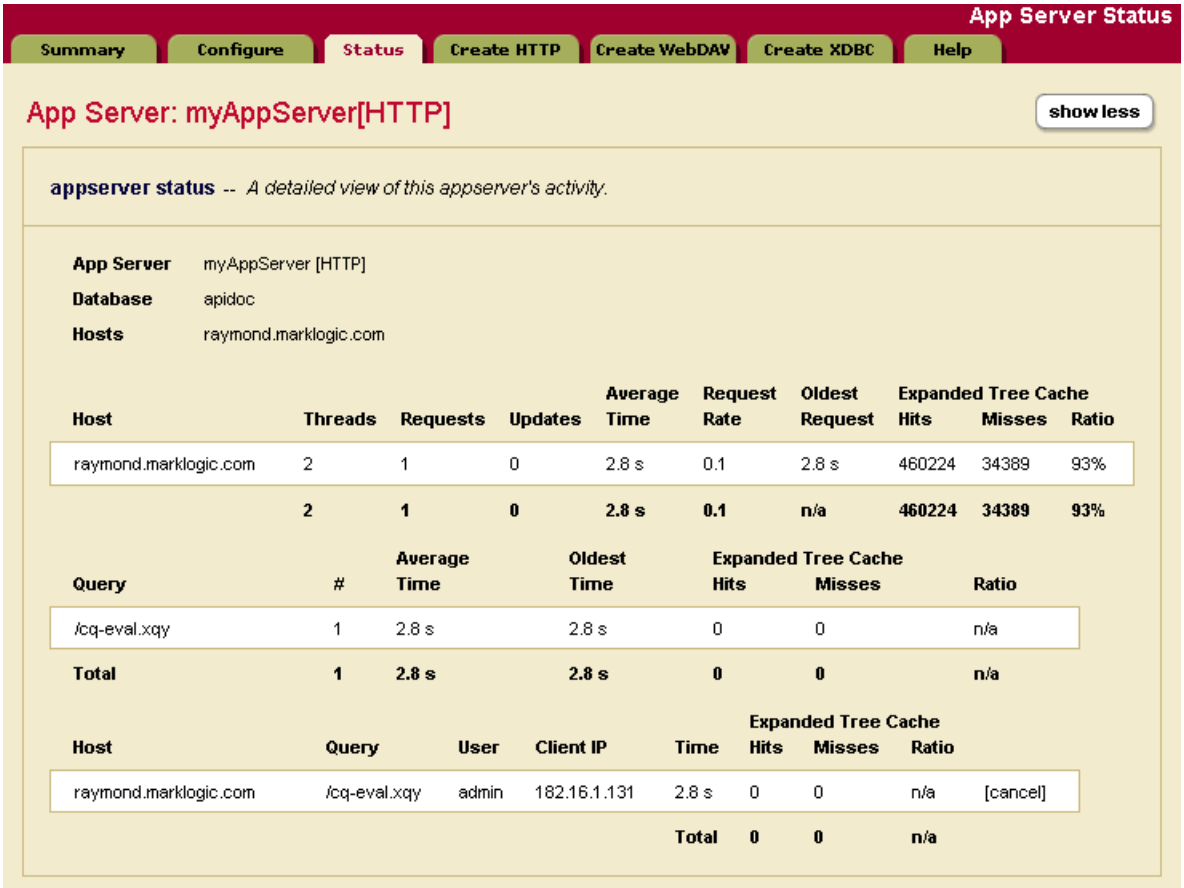
6. Click Delete.

7. 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.

7.2.5 Canceling a Request

You can cancel a request in the App Server Status page of the Admin Interface (Groups > *group_name* > App Servers > *app_server_name* > Status tab).



To cancel a long-running request (for example, a long-running query statement or update statement), perform the following steps:

1. Click the Group menu item in the Admin Interface.

2. Navigate to the App Server in which the request was issued, either from the tree menu or from the summary page.

3. Click the Status tab.

4. Click the Show More button.

5. At the bottom right of the App Server Status page, click the cancel button 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.

8.0 XDBC Servers

This chapter describes XDBC servers and provides procedures for configuring them. The following sections are included:

- XDBC Server Overview
- Procedures for Creating and Managing XDBC Servers

This chapter describes how to use the Admin Interface to create and configure XDBC servers. For details on how to create and configure XDBC servers programmatically, see Creating and Configuring App Servers in the *Scripting Administrative Tasks Guide*.

8.1 XDBC Server Overview

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 or .NET 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 “Security Administration” on page 292. For conceptual information on the MarkLogic Server security model, see *Understanding and Using Security Guide*.

8.2 Procedures for Creating and Managing XDBC Servers

Use the following procedures to create and manage XDBC servers:

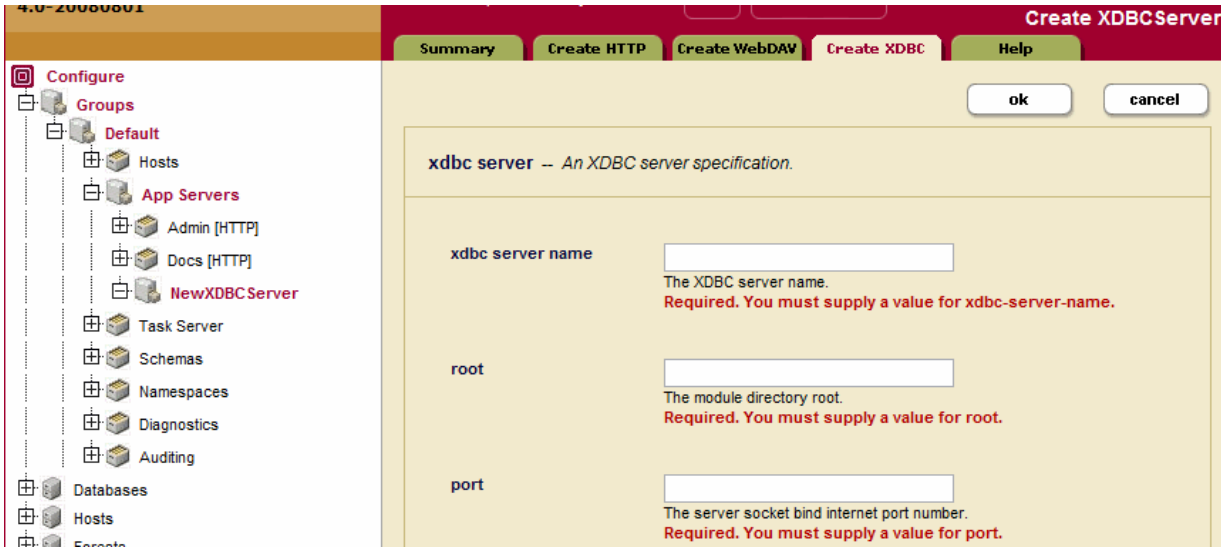
- Creating a New XDBC Server
- Setting Output Options for an XDBC Server
- Viewing XDBC Server Settings
- Deleting an XDBC Server

For the procedure to cancel a running request on an XDBC server, see “Canceling a Request” on page 44.

8.2.1 Creating a New XDBC Server

To create a new server, complete the following steps:

1. Click the Groups icon.
2. Click the group in which you want to define the XDBC server (for example, Default).
3. Click the App Servers icon on the left tree menu.
4. Click the Create XDBC tab at the top right. The Create XDBC Server page displays.



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 directory 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`
Sun Solaris	`/opt/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.

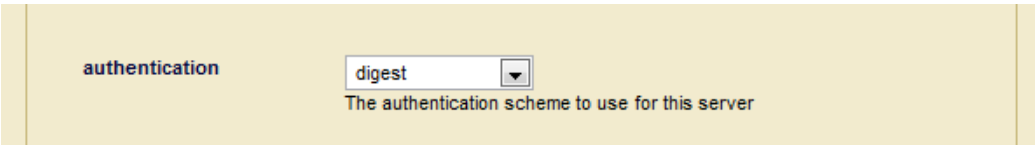
7. In the Port field, enter the port number through which you want to make this XDBC server available.

   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 “Modules Database” on page 121.

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. Select an authentication scheme: digest, basic, or digestbasic. 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 “Storing and Monitoring the Last User Login Attempt” on page 118.
    - Address specifies the IP address for the App Server.
    - 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*.
- 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 Sessions" on page 116.
- Log Errors specifes 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 properties associated with SSL support are described in "Configuring SSL on App Servers" on page 78.

13. Scroll to the top or bottom and click OK.

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 "Managing User Sessions and Monitoring Login Attempts" on page 116.

8.2.2 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 (http://www.w3.org/TR/xslt20#serialization).

To specify defaults for the App Server, complete the following steps:

1. Click the Groups icon in the left frame.
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*.

8.2.3 Viewing XDBC Server Settings

To view the settings for an XDBC server, complete the following 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 for the XDBC server.
6. View the settings.

8.2.4 Deleting an XDBC Server

To delete the settings for an XDBC server, complete the following steps:

1. Click on the Groups icon.
2. Click on the group which contains the XDBC server you want to delete (for example, Default).
3. Click the App Servers icon on the left tree menu.
4. Locate the XDBC server to be deleted, either in the tree menu or on the summary page.
5. Click the icon for this XDBC server.
6. Click Drop.
7. 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.

9.0 WebDAV Servers

A WebDAV server in MarkLogic Server is similar to an HTTP server, but has the following 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.

This chapter describes WebDAV servers in MarkLogic Server and includes the following sections:

- WebDAV Server Overview
- Procedures for Creating and Managing WebDAV Servers
- WebDAV Clients
- Example: Setting Up a WebDAV Server to Add/Modify Documents Used By Another Server

This chapter describes how to use the Admin Interface to create and configure WebDAV servers. For details on how to create and configure WebDAV servers programmatically, see Creating and Configuring App Servers in the *Scripting Administrative Tasks Guide*.

9.1 WebDAV Server Overview

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 “WebDAV Clients” on page 63. For general information about WebDAV and the WebDAV protocol, see the following web site:

http://webdav.org

This section provides an overview of WebDAV servers in MarkLogic Server, and includes the following topics:

- Accesses a Database for Read and Write, Not XQuery Execution
- WebDAV Server Security
- Directories
- Server Root Directory
- Documents in a WebDAV Server

9.1.1 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).

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 “Example: Setting Up a WebDAV Server to Add/Modify Documents Used By Another Server” on page 66.

9.1.2 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 “Security Administration” on page 292 and the chapters related to security in the *Application Developer’s Guide*.

9.1.3 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.

The following sections describe further details about directories:

- Automatic Directory Creation in a Database Settings
- Properties and URIs of Directories

For more details on directories and properties, see the "Property Documents and Directories" chapter of the *Application Developer's Guide*.

9.1.3.1 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 the `xdmp:document-load` function, 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 the `xdmp:directory-create` and `xdmp:directory-delete` built-in functions.

For details on all of the directory creation settings, see "Basic Administrative Settings" on page 122.

9.1.3.2 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 the `xdmp:document-properties()` function 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.

The `xdmp:document-properties()` function 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 the XQuery `xdmp:node-uri` built-in function 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.

9.1.4 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 *Understanding and Using Security Guide*.

9.1.5 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.

9.2 Procedures for Creating and Managing WebDAV Servers

This section includes procedures to perform the following actions:

- Creating a New WebDAV Server
- Setting Output Options for a WebDAV Server
- Viewing WebDAV Server Settings
- Deleting a WebDAV Server

For the procedure to cancel a running request on a WebDAV server, see “Canceling a Request” on page 44.

9.2.1 Creating a New WebDAV Server

To create a new server, complete the following steps:

1. Click the Groups icon.
2. Click the group in which you want to define the WebDAV server (for example, Default).
3. Click the App Servers icon on the left tree menu.
4. Click the Create WebDAV tab at the top right.

   The WebDAV Server Create page displays.
5. Go to the WebDAV 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” on page 57.

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

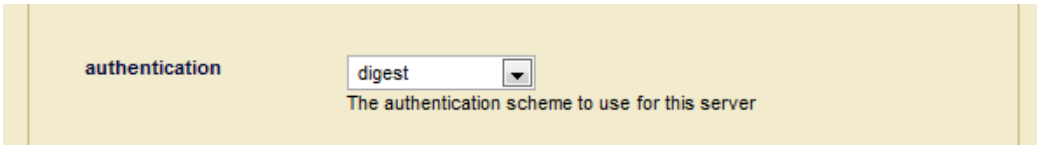
7. Go to the Port field and enter the port number through which you want to make this WebDAV server available. The port number must not be assigned to any other server.

8. Go to the Database field and select the database to be accessed by this WebDAV server.

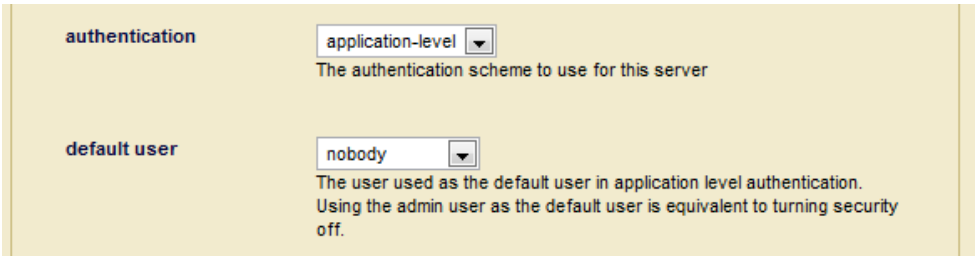
   Multiple HTTP, ODBC, XDBC, and WebDAV servers can be connected to the same database.

   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” on page 55.

9. Scroll to the Authentication field. Select an authentication scheme: digest, basic, digestbasic, or application-level. 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.

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” on page 118.
    - Address specifies the IP address for the App Server.
    - 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 Sessions” on page 116.
- Log Errors specifes 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 properties associated with SSL support are described in “Configuring SSL on App Servers” on page 78.

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.

9.2.2 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 (http://www.w3.org/TR/xslt20#serialization).

To specify defaults for the App Server, complete the following steps:

1. Click the Groups icon in the left frame.
2. Click the group which contains the WebDAV 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*.

9.2.3 Viewing WebDAV Server Settings

To view the settings for a WebDAV server, complete the following steps:

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

6. View the settings.

9.2.4 Deleting a WebDAV Server

To delete the settings for a WebDAV server, complete the following steps:

1. Click the Groups icon.
2. Click the group which contains the WebDAV server you want to delete (for example, Default).
3. Click the WebDAVServers icon on the left tree menu.
4. Click the Configure tab at the top right.
5. Locate the WebDAV server to be deleted, either in the tree menu or on the summary page.
6. Click the icon for this WebDAV server.
7. Click Delete.
8. A confirmation message displays. Confirm the delete and click OK.

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

9.3 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. The following topics are included:

- Tested WebDAV Clients
- General Steps to Connect to a Server
- Steps to Connect to a Web Folder in Windows Explorer

9.3.1 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 XP, Windows Vista, WIndows 7 in many configurations	Allows drag and drop from Windows. For instructions on setting up, see "Steps to Connect to a Web Folder in Windows Explorer" on page 65. Some Windows clients (for example Windows Vista and Windows 7 clients in most configurations) require digest authentication.
PerlDAV	http://www.webdav.org/perldav/	A command line, perl-based WebDAV client. Designed to be scriptable and to allow you to send individual WebDAV calls.
XML Spy	Altova Software (http://www.altova.com/)	Allows you to open, edit, and save XML files in XML Spy. Use the File > Open URL menu item in XML Spy.
jEdit DAV plug-in	Available on developer.marklogic.com	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`).

Note: Windows Vista and Windows 7 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 Vista or Windows 7 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.

9.3.2 General Steps to Connect to a Server

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

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 the following URL in the appropriate place for your WebDAV client:

   ```
   http://marklogic.myCompany.com:9001/
   ```

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 *Understanding and Using Security Guide*.

4. 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.

9.3.3 Steps to Connect to a Web Folder in Windows Explorer

If you are running Windows, perform the following steps to use the Windows Explorer WebDAV client:

1. Double-click the My Network Places icon on your desktop.

2. In My Network Places, double-click the Add Network Places icon.

3. 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 the following URL:

```
http://marklogic.myCompany.com:9001/
```

4. Click Next.

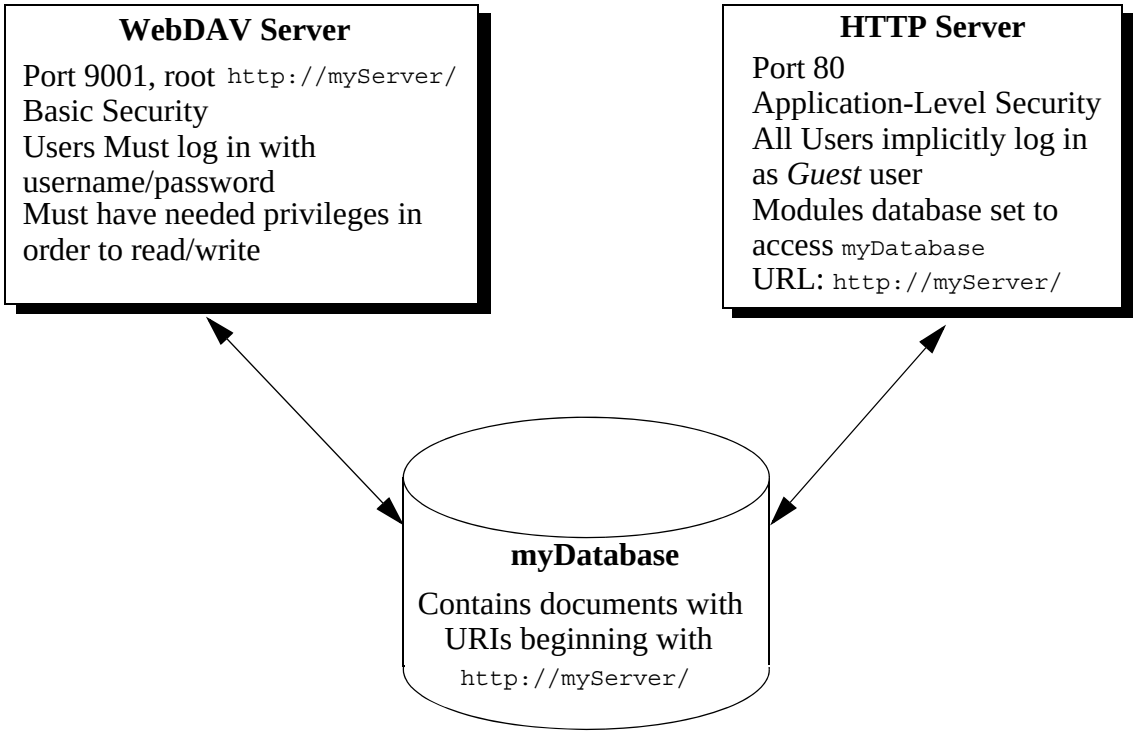
5. If prompted, enter your username and password for the WebDAV server.

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.

9.4 Example: Setting Up a WebDAV Server to Add/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 “Modules Database” on page 121.

10.0 ODBC Servers

An ODBC server is one of several components that support SQL queries to MarkLogic Server. This chapter describes ODBC servers and provides procedures for configuring them. The following sections are included:

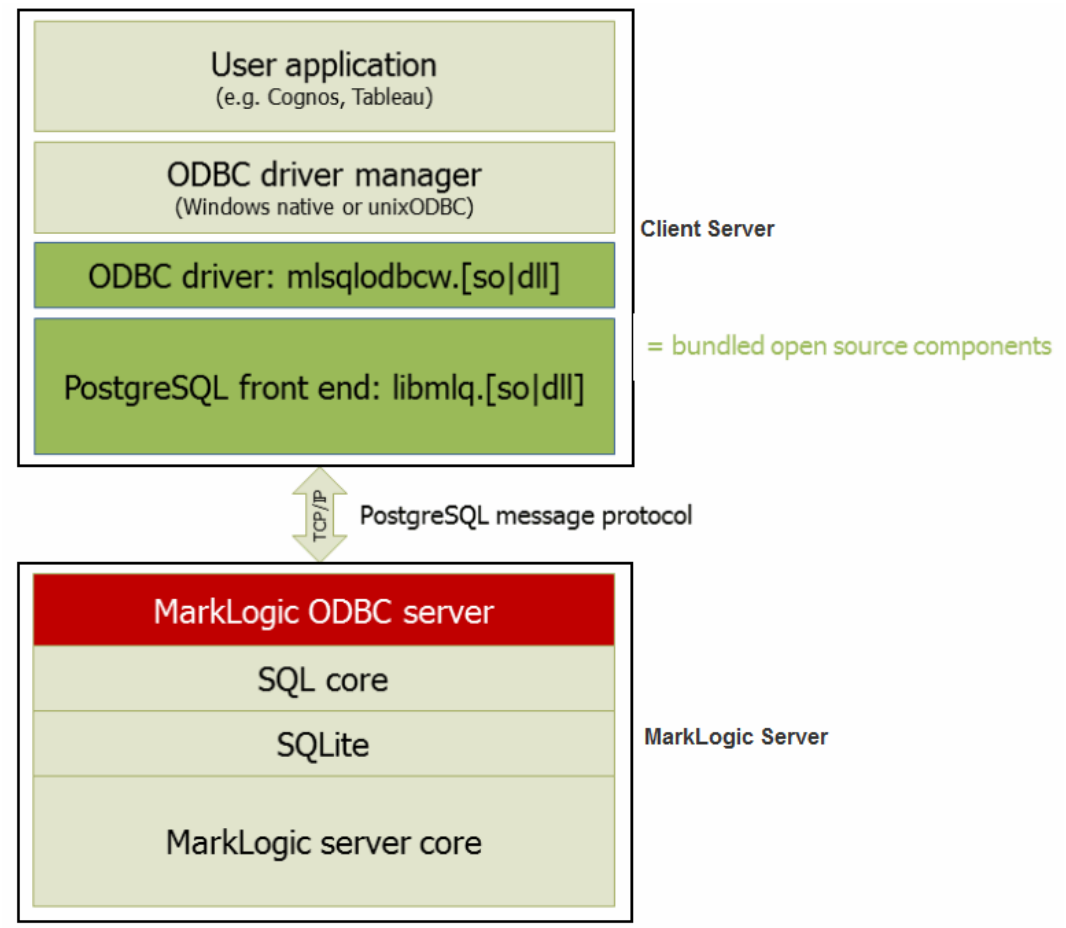
- ODBC Server Overview
- Procedures for Creating and Managing ODBC Servers

This chapter describes how to use the Admin Interface to create and configure ODBC servers. For details on how to create and configure ODBC servers programmatically, see Creating and Configuring App Servers in the *Scripting Administrative Tasks Guide*.

10.1 ODBC Server Overview

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.



10.2 Procedures for Creating and Managing ODBC Servers

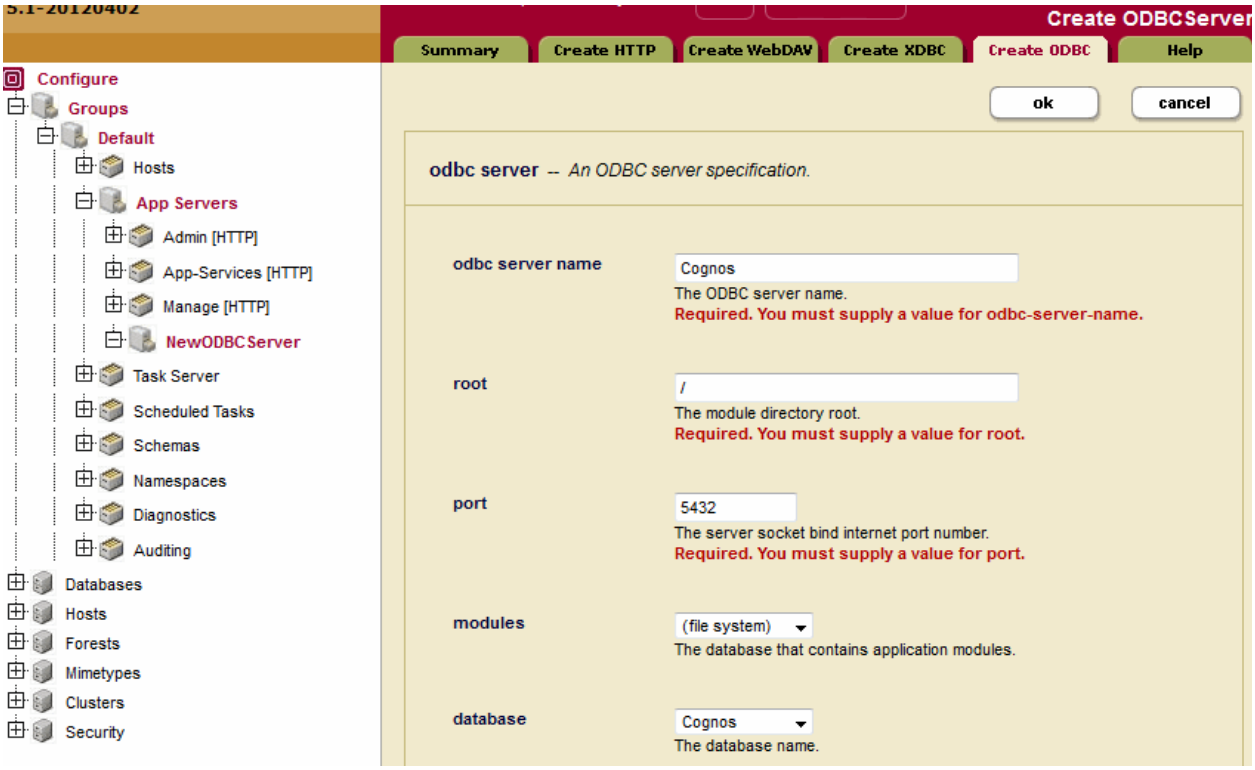
Use the following procedures to create and manage ODBC servers:

- Creating a New ODBC Server
- Setting Output Options for an ODBC Server
- Viewing ODBC Server Settings
- Deleting an ODBC Server
- Canceling a Request

10.2.1 Creating a New ODBC Server

To create a new server, complete the following steps:

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



5. In the 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 directory 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`
Sun Solaris	`/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.

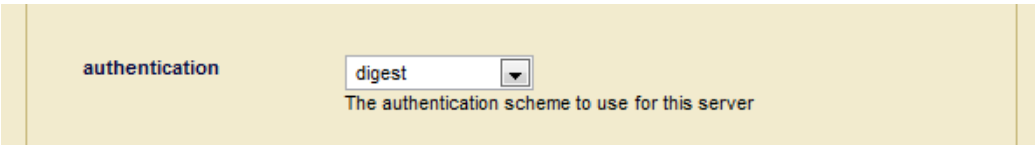
   Warning Do not create ODBC server root directories named Docs, Data or Admin. These directories are reserved by MarkLogic Server for other purposes. Creating ODBC server root directories with these names can result in unpredictable behavior of the server and may also complicate the software upgrade process.

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.

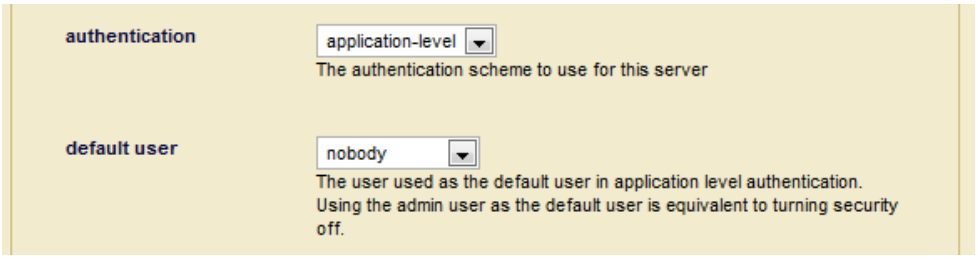
8. In the Modules field, select the database to use as the modules database for your XQuery documents, or leave it at the default of storing your XQuery modules on the file system. For information on what a modules database is, see “Modules Database” on page 121.

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: digest, basic, digestbasic, or application-level. 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 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 “Storing and Monitoring the Last User Login Attempt” on page 118.
- Address specifies the IP address for the App Server.
- 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 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" 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 "Managing Concurrent User Sessions" on page 116.
- Log Errors specifes 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" on page 78.

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 "Managing User Sessions and Monitoring Login Attempts" on page 116.

10.2.2 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 (http://www.w3.org/TR/xslt20#serialization).

To specify defaults for the App Server, complete the following steps:

1. Click the Groups icon in the left frame.
2. Click the group which contains the ODBC 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*.

10.2.3 Viewing ODBC Server Settings

To view the settings for a particular ODBC server, complete the following steps:

1. Click the Groups icon in the left frame.
2. Click the group which contains the ODBC server you want to view (for example, Default).
3. Click the App Servers icon on 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 icon for the ODBC server.
6. View the settings.

10.2.4 Deleting an ODBC Server

To delete the settings for an ODBC server, complete the following steps:

1. Click the Groups icon in the left frame.
2. Click the group which contains the ODBC server you want to delete (for example, Default).
3. Click the App Servers icon on 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 icon for the ODBC server.

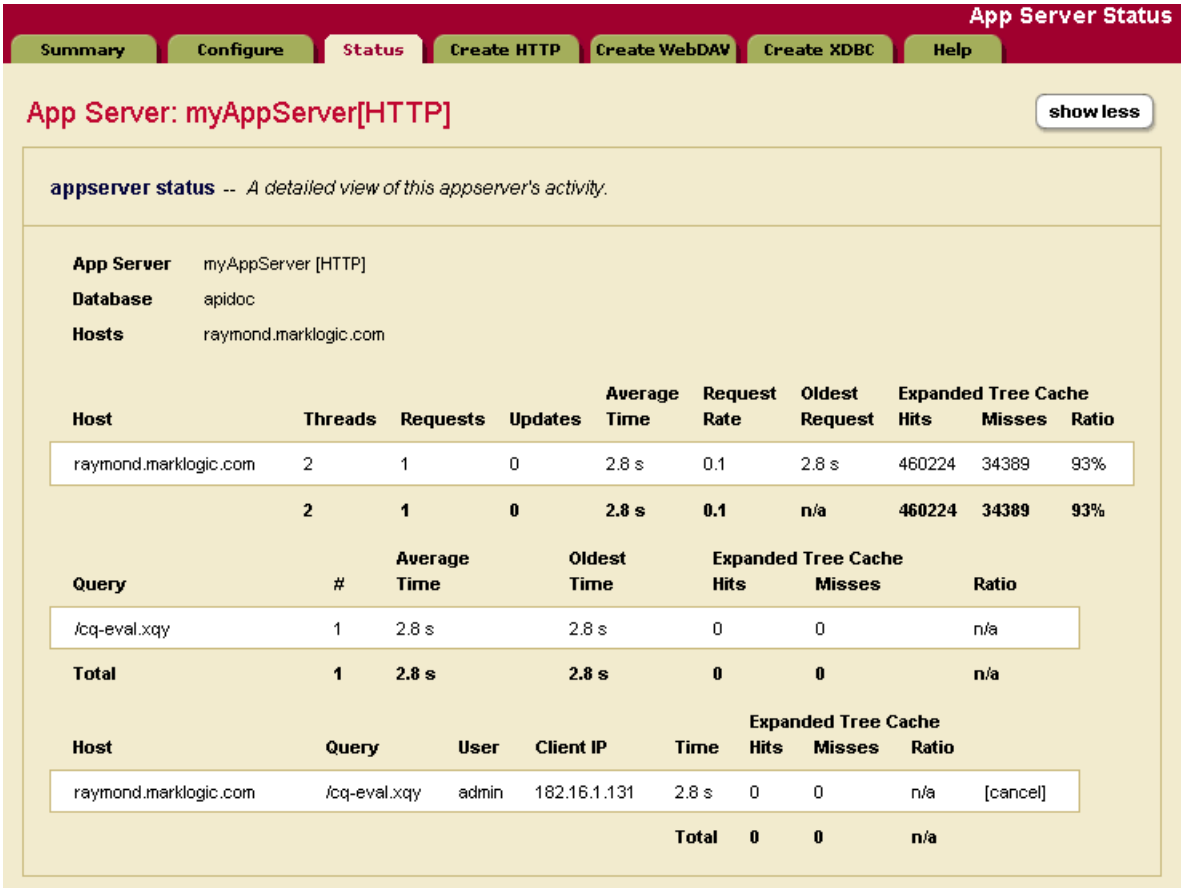
6. Click Delete.

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.

10.2.5 Canceling a Request

You can cancel a request in the App Server Status page of the Admin Interface (Groups > *group_name* > App Servers > *app_server_name* > Status tab).



To cancel a long-running request (for example, a long-running query statement or update statement), perform the following steps:

1. Click the Group menu item in the Admin Interface.

2. Navigate to the App Server in which the request was issued, either from the tree menu or from the summary page.

3. Click the Status tab.

4. Click the Show More button.

5. At the bottom right of the App Server Status page, click the cancel button 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.

11.0 Configuring SSL on App Servers

This chapter describes SSL support in the MarkLogic Server, and includes the following sections:

- Understanding SSL
- General Procedure for Setting up SSL for an App Server
- Procedures for Enabling SSL on App Servers
- Accessing an SSL-Enabled Server from a Browser or WebDAV Client
- Procedures for Obtaining a Signed Certificate
- Viewing Trusted Certificate Authorities
- Importing a Certificate Revocation List into MarkLogic Server
- Deleting a Certificate Template

This chapter describes how to use the Admin Interface to configure SSL on App Servers. For details on how to configure SSL programmatically, see Enabling SSL on an App Server in the *Scripting Administrative Tasks Guide*.

11.1 Understanding SSL

SSL (Secure Sockets Layer) is a transaction security standard that provides encrypted protection between browsers and App Servers. When SSL is enabled for an App Server, browsers communicate with the App Server by means of an HTTPS connection, which is HTTP over an encrypted Secure Sockets Layer. HTTPS connections are widely used by banks and web vendors for secure transactions over the web.

A browser and App Server create a secure HTTPS connection by using a handshaking procedure. When browser connects to an SSL-enabled App Server, the App Server sends back its identification in the form of a digital certificate that contains the server name, the trusted certificate authority, and the server's public encryption key. The browser uses the server's public encryption key from the digital certificate to encrypt a random number and sends the result to the server. From the random number, both the browser and App Server generate a *session key.* The session key is used for the rest of the session to encrypt/decrypt all transmissions between the browser and App Server, enabling them to verify that the data didn't change in route.

The end result of the handshaking procedure described above is that only the server is authenticated. The client can trust the server, but the client remains unauthenticated. MarkLogic Server supports mutual authentication, in which the client also holds a digital certificate that it sends to the server. When mutual authentication is enabled, both the client and the server are authenticated and mutually trusted.

MarkLogic Server uses OpenSSL to implement the Secure Sockets Layer (SSL v3) and Transport Layer Security (TLS v1) protocols.

The following are the definitions for the SSL terms used in this chapter:

- A *certificate*, or more precisely, a *public key certificate*, is an electronic document that incorporates a digital signature to bind together a public key with identity information, such as the name of a person or an organization, address, and so on. The certificate can be used to verify that a public key belongs to an individual or organization. In a typical public key infrastructure (PKI) scheme, the signature will be that of a 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. 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.
- A *certificate chain* is a group of interdependent CAs. A certificate chain consists of a single trusted *root CA*, one or more *intermediate CA*, and one or more *end CA*. The intermediate and end certificates must be imported into MarkLogic.

  Note: MarkLogic supports only one intermediate CA per host.

- A *certificate request* is a request data structure containing a subset of the information that will ultimately end up in the certificate. A certificate request is sent to a *certificate authority* for certification.
- A *key* is a piece of information that determines the output of a cipher. SSL/TLS communications begin with a public/private key pair that allow the client and server to securely agree on a session key. The public/private key pair is also used to validate the identity of the server and can optionally be used to verify the identity of the client.
- A *certificate template* is a MarkLogic construct that is used to generate certificate requests for the various hosts in a cluster. The template defines the name of the certificate, a description, and identity information about the owner of the certificate.
- A *cipher* is an algorithm for encrypting information so that it's only readable by someone with a key. A cipher can be either symmetric and asymmetric. Symmetric ciphers use the same key for both encryption and decryption. Asymmetric ciphers use a public and private key.

11.2 General Procedure for Setting up SSL for an App Server

This section describes the general procedure for setting up SSL on an App Server. The general steps are:

- Create a certificate template, as described in “Creating a Certificate Template” on page 81.
- Enable SSL for the App Server, as described in “Enabling SSL for an App Server” on page 83.
- Access the SSL-enabled server from a browser, as described in “Accessing an SSL-Enabled Server from a Browser or WebDAV Client” on page 84.
- Generate a certificate request and send it off to a certificate authority, as described in “Generating and Downloading Certificate Requests” on page 99.
- When you receive the signed certificate from the certificate authority, import it into MarkLogic Server for use by your App Server, as described in “Importing a Signed Certificate into MarkLogic Server” on page 101.

  Note: Certificate templates, requests, and the resulting signed certificates are only valid within a single cluster.

11.3 Procedures for Enabling SSL on App Servers

The following sections describe how to enable SSL for an App Server:

- Creating a Certificate Template
- Enabling SSL for an App Server

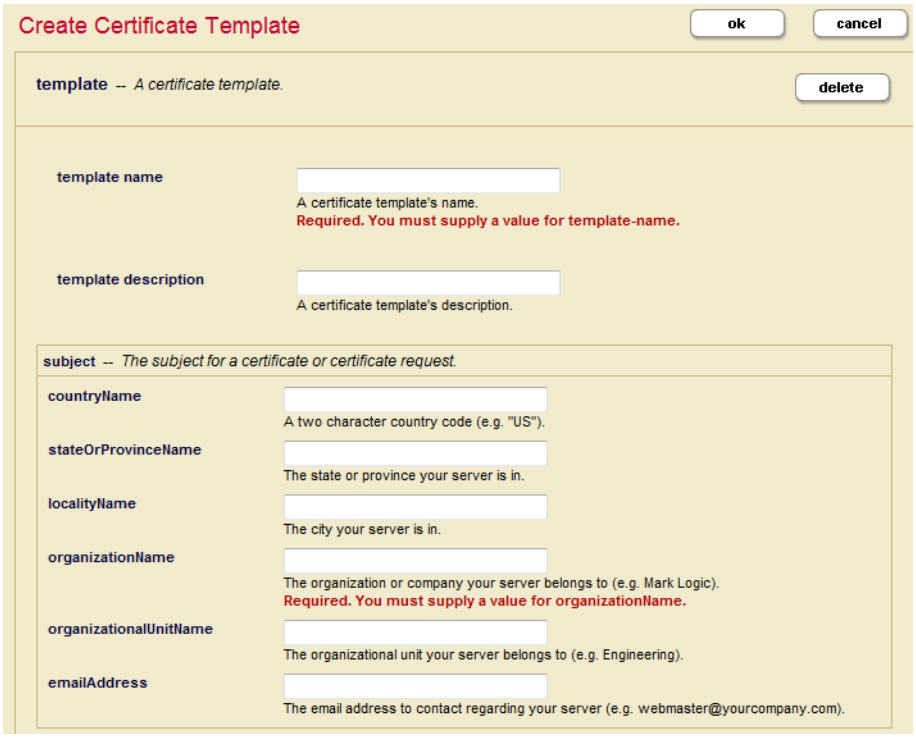
11.3.1 Creating a Certificate Template

Access to an SSL-enabled server is managed by a public key in a signed certificate obtained from a certificate authority. The first step in producing a request for a signed certificate is to define a certificate template. This procedure will produce a self-signed certificate that your browser can temporarily use to access an SSL-enabled server until you receive a signed certificate from a certificate authority.

1. Click the Security icon in the left frame.

2. Click the Certificate Templates icon on the left tree menu.

3. Click the Create tab. The Create Certificate Template page will display:



4. In the Template Name field, enter a shorthand name for this certificate template. MarkLogic Server will use this name to refer to this template on display screens in the Admin Interface.

5. You can enter an optional description for the certificate template.



6. Enter the name of your company or organization in the Organization Name field.

7. You can optionally fill in subject information, such as your country, state, locale, and email address. Country Name must be two characters, such as US, UK, DE, FR, ES, etc.



8. When you have finished filling in the fields, click OK. MarkLogic Server automatically generates a Self-Signed Certificate Authority, which in turn automatically creates a signed certificate from the certificate template for each host. For details on how to view the Certificate Authority and signed certificate, see “Viewing Trusted Certificate Authorities” on page 102.

11.3.2 Enabling SSL for an App Server

After creating a certificate template, you can enable SSL for an HTTP, ODBC, WebDAV, or XDBC server.

1. Click the Groups icon in the left frame.

2. Click the group in which you want to define the HTTP server (for example, Default).

3. Click the App Servers icon on the left tree menu.

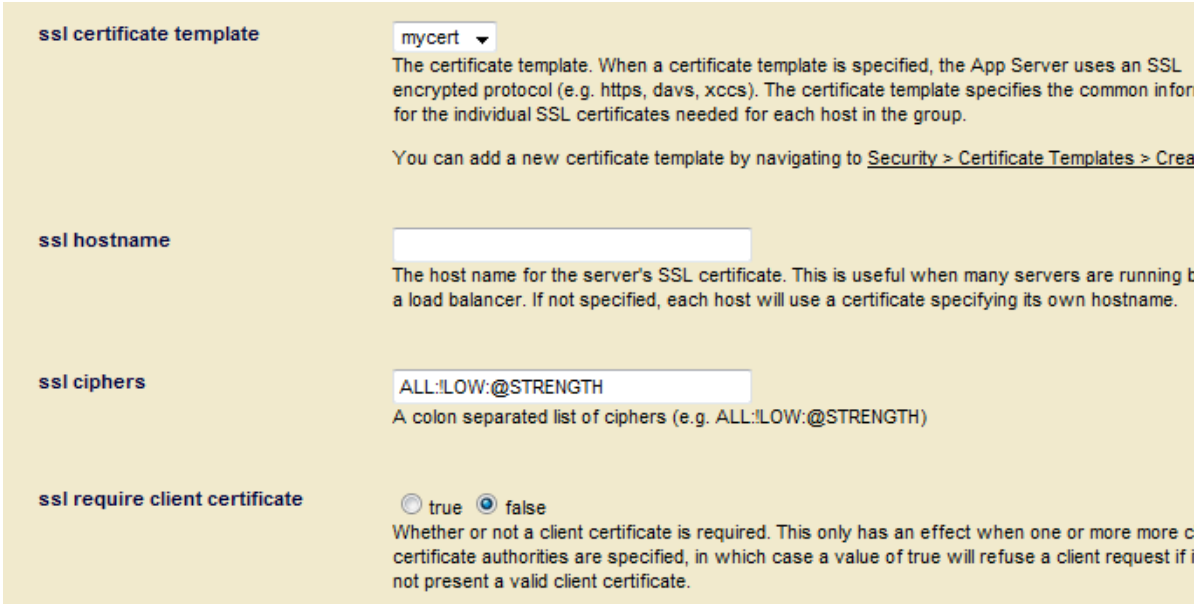
4. Either create a new server by clicking on one of the Create *server_type* tabs or select an existing server from the left tree menu.

   The SSL fields are located at the bottom of the server specification page.

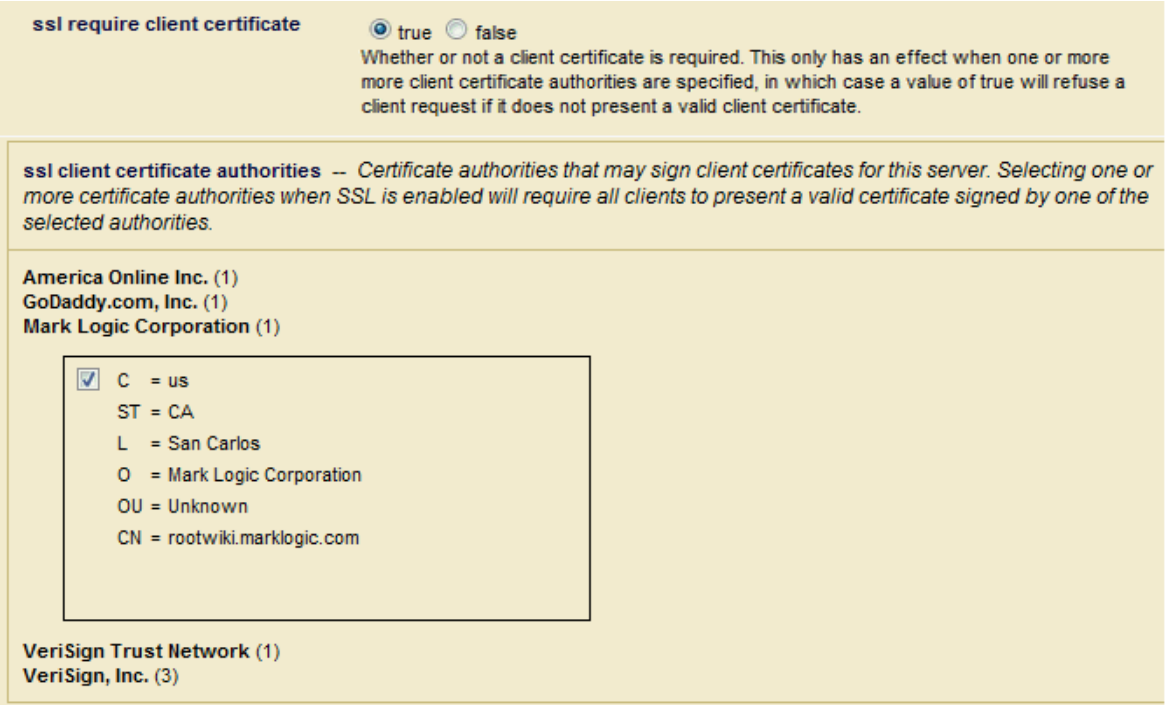
5. In the SSL Certificate Template field, select the certificate template you created in "Creating a Certificate Template" on page 81. Selecting a certificate template implicitly enables SSL for the App Server.

6. (Optional) The SSL Hostname field should only be filled in when a proxy or load balancer is used to represent multiple servers. In this case, you can specify an SSL hostname here and all instances of the application server will identify themselves as that host.

7. (Optional) In the SSL Ciphers field, you can either use the default (`ALL:!LOW:@STRENGTH`) or one or more of the SSL ciphers defined in `http://www.openssl.org/docs/apps/ciphers.html`.



8. (Optional) If you want SSL to require clients to provide a certificate, select True for SSL Require Client Certificate. Then select Show under SSL Client Certificate Authorities and which certificate authority is to be used to sign client certificates for the server:



11.4 Accessing an SSL-Enabled Server from a Browser or WebDAV Client

When you create a certificate template and set it in your App Server, MarkLogic Server automatically generates a temporary self-signed MarkLogic certificate authority that signs host certificates. If you have not yet received a signed certificate for your SSL-enabled App Server from a certificate authority, your browser must accept the temporary self-signed certificate authority before it can access the App Server. There are two alternative ways to do this, both of which are browser-dependent and described below.

To enable WebDAV clients to access an SSL-enabled App Server, you must follow the procedure described in “Importing a Self-Signed Certificate Authority into Windows” on page 92.

To enable a single browser to access the SSL-enabled App Server, you can create a security exception for the self-signed certificate in your browser, as described in the following sections:

- Creating a Security Exception in Internet Explorer
- Creating a Security Exception in Mozilla Firefox

If you need to enable a number of browsers to access the SSL-enabled App Server, you might want each browser to import the self-signed certificate authority for the certificate template. Once this is done, all certificates signed by the certificate authority will be trusted by the browser, so you can distribute new certificates without requiring each browser to create new security exceptions. The following sections describe how to import the self-signed MarkLogic certificate authority:

- Importing a Self-Signed Certificate Authority into Windows
- Importing a Self-Signed Certificate Authority into Mozilla Firefox

11.4.1 Creating a Security Exception in Internet Explorer

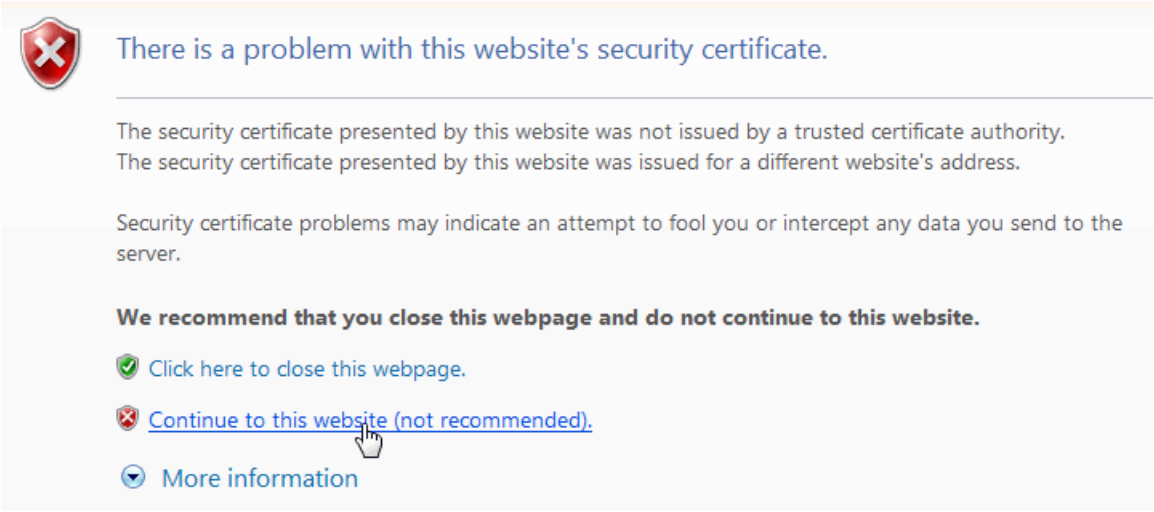
If you have not imported the certificate authority for the certificate template into Windows, when you first access an SSL-enabled server with your IE browser, you will receive an error notifying you that there is a problem with this website's security certificate. You can bypass this security exception by accepting the certificate. For example, if you enabled SSL on the HTTP server, App-Services, each host can accept the self-signed certificate as described below.

1. Access the server with the URL:

```
https://gordon-1:8000/
```

Note: Remember to start your URL with HTTPS, rather than HTTP. Otherwise, the browser will return an error.

2. The server responds with a "There is a problem with this website's security certificate" notification similar to:



3. Click on "Continue to this website (not recommended)"

4. Enter your MarkLogic Server username and password at the prompt.

11.4.2 Creating a Security Exception in Google Chrome

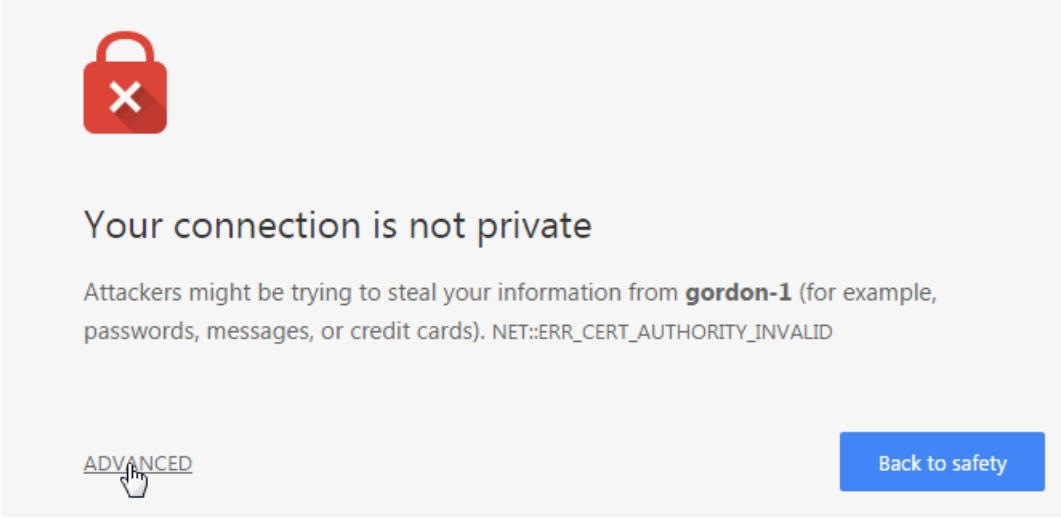
If you have not imported the MarkLogic certificate authority into your Chrome browser, when you first access an SSL-enabled server, you will receive an error notifying you that you have accessed an untrusted server. You can bypass this security exception by accepting the certificate. For example, if you enabled SSL on the HTTP server, App-Services, you can accept the self-signed certificate as described below.

1. Access the server with the URL:

```
https://gordon-1:8000/
```

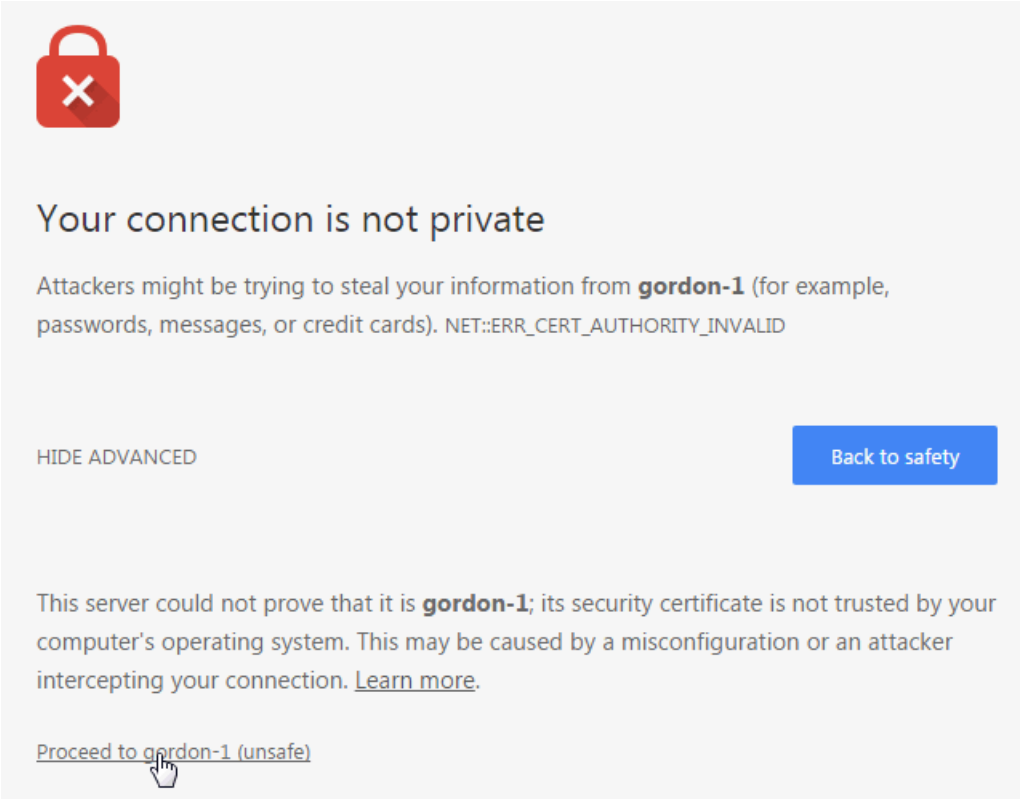
Note: Remember to start your URL with HTTPS, rather than HTTP. Otherwise, the browser will return an error.

2. The server responds with a "Your connection is not private" notification similar to:



3. Click on "Advanced."

4. At the bottom of the expanded window, select "Proceed to *hostname* (unsafe)."



5. Enter your MarkLogic Server username and password at the prompt.

11.4.3 Creating a Security Exception in Mozilla Firefox

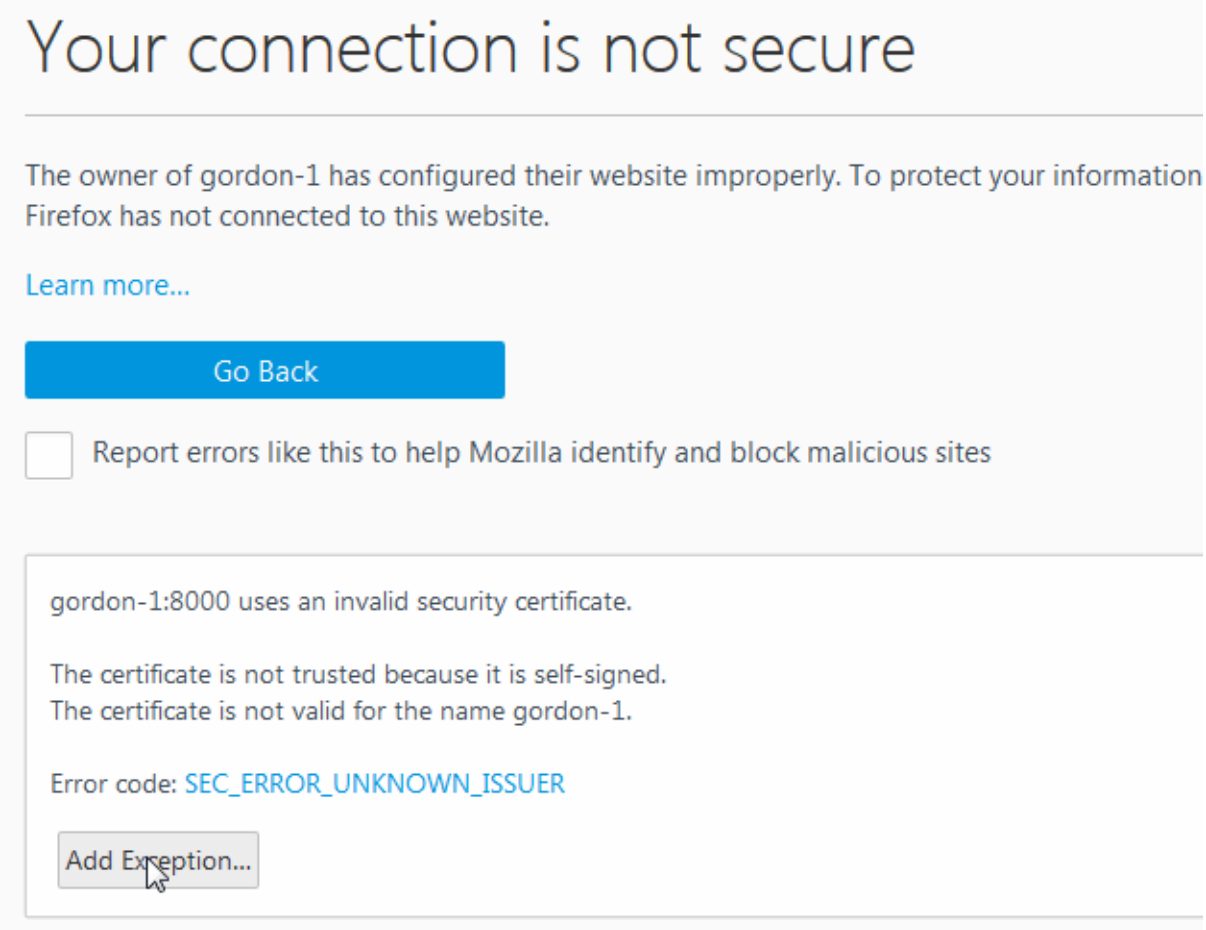
If you have not imported the MarkLogic certificate authority into your Firefox browser, when you first access an SSL-enabled server, you will receive an error notifying you that you have accessed an untrusted server. You can bypass this security exception by accepting the certificate. For example, if you enabled SSL on the HTTP server, App-Services, you can accept the self-signed certificate as described below.

1. Access the server with the URL:

```
https://gordon-1:8000/
```

Note: Remember to start your URL with HTTPS, rather than HTTP. Otherwise, the browser will return an error.

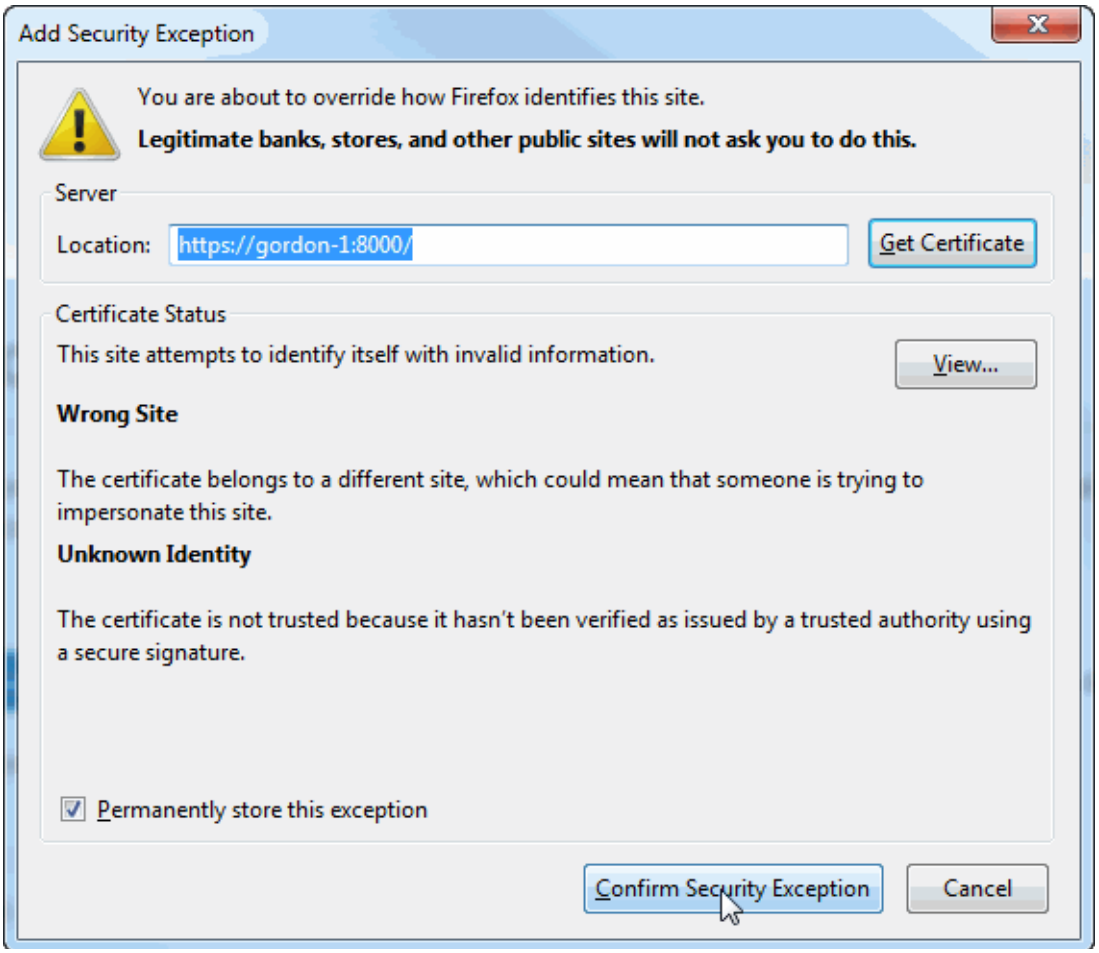
2. The server responds with an "Your connect is not secure" notification similar to:



Note: If you see another type of error message, see "What to do if you don't get an 'Or you can add an exception' Prompt" on page 90.

3. Click "Or you can add an exception."

4. In the Add Security Exception page, click Permanently store this exception and click Confirm Security Exception.

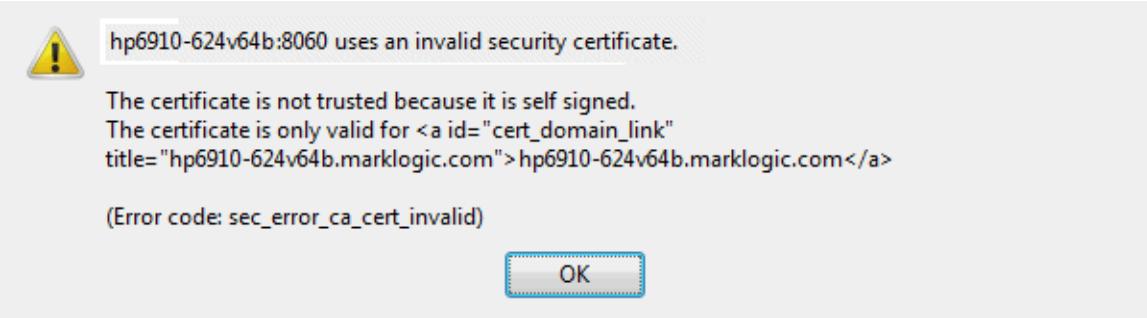


5. In the Certificate Manager page, click OK.

6. The server then prompts you for a username and password before connecting you to the server.

11.4.3.1 What to do if you don't get an 'Or you can add an exception' Prompt

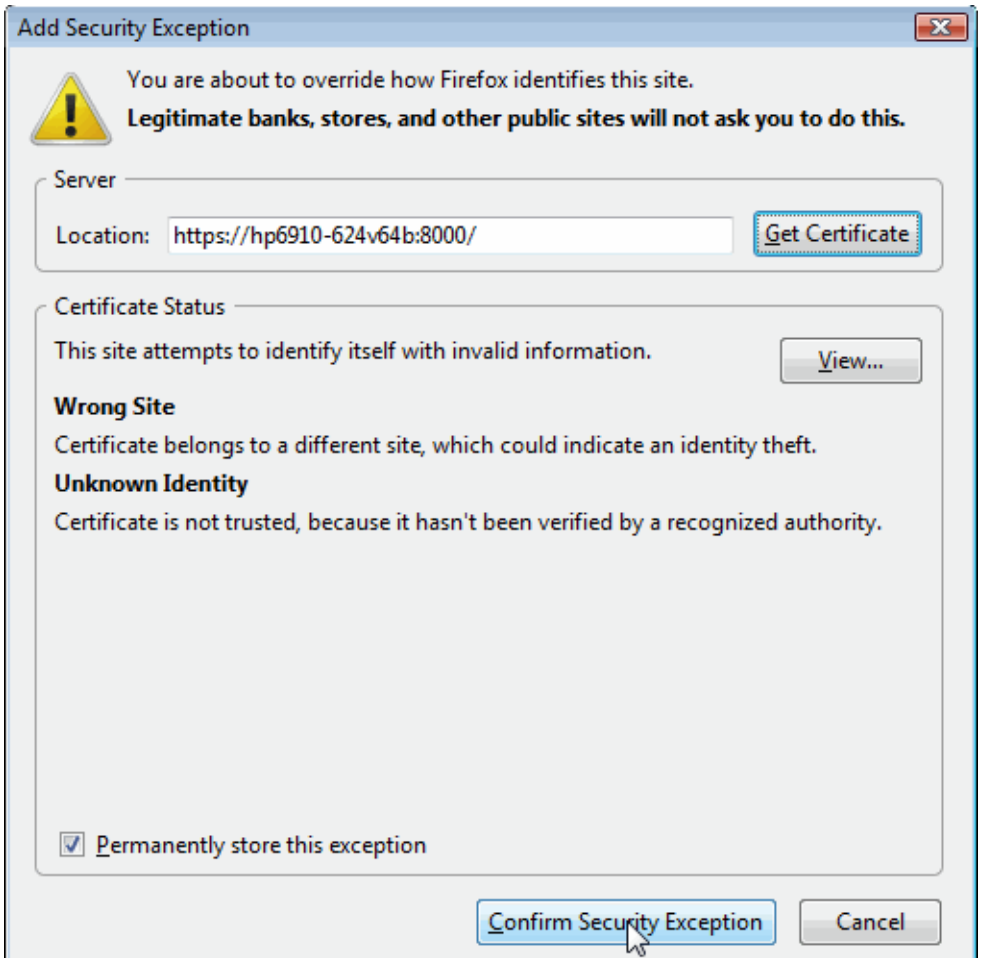
When using Mozilla Firefox, you may encounter an error message that does not allow you the option to add an exception for the self-signed certificate. This type of error looks like:



In this case, click OK and follow this procedure:

1. In your browser, navigate to Tools > Options.
2. Click Advanced and click the Encryption tag.
3. Click View Certificates.
4. In the Certificate Manager, click the Servers tab.
5. In the Add Security Exception page, click Add Exception
6. Enter the URL, including the host and port, to the SSL-enabled server in the Location field.

7. Click Get Certificate and click Confirm Security Exception.



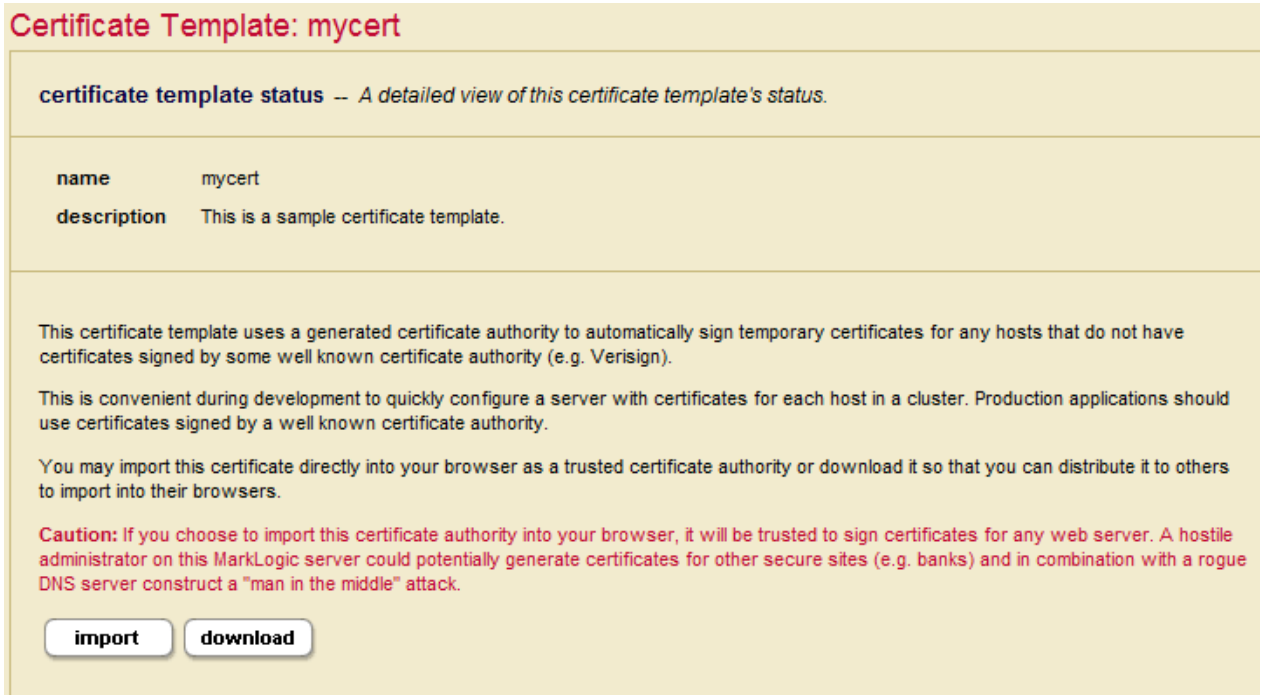
8. In the Certificate Manager, click OK.

You should now be able to access the SSL-enabled server.

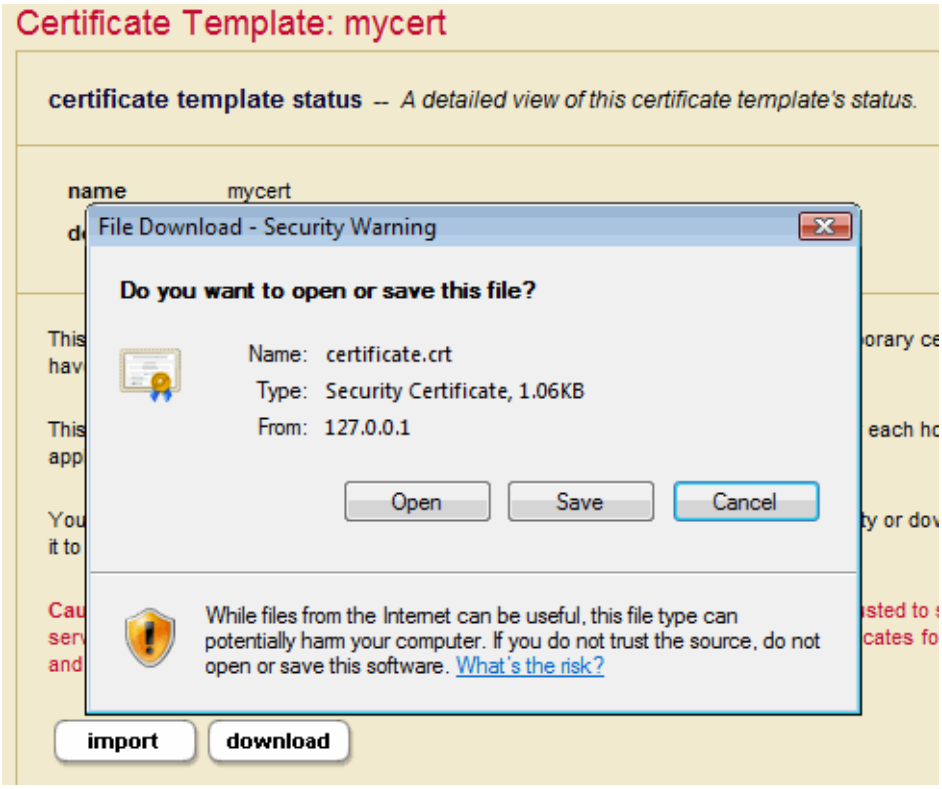
11.4.4 Importing a Self-Signed Certificate Authority into Windows

This section describes how to import the Certificate Authority into Windows for use by the Internet Explorer browser and WebDAV clients.

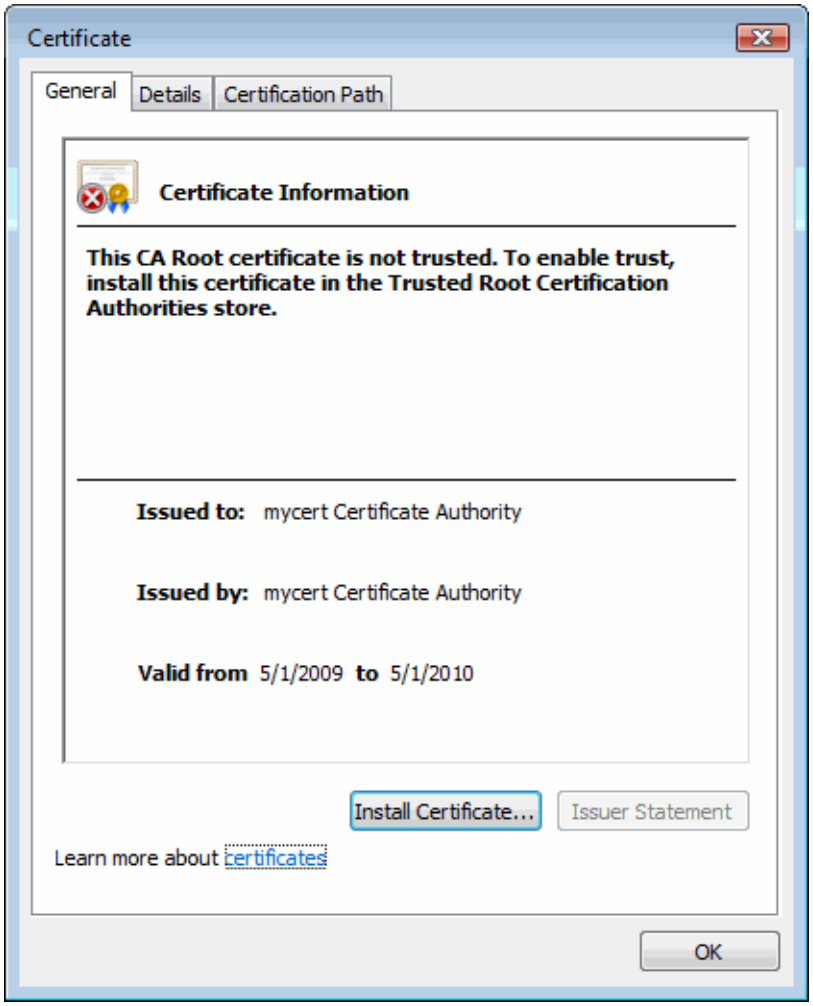
1. Open the Admin interface in your Internet Explorer browser.
2. Click the Security icon in the left frame.
3. Click the Certificate Templates icon on the left tree menu.
4. Click the certificate template name on the left tree menu. The Configure certificate template page will display.
5. Click the Status tab to display the certificate template Status page.
6. Click on Import.



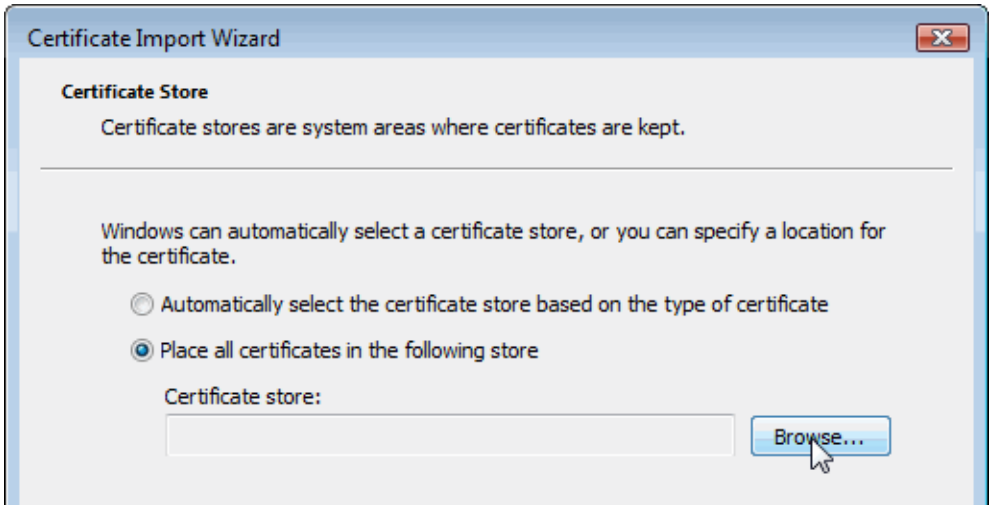
7. In the “Do you want to open or save this file?” window, click Open.



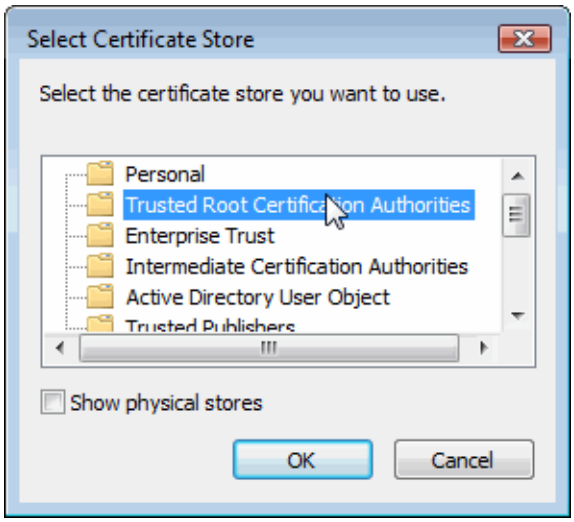
8. In the “Certificate Information” window, click Install Certificate.



9. In the Certificate Import Wizard window, select “Place all certificates in the following store” and click Browse.

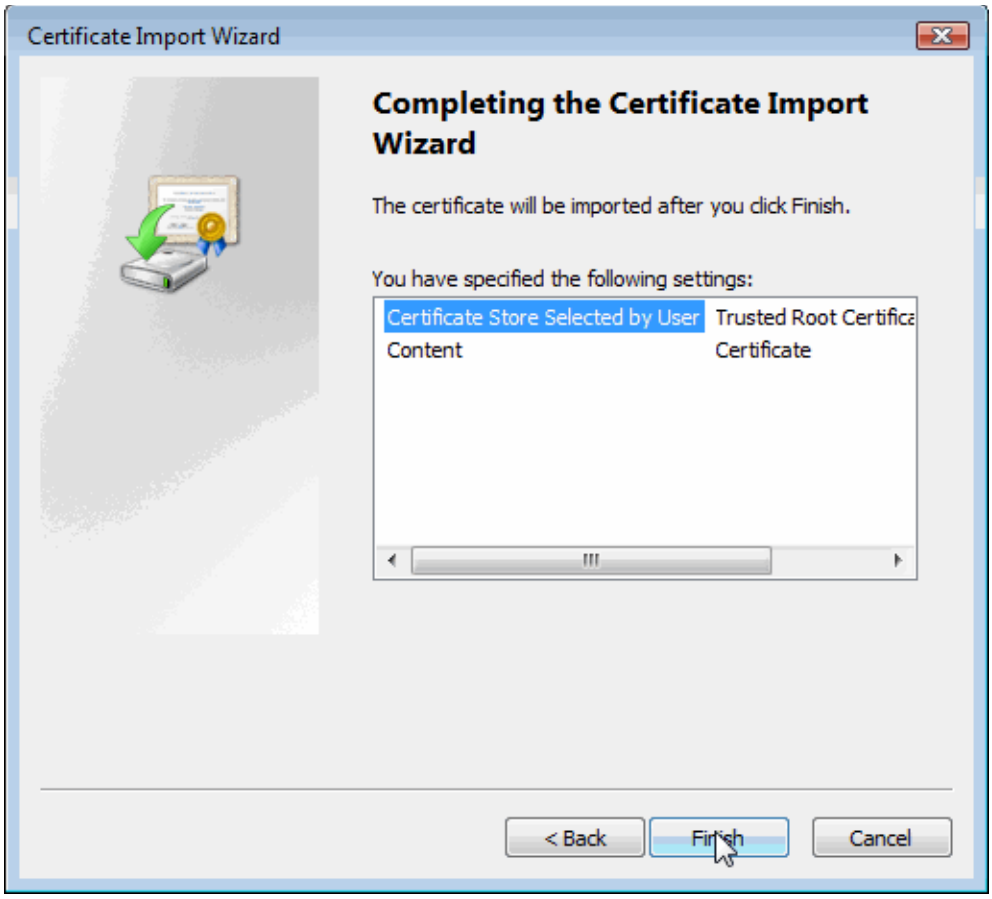


10. In the Select Certificate Store window, select “Trusted Root Certification Authorities” and click OK.

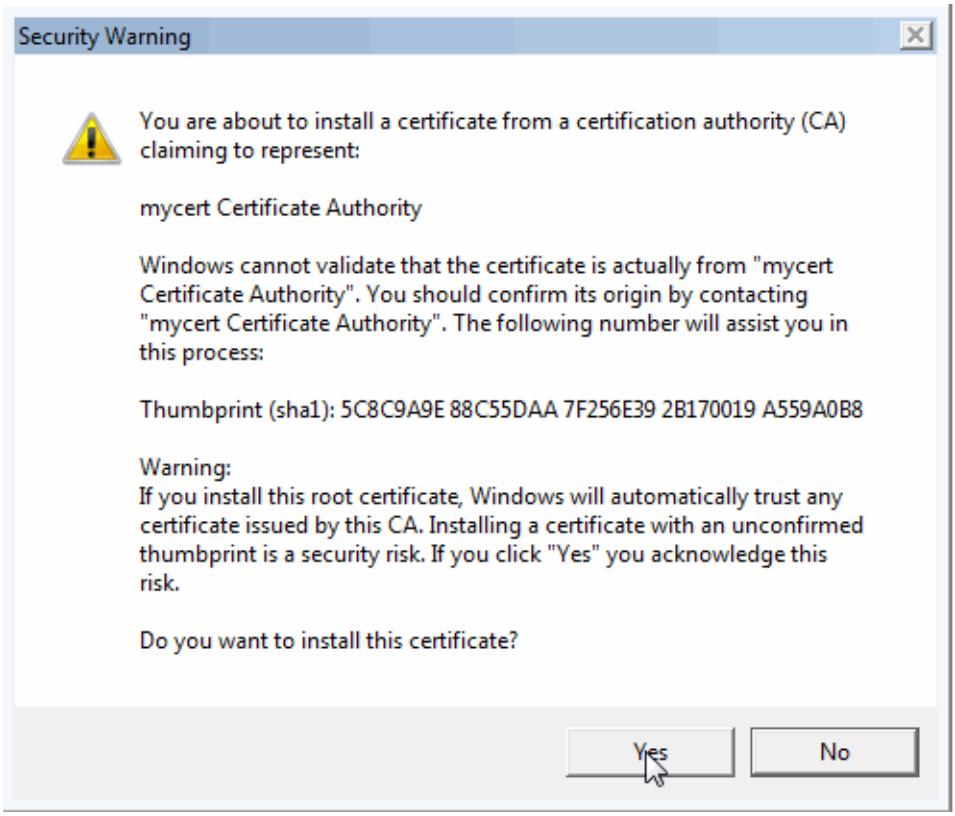


11. In the Certificate Import Wizard window, click Next.

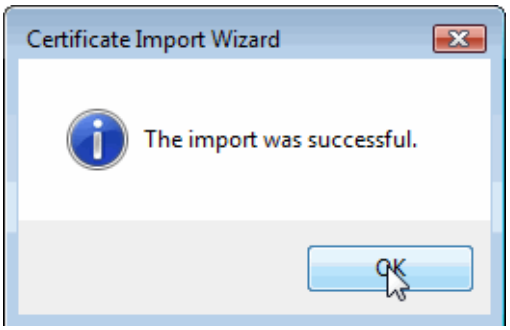
12. On the Completing the Certificate Import Wizard page, select “Certificate Store Selected by User” and click Finish



13. In the Security Warning page, click Yes.



14. When you see “The import was successful prompt,” click OK.



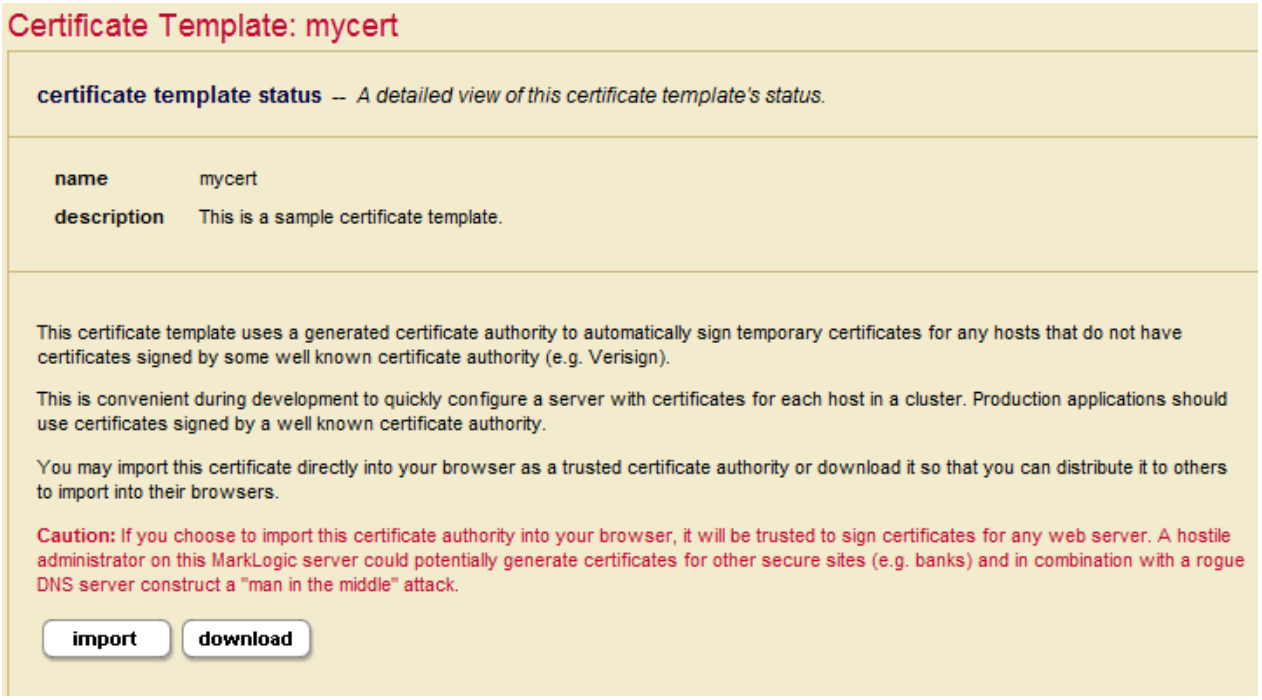
15. In the Certificate Information window, click OK to exit.

You should now be able to access the SSL-enabled server from your Internet Explorer browser or WebDAV client.

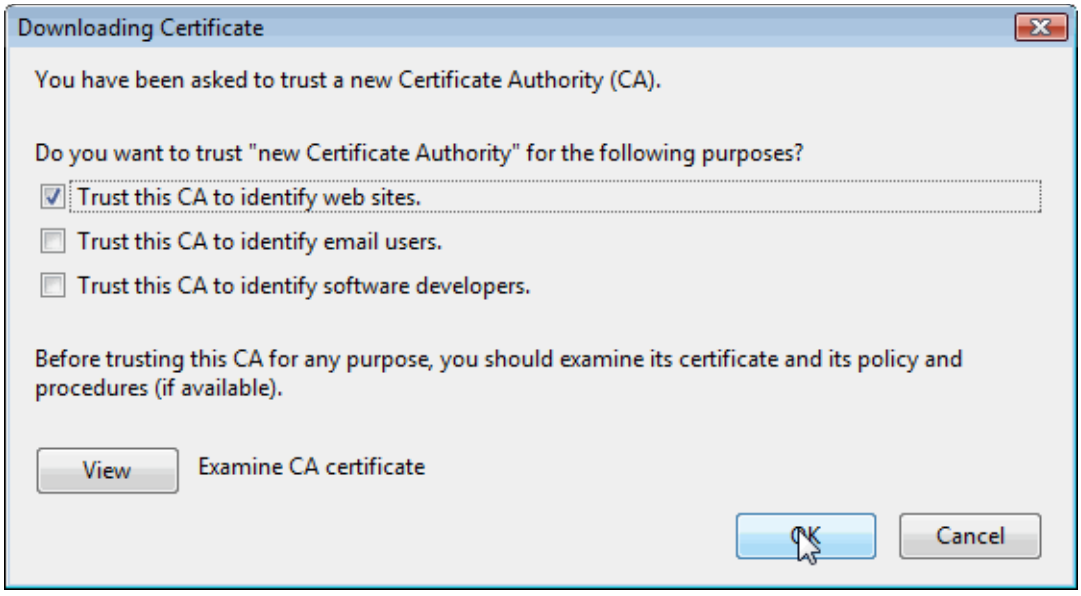
11.4.5 Importing a Self-Signed Certificate Authority into Mozilla Firefox

This section describes how to import the Certificate Authority into your Mozilla Firefox browser.

1. Open the Admin interface in your Firefox browser.
2. Click the Security icon in the left frame.
3. Click the Certificate Templates icon on the left tree menu.
4. Click the certificate template name on the left tree menu. The Configure certificate template page will display.
5. Click the Status tab to display the certificate template Status page.
6. Click on Import.



7. Select “Trust this CA to identify web sites” in the “Downloading Certificate” window and click OK:



You should now be able to access the SSL-enabled server from your Mozilla Firefox browser.

11.5 Procedures for Obtaining a Signed Certificate

Use the following procedures to obtain a signed certificate and import into your server:

- Generating and Downloading Certificate Requests
- Importing a Signed Certificate into MarkLogic Server

Note: No outside authority is used to sign certificates used between servers communicating over the internal XDQP connections in a cluster. Such certificates are self-signed and trusted by each server in the cluster. For details, see “Enabling SSL communication over XDQP” on page 34.

11.5.1 Generating and Downloading Certificate Requests

Once the server is created or modified with SSL enabled, you can generate one or more PEM-encoded certificate requests.

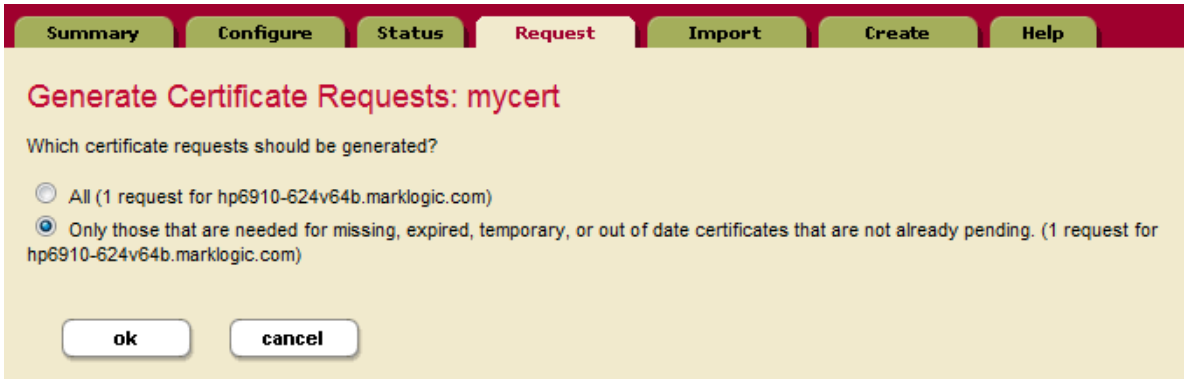
Note: You must first assign the certificate template to an App Server, as described in “Enabling SSL for an App Server” on page 83, before you can generate a certificate request.

1. Click the Security icon in the left frame.

2. Click the Certificate Templates icon on the left tree menu.

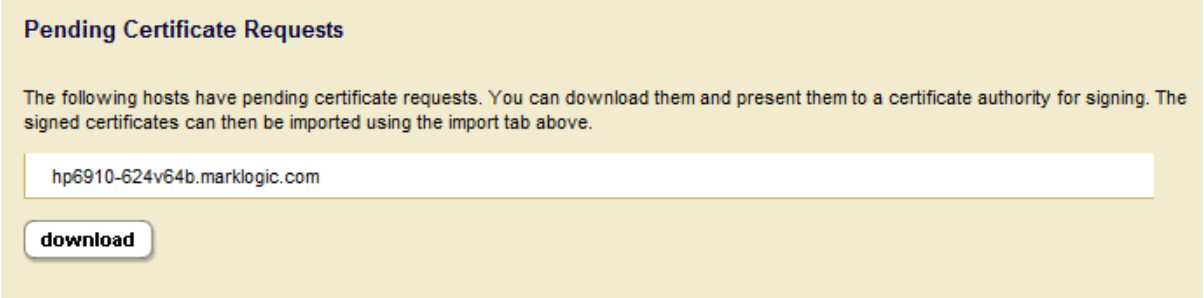
3. Click the certificate template name on the left tree menu. The Configure certificate template page will display.

4. Click the Request tab. The Generate Certificate Request page will display:



5. Select either “All” or “Only those that are needed for missing, expired, self-signed, or out of date certificates that are not already pending,” then click OK.

6. The certificate template Status page will display. Click on Download to download the certificate request to your file system.



7. If the file does not already have a ‘zip’ extension, rename the file by replacing the ‘xqy’ extension with ‘zip’.

8. Send the zip file containing the certificate requests to a Certificate Authority, such as Verisign.

11.5.2 Importing a Signed Certificate into MarkLogic Server

When you receive the PEM file(s) containing signed certificate(s) from the certification authority, import the PEM file(s) into MarkLogic Server. If you are using chained certificates, you will need to import the end and intermediate certificate PEM files into MarkLogic Server. If your MarkLogic Server is to act as a client , you must also import the root certificate.

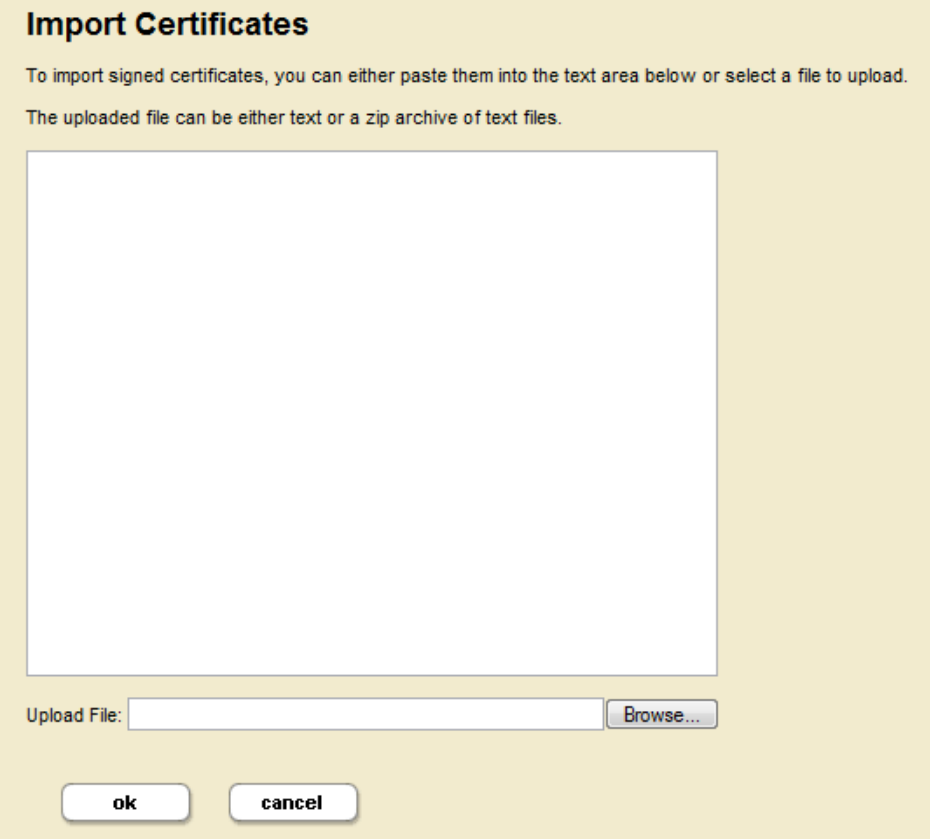
Note: Because the signed certificate is from a trusted certification authority, browsers are already configured to trust the certificate.

1. Click the Security icon in the left frame.

2. Click the Certificate Templates icon on the left tree menu.

3. Click the certificate template name on the left tree menu. The Configure certificate template page will display.

4. Click the Import tab. The Import Certificates page will display:



5. Click on Browse to locate the PEM file(s) containing the signed certificate(s) and select OK. Zip files can be uploaded directly without the need to unzip them. Alternatively, you can paste an individual certificate(s) into the text area.

11.6 Viewing Trusted Certificate Authorities

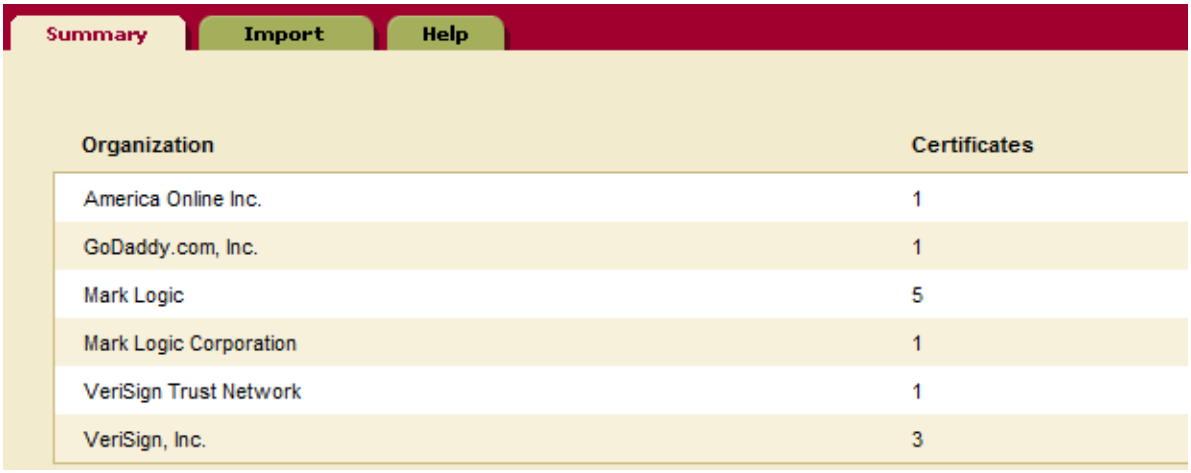
You can list all of the certificate authorities that are known to and trusted by the server in the Certificate Authority page. Each CA in the list links to the corresponding Certificate Authority page for that CA.

The Certificate Authority page provides detailed information on the CA, a list of revoked certificates, the option to manually revoke a certificate by ID, and the ability to delete the CA from the server.

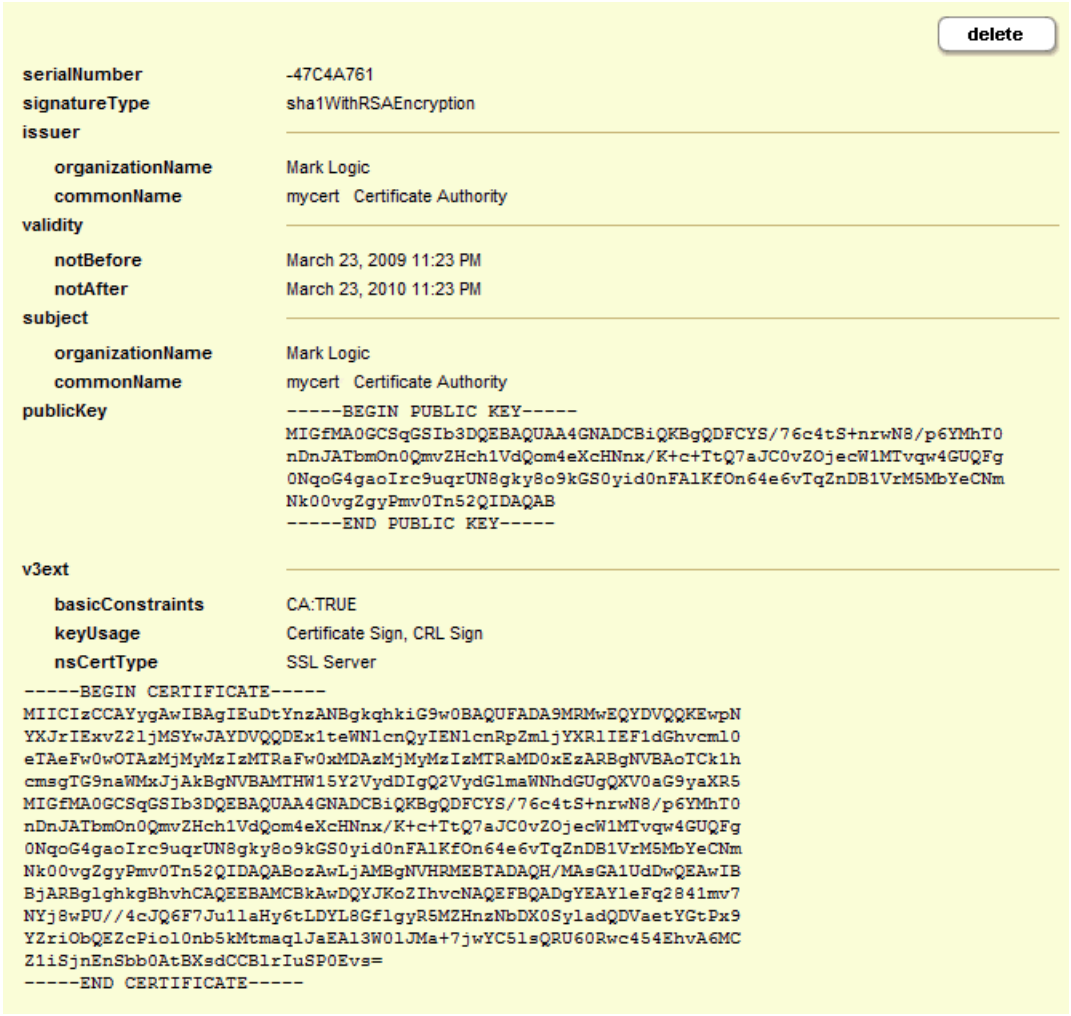
1. Click the Security icon in the left frame.

2. Click the Certificate Authority icon on the left tree menu.

3. The Certificate Authority Summary page displays the list of trusted CAs:



4. Click on a CA in the list to display the details on the CA:



11.7 Importing a Certificate Revocation List into MarkLogic Server

A Certificate Revocation List (CRL) is a list of certificate serial numbers that have been revoked by a certificate authority. The CRL is signed by the certificate authority to verify its accuracy. The CRL contains the revocation date of each certificate, along with the date the CRL was published and the date it will next be published, which is useful in determining whether a newer CRL should be fetched.

You can use the `pki:insert-certificate-revocation-list` function to import a CRL into the Security database. certificate authorities typically allow the CRL to be downloaded via HTTP. The document URL in the database is derived from the URL passed in to the function, so Inserting a newer CRL retrieved from the same URL will replace the previous one in the database.

For example, the following script imports a PEM- or DER-encoded CRL from Verisign into the Security database:

```
xquery version "1.0-ml";
import module namespace pki = "http://marklogic.com/xdmp/pki"
         at "/MarkLogic/pki.xqy";

let $URI := "http://crl.verisign.com/pca3.crl"

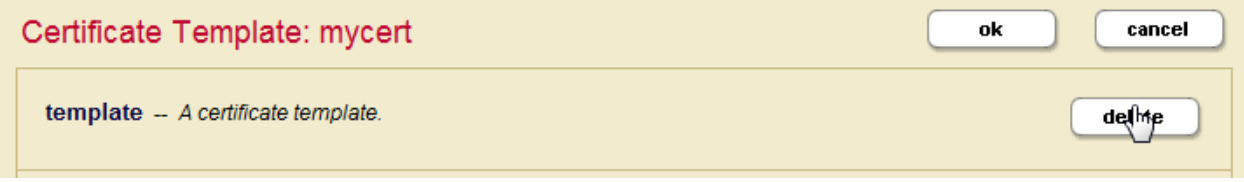
return
    pki:insert-certificate-revocation-list(
         $URI,
         xdmp:document-get($URI)/binary() )
```

11.8 Deleting a Certificate Template

Deleting a template deletes all signed certificates and pending requests for the template. Before deleting a certificate template, ensure that a certificate with that name is not in use by a server. If a certificate with the same name as the certificate template is in use by a server, the delete operation returns an “Invalid input” error.

To delete an unused certificate template:

1. Click the Security icon in the left frame.
2. Click the Certificate Templates icon on the left tree menu.
3. Click the certificate template name on the left tree menu.
4. On the Certificate Template page, click Delete:



5. In the confirmation page, select OK.

12.0 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 chapter describes the auditing features and includes the following parts:

- Overview of Auditing
- Auditable Events
- Audit Log Content

12.1 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 and includes the following parts:

- Audit Log Files
- Restricting Audit Events
- Audit Successful, Unsuccessful, or Both Types of Events
- Enabled at the Group Level

12.1.1 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 “Auditable Events” on page 108. 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 “Sample Audit Logs” on page 113.
- You can configure how often to rotate the audit files (similar to the log files, as described in “Log Files” on page 386).
- 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	`C:\Program Files\MarkLogic\Data\Logs\AuditLog.txt`
Red Hat Linux	`/var/opt/MarkLogic/Logs/AuditLog.txt`
Sun Solaris	`/var/opt/MARKlogic/Logs/AuditLog.txt`
Mac OS X	`~Library/Application Support/MARKlogic/Data/Logs/AuditLog.txt`

12.1.2 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 that 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 “Configuring Auditing to Audit Certain Events and Set Up Certain Restrictions” on page 115.

12.1.3 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 “Configuring Auditing to Audit Certain Events and Set Up Certain Restrictions” on page 115.

12.1.4 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 "Enabling Auditing for a Group" on page 114.

12.2 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 a 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
`authentication-failure`	Audits failed authentication attempts.	N/A	Yes	Failure Only

Event	Description	URI Restrictions	Role/User Restrictions	Success or Failure Restrictions
`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-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
`estimate`	Audits when an `xdmp:estimate` 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

Event	Description	URI Restrictions	Role/User Restrictions	Success or Failure Restrictions
`exists`	Audits when an `xdmp:exists` expression is evaluated.	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
`lexicon-read`	Audits when a value lexicon (for example, `cts:element-values`) call is used.	N/A	Yes	Success Only
`no-permission`	Audits when an operation fails because of a `SEC-PERMDENIED` 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
`permissions-change`	Audits when permissions on a document are modified.	Yes	Yes	Yes
`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 an attempt to add or remove a role from a user fails.	N/A	Yes	Failure Only

Event	Description	URI Restrictions	Role/User Restrictions	Success or Failure Restrictions
`search`	Audits when a `cts:search` expression is evaluated.	N/A	Yes	Success Only
`security-access`	Audits when one of the following security-related functions are called: `xdmp:can-grant-roles`, `xdmp:has-privilege`, `xdmp:user-roles`, `xdmp:role-roles`, `xdmp:privilege-roles`, `xdmp:amp-roles`, `xdmp:get-current-role`, `xdmp:user`, `xdmp:role`, `xdmp:amp`.	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
`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

Event	Description	URI Restrictions	Role/User Restrictions	Success or Failure Restrictions
`user-role-addition`	Audits when a role is added to a user.	N/A	Yes	Yes
`user-role-removal`	Audits when a role is removed from a user.	N/A	Yes	Yes

12.2.1 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 “Auditable Events” on page 108.	`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`

Log Entry Field	Description	Example
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`

12.2.2 Sample Audit Logs

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

```
2012-03-26 10:55:53.735 event=amp-usage;
function=http://marklogic.com/xdmp/admin:read-config-file;
uri=/MarkLogic/admin.xqy; database=filesystem;
success=true; user=admin; roles=admin;


2012-03-23 15:04:32.141 event=security-access; user(admin-xxx);
success=true;
user=admin-xxx; roles=admin;


2012-03-23 15:04:32.176 event=document-read;
uri=http://marklogic.com/xdmp/privileges/7768816571421711260;
database=Security; success=true; user=admin-xxx; roles=admin;


2012-03-23 15:05:26.957 event=lexicon-read;
expr=cts:element-value-query(xs:QName("info:status"),
("active", "unloading"), ("unstemmed","lang=en"), 1);
database=App-Services; success=true; user=infostudio-admin;
roles=cpf-restart,infostudio-user;


2012-03-23 15:07:05.164 event=document-update; type=node-update;
uri=/workspaces/14061174091930893017.xml; database=App-Services;
success=true; user=admin-xxx; roles=admin;


2012-03-23 15:07:05.674 event=document-insert; type=insert;
uri=/queries/5523898374388210414.txt; database=App-Services;
success=true; user=admin-xxx; roles=admin;
```

12.3 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 “Groups” on page 30. This section describes the following audit configuration procedures:

- Enabling Auditing for a Group
- Disabling Auditing for a Group
- Configuring Auditing to Audit Certain Events and Set Up Certain Restrictions

12.3.1 Enabling Auditing for a Group

Perform the following steps to enable auditing for a group:

1. Access the Admin Interface with a browser.
2. Open the Audit Configuration screen (Groups > *group_name* > Auditing).
3. Select True for the Audit Enabled radio button.
4. Configure any audit events and/or audit restrictions you want.
5. Click OK.

12.3.2 Disabling Auditing for a Group

Perform the following steps to disable auditing for a group:

1. Access the Admin Interface with a browser.
2. Open the Audit Configuration screen (Groups > *group_name* > Auditing).
3. Select False for the Audit Enabled radio button.
4. Click OK.

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

12.3.3 Configuring Auditing to Audit Certain Events and Set Up Certain Restrictions

The following is the general procedure for configuring audit events and audit restrictions. Your procedure will vary depending on what events and restrictions you choose to configure.

1. Access the Admin Interface with a browser.

2. Open the Audit Configuration screen (Groups > *group_name* > Auditing).

3. Under Audit Events, choose the events you want audited. For a description of each event, see “Auditable Events” on page 108.

4. Under Audit Restrictions, enter any restrictions you want. For details on audit restrictions, see “Restricting Audit Events” on page 107.

5. Click OK to save your changes.

13.0 Managing User Sessions and Monitoring Login Attempts

MarkLogic Server provides facilities to control and manage user sessions and monitoring login attempts. This chapter describes how to use and mange these features and includes the following parts:

- Managing Concurrent User Sessions
- Setting Request Blackouts on an App Server
- Storing and Monitoring the Last User Login Attempt

13.1 Managing Concurrent User Sessions

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

- Limiting Concurrent Requests with User Session Limits
- Configuring User Concurrent Session Controls

13.1.1 Limiting Concurrent Requests with User Session 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 defined to be 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.

13.1.2 Configuring User Concurrent Session Controls

To configure a user concurrent session limit, perform the following steps in the Admin Interface:

1. Click the Groups icon.
2. Click the group in which the App Server you want to configure resides (for example, Default).
3. Click the App Servers icon on the left tree menu.

4. Select the App Server in which you want to configure concurrent session limits. The App Server Configuration page displays.

5. In the `concurrent request limit` field, enter a value corresponding to the maximum number of concurrent user sessions you want to allow. For example, if you want only 3 concurrent sessions, 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.

13.2 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.

- Configuring Request Blackouts
- Deleting Request Blackouts

13.2.1 Configuring Request Blackouts

Perform the following to configure request blackout periods:

1. In the Admin Interface tree menu, click the Groups > *group_name* > App Servers > *app_server_name* link, where *group_name* is the name of the group and *app_server_name* is the name of the App Server in which you want to specify a request blackout period.

2. Click the Request Blackout menu item under your App Server. The Request Blackout Policy Configuration page appears.

3. Click the Create tab. The Add Request Blackout page appears.

4. 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.

5. Click OK to create the blackout period.

The new blackout period will take effect immediately.

13.2.2 Deleting Request Blackouts

Perform the following to delete a request blackout period:

1. In the Admin Interface tree menu, click the Groups > *group_name* > App Servers > *app_server_name* link, where *group_name* is the name of the group and *app_server_name* is the name of the App Server in which you want to specify a request blackout period.

2. Click the Request Blackout menu item under your App Server. The Request Blackout Policy Configuration page appears.

3. In the area corresponding to the blackout period you want to delete, click the Delete button.

4. Click OK on the confirmation page to delete the blackout period.

The blackout period is deleted immediately.

13.3 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, and contains the following parts:

- Storing Last User Login Information in a Last-Login Database
- Configuring User Login Monitoring
- Displaying the Last Login Information for an App Server or for the Admin Interface

13.3.1 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.

13.3.2 Configuring User Login Monitoring

Perform the following steps to set up user login monitoring for a given App Server.

1. Click the Groups icon.
2. Click the group in which the App Server you want to configure resides (for example, Default).
3. Click the App Servers icon on the left tree menu.
4. Select the App Server in which you want to configure the last-login database. The App Server Configuration page displays.
5. Select a database for the Last Login 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, select `true` on the Display Last Login radio button.
7. Click OK to save the changes.

13.3.3 Displaying the Last Login Information for an App Server or for the Admin Interface

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
```

14.0 Databases

This section introduces basic database management procedures. Later sections in this guide introduce some concepts for tuning the performance of your databases. For information on database backup and restore operations, see “Backing Up and Restoring a Database” on page 234. The following topics are included:

- Understanding Databases
  - Schemas and Security Databases
  - Modules Database
  - Triggers Database
  - Database Settings
  - Example of Databases in MarkLogic Server
- Creating a New Database
- Attaching and/or Detaching Forests to/from a Database
- Viewing Database Settings
- Loading Documents into a Database
- Merging a Database
- Reindexing a Database
- Clearing a Database
- Disabling a Database
- Deleting a Database
- Checking and Setting Permissions for a Document in a Database

This chapter 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*.

14.1 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 “Understanding Forests” on page 274 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.

14.1.1 Schemas and Security Databases

The installation process creates five *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.

14.1.2 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 “Directories” on page 55 and “Server Root Directory” on page 57.

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 “Security Administration” on page 292 and the chapters related to security in the *Application Developer’s Guide*.

14.1.3 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*.

14.1.4 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. You can configure the following basic types of settings for each database:

- Basic Administrative Settings
- Index Settings that Affect Documents
- Rebalancer Settings
- Reindexing Settings
- Document and Directory Settings
- Memory and Journal Settings
- Other Settings
- Merge Control Settings

14.1.4.1 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
`database name`	The name of the database.
`security database`	The name of the security database which this database accesses.
`schema database`	The name of the schemas database which this database accesses.
`triggers database`	The name of the triggers database which this database accesses.

14.1.4.2 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" on page 321.

In general, adding index options will have the effect of slowing document loading and increasing the size of database files.

Database Setting	Description
`language`	Specifies the default language for content in this database. Any content without an `xml:lang` 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`	Stemmed word searches enabled. 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 `be` will also match the term `is`). For more details on stemmed searches, see Understanding and Using Stemmed Searches in the *Search Developer's Guide*.
`word searches`	Unstemmed word searches enabled. Enables searches for exact matches of words.
`word positions`	Index word positions for faster phrase and `cts:near-query` 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, `sem:sparql` can be used, but document loading is slower and the database files are larger. **Note:** This feature requires a valid semantics license key.
`triple positions`	Specifies whether to index positional data to speed up the performance of proximity queries that use the `cts:triple-range-query` function.
`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.

Database Setting	Description
`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 `cts:near-query` 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 `cts:near-query` searches that use `cts:element-value-query`.
`attribute value positions`	Index attribute word positions for faster attribute-based phrase and `cts:near-query` searches that use `cts:element-value-query` and faster `cts:element-query` searches that use a `cts:element-attribute-*-query`.
`field value searches`	Enables searches that use `cts:field-value-query`.
`field value positions`	Enables positions for searches that use `cts:field-value-query`.
`three character searches`	Enables 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 `three character search` index with a codepoint collation word lexicon. For more details about wildcard searches, see Understanding and Using Wildcard Searches in the *Search Developer's Guide*.
`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 *Search Developer's Guide*.
`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 *Search Developer's Guide*.
`trailing wildcard word positions`	Index word positions for trailing wildcard searches.

Database Setting	Description
`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 `three character search` index to speed wildcard searches. For more details about wildcard searches, see Understanding and Using Wildcard Searches in the *Search Developer's Guide*.
`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 `three character searches` and a word lexicon. For more details about wildcard searches, see Understanding and Using Wildcard Searches in the *Search Developer's Guide*.
`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 `three character searches` and a word lexicon. For more details about wildcard searches, see Understanding and Using Wildcard Searches in the *Search Developer's Guide*.
`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 `cts:uris` or `cts:uri-match` 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 `cts:collections` or `cts:collection-match` APIs.

Contains SAP BusinessObjects Text Analysis XI from SAP AG. Copyright © 1996-2011. All rights reserved.

14.1.4.3 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 “Database Rebalancing” on page 185.

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 “Rebalancer Document Assignment Policies” on page 187.
`rebalancer enable`	When set to `true`, the database rebalancer will automatically redistribute the content across the database forests. When set to `false`, rebalancing is disabled.
`rebalancer throttle`	Sets the priority of system resources devoted to rebalancing. Higher numbers give rebalancing a higher priority.

14.1.4.4 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 `true`, index configuration changes automatically initiate a background reindexing operation on the entire database. When set to `false`, 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 `reindexer enable` to `true`. For information on how the reindexer effects queries, see “Query Behavior with Reindex Settings Enabled and Disabled” on page 337.

Contains SAP BusinessObjects Text Analysis XI from SAP AG. Copyright © 1996-2011. All rights reserved.

Database Setting	Description
`reindexer throttle`	Sets the priority of system resources devoted to reindexing. Higher numbers give reindexing a higher priority.
`reindexer timestamp`	Specifies the timestamp of fragments to force a reindex/refragment operation. Click the `get current timestamp` button to enter the current system timestamp. When you set this parameter to a timestamp and `reindex enable` is set to `true`, 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.

14.1.4.5 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
`directory creation`	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 `automatic`. The following are the settings: • `automatic`—directories are automatically created based on the URI of a document. • `manual-enforced`—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: `http://marklogic.com/file.xml` then the directory with URI `http://marklogic.com/` must exist. Otherwise, an error is raised. This setting provides the same behavior as a file system. • `manual`—directories are not automatically created, but documents can still be created without corresponding directories. For more information about directories, see “Directories” on page 55. For more information about Modules databases, see “Modules Database” on page 121.

Database Setting	Description
`maintain last modified`	Creates and updates the last-modified property each time a document is created or updated. The default is `false`.
`maintain directory last modified`	Creates and updates the last-modified property on a directory each time a directory is created or updated. If set to `true`, 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 `false`. The default is `false`.
`inherit permissions`	When set to `true`, 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.
`inherit collections`	When set to `true`, 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.
`inherit quality`	When set to `true`, documents and directories automatically inherit any quality settings from their parent directory (if quality is not set explicitly when creating the document or directory).

14.1.4.6 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, contact MarkLogic Support.

Database Setting	Description
`in memory limit`	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.
`in memory list size`	The size, in megabytes, of the in-memory list storage.
`in memory tree size`	The size, in megabytes, of the in-memory tree storage. The `in memory tree size` 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 threhold" database configuration setting.
`in memory range index size`	The size, in megabytes, of the in-memory range index storage.
`in memory reverse index size`	The size, in megabytes, of the in-memory reverse index storage.
`in memory triple index size`	The size, in megabytes, of the in-memory triple index storage.
`large size threshold`	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.

Database Setting	Description
`locking`	Specifies how robust transaction locking should be. When set to `strict`, locking enforces mutual exclusion on existing documents and on new documents. When set to `fast`, 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 `strict`. 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 `off`, 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.
`journaling`	Specifies how robust transaction journaling should be. When set to `strict`, the journal protects against MarkLogic Server process failures, host operating system kernel failures, and host hardware failures. When set to `fast`, the journal protects against MarkLogic Server process failures but not against host operating system kernel failures or host hardware failures. When set to `off`, the journal does not protect against MarkLogic Server process failures, host operating system kernel failures, or host hardware failures.
`journal size`	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 `in memory list size` and the `in memory tree size`. 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. 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).
`preallocate journals`	As of 8.0-4, this setting has no effect.

Database Setting	Description
`preload mapped data`	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.
`range index optimize`	Specifies how range indexes are to be optimized. When set to `facet-time`, range indexes are optimized to minimize the amount of CPU time used. When set to `memory-size`, range indexes are optimized to minimize the amount of memory used.

14.1.4.7 Other Settings

The following are the remaining database configuration options.

Database Setting	Description
`position list max size`	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 (`cts:near-query`) for frequently occurring words that have reached this limit (words like *a*, *an*, *the*, 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.
`format compatibility`	Specifies the version compatibility that MarkLogic Server applies to the indexes for this database during request evaluation. Setting this to a value other than `automatic` 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 `automatic` is recommended for most installations.

Database Setting	Description
`index detection`	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 `none` also causes queries to use the current database index settings, even if some settings have not completed reindexing. The default value of `automatic` is recommended for most installations.
`expunge locks`	Specifies if MarkLogic Server will automatically expunge any lock fragments created using `xdmp:lock-acquire` with specified timeouts. If you set this to none, the lock fragments will remain in the database after the locks expire (although they will no longer be locking any documents) until they are explicitly removed with `xdmp:lock-release`. Setting this to `none` is only recommended to speed cluster startup time for extremely large clusters. The default setting of `automatic`, which cleans up the locks as they expire, is recommended for most installations.
`tf normalization`	Specifies whether to use the default term-frequency normalization (`scaled-log`), which scales the term frequency based on the size of the document, or to use the `unscaled-log`, 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.

14.1.4.8 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 “Understanding and Controlling Database Merges” on page 169.

14.1.5 Example of Databases in MarkLogic Server

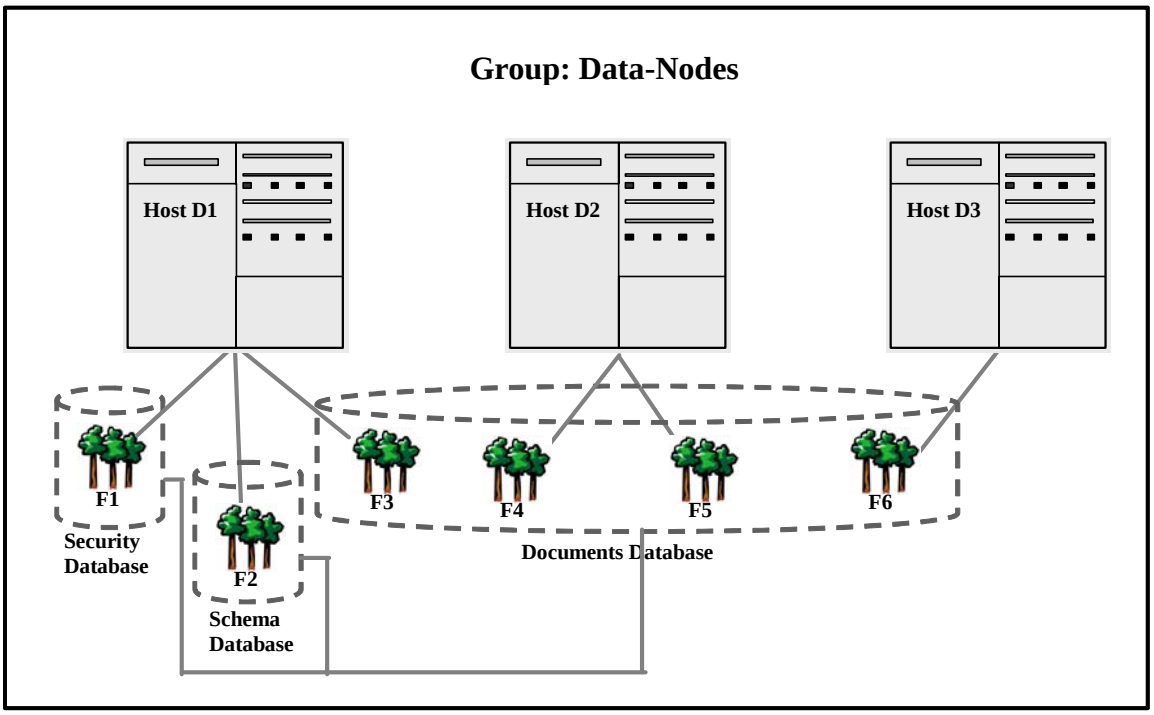
This section provides an example which demonstrates the concept of a database and the relationships between a database, a host and a forest in MarkLogic Server.

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 Database**, **Schema Database** and **Documents Database**. In the diagram below, 3 **Forests,** F1, F2 and F3 are configured on Host D1 and assigned to the **Security Database**, **Schema Database** and **Documents Database** 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**.

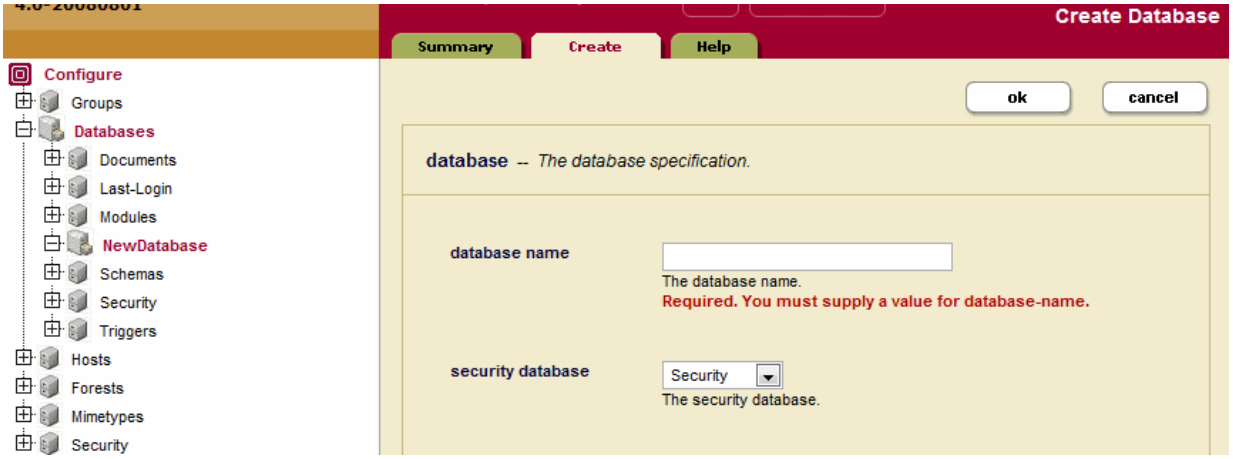


14.2 Creating a New Database

Follow the following steps to create a new database.

1. Click the Databases icon in the left tree menu.

2. Click the Create tab at the top right. The Create Database page displays:



3. Enter the name of the database. This is the name the system will use to refer to this database.

4. Select a security database to be associated with this database. We recommend selecting *Security* as the security database.

5. Select a schema database to be associated 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.

14.3 Attaching and/or Detaching Forests to/from a Database

In order to query content in a forest, it 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.

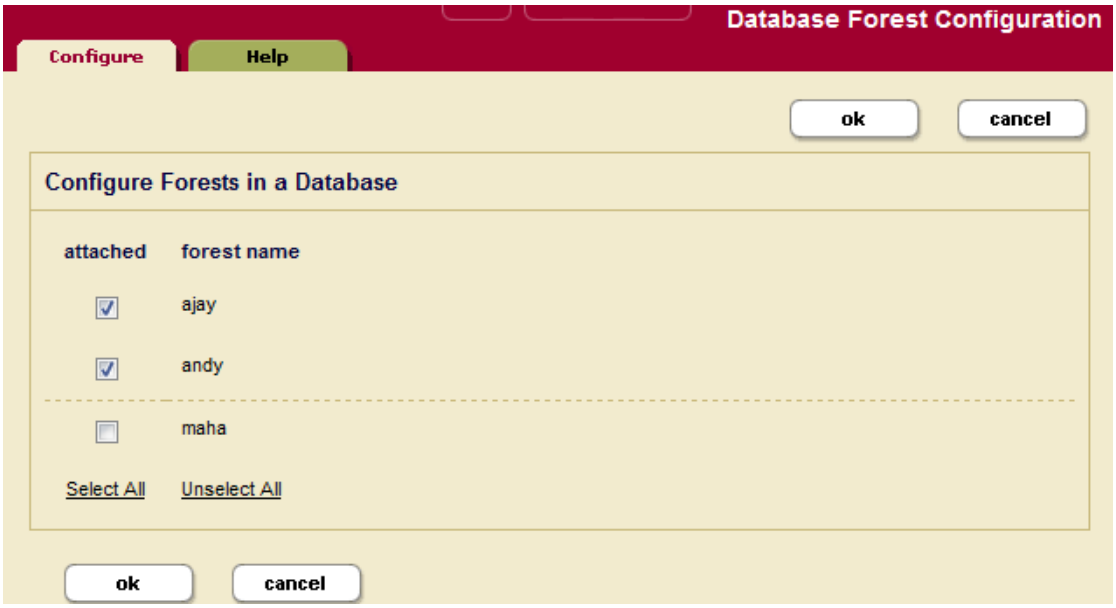
> **Note:** If you attach a new forest to a database that makes use of the journal archiving feature described in "Backing Up Databases with Journal Archiving" on page 241, the forest will not participate in journal archiving until the next time the database is backed up. For details on how to do an immediate backup of a database, see "Backing Up a Database Immediately" on page 245.

You can also attach and detach forests from databases using the Forest Summary page, as described in "Attaching and Detaching Forests Using the Forest Summary Page" on page 282.

Perform the following steps using the Admin Interface to attach or detach one or more forests to a database:

1. Click the database to which you want to attach forests.

2. Click the Forests icon for the database. The Database Forest Configuration Page appears.



3. Check the box corresponding to forest(s) you want to attach to the database. You can also uncheck forests you want to detach from the database.

4. 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.

14.4 Viewing Database Settings

To view the settings for a particular database, perform the following 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.

14.5 Loading Documents into a Database

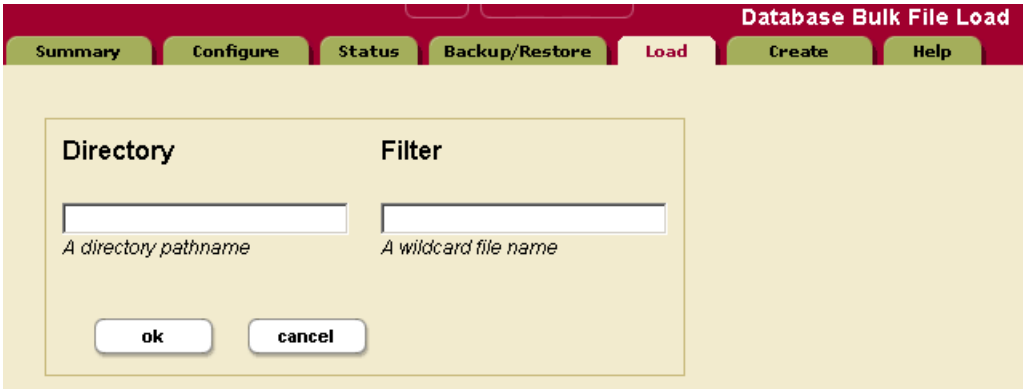
You can use the Admin Interface to load documents into the database. The documents will be loaded with the default permissions and added to the default collections of the user with which you logged into the Admin Interface.

To load a set of documents into a database, perform the following steps:

1. Click the Databases icon on 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 OK to proceed.

7. The load confirmation screen will list all documents in the specified directory matching the specified filter. 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.

14.6 Merging a Database

You can merge all of the forest data in the database using the Admin Interface. As described in “Understanding and Controlling Database Merges” on page 169, 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, complete the following procedure:

1. Click the Databases icon on the left tree menu.

2. Decide which database you want to merge.

3. Click the database name, either on the tree menu or the summary page.

   The Database Configuration page displays.

4. Click the Merge button on the Database Configuration page.

   A confirmation message displays.

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.

14.7 Reindexing a Database

You can reindex all of the document data in the database using the Admin Interface. As described in “Text Indexing” on page 321, 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` 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` is set to true). The Reindex button forces a complete reindex/refragment operation on the database.

To reindex the database, complete the following procedure:

1. Click the Databases icon on the left tree menu.

2. Decide which database you want to reindex.

3. Click the database name, either on the tree menu or the summary page.

   The Database Configuration page displays.

4. Click the Reindex button on the Database Configuration page.

   A confirmation message displays.

5. 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.

14.8 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 in tact.

To clear all data from a database, complete the following procedure:

1. Click the Databases icon on the left tree menu.
2. Decide which database you want to clear.
3. Click the database name, either on the tree menu or the summary page.

   The Database Configuration page displays.
4. Click the Clear button on the Database Configuration page.

   A confirmation message displays.
5. 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.

14.9 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, complete the following procedure:

1. Click the Databases icon on the left tree menu.

2. Decide which database you want to disable.

3. Click the database name, either on the tree menu or the summary page.

   The Database Configuration page displays.

4. Click the Disable button on the Database Configuration page.

   A confirmation message displays.

5. Click either Disable Database to disable only the database, or Disable Database and Forests to disable the database and its forests.

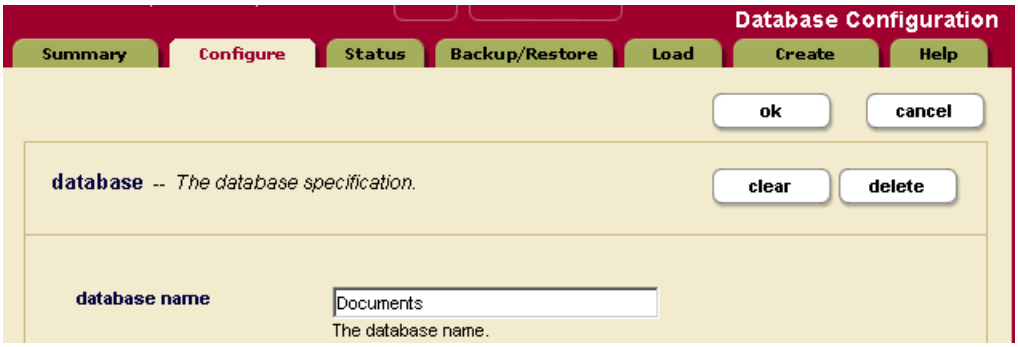
14.10 Deleting a Database

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. Perform the following steps to delete a database:

1. Click the Databases icon on the left tree menu.

2. Locate the database you want to delete, either in the tree menu or in the Database Summary table.

3. Click the name of the database which you want to delete.



4. Click on the Delete button near the top right.

   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.

5. Assuming that there are not any HTTP, WebDAV, or XDBC servers referring to the database, a delete confirmation screen appears. Click OK.

6. If you want to delete the forests used by the database, follow the procedure described in “Deleting a Forest from a Host” on page 289 for each forest.

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

14.11 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 *Understanding and Using Security Guide*.

To check and/or set permissions on a document or directory in a database using the Admin Interface, perform the following steps:

1. Click the Databases icon on the left tree menu.

2. Locate the database for which you want to check or set permissions, either in the tree menu or in the Database Summary table.

3. Click the name of the database where the document to which you want to check or set permissions is stored. The Database Configuration page appears.

4. Click the Permissions link for the selected database in the left tree menu. The Permissions Admin page appears.

5. Enter the URI of the document or directory and click OK.

6. 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.

7. To commit your changes, click OK. To cancel the action, press Cancel.

15.0 Word Query Database Settings

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

- Understanding the Word Query Configuration
- Configuring Customized Word Query Settings

15.1 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 and includes the following parts:

- Overview of Configuration Options
- Understanding Which Elements are Included and Excluded
- Adding a Weight to Boost or Lower the Relevance of an Included Element
- Specifying An Attribute Value for an Included Element
- Understanding the Index Option Configuration

15.1.1 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 can add extra index options for word query. These added index options will not affect other queries (for example, `cts:element-word-query`, `cts:element-attribute-word-query`).
- 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. However, defining word query excludes disables the use of word positions for some queries and can lead to false positives.
- 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.

15.1.2 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 “Fields Database Settings” on page 152.

By default, all XML 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 use not 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 `xdmp.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).

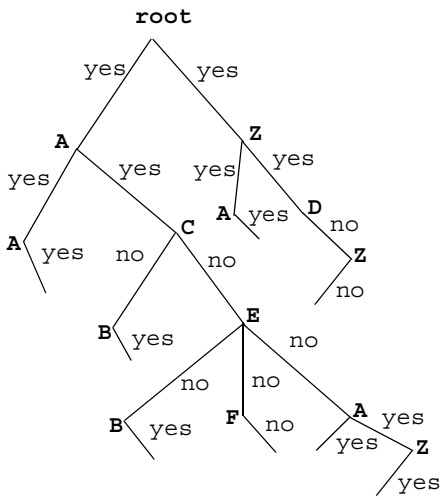
6. During this process, when an element is encountered that is neither included nor excluded, it inherits the included state (*not included* or *included*) from the parent element.

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.

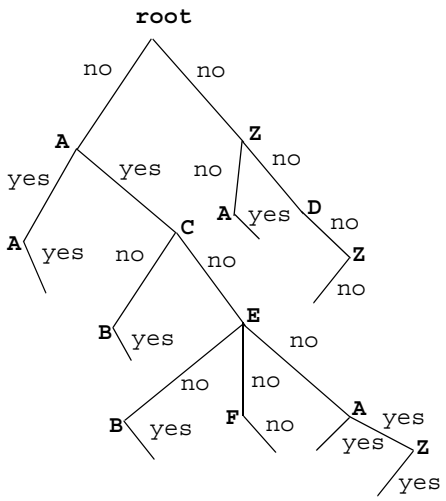
The only way to guarantee an element's text node children will be included (assuming you have any elements included and/or excluded) is to add it to the included list, and the only way to guarantee an element is not included is to add it to the excluded list.

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 rode 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.

Root: Included
Includes: A, B
Excludes: C, D



Root: Excluded
Includes: A, B
Excludes: C, D



The lines indicate text nodes, Yes is included, No is excluded

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.

15.1.3 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 the chapter Composing cts:query Expressions in the *Search Developer's Guide*.

15.1.4 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.

15.1.5 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.

15.2 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 "Understanding the Word Query Configuration" on page 143. 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 "Fields Database Settings" on page 152.

Use the Admin Interface to perform the following steps to add a new field configuration to a database.

1. Access the Admin Interface in a browser.

2. Navigate to and click the database for which you want to modify the word query configuration, either from one of the summary tables or in the left tree menu.

3. Under the database in which you want to create the field, click the Word Query link. The Word Query Configuration page appears.

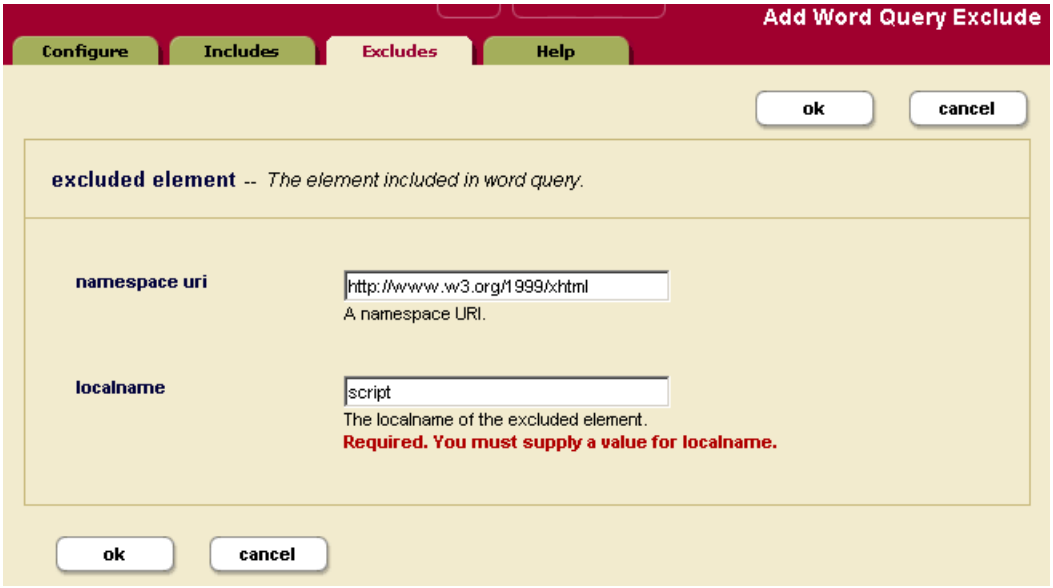
4. If you want the word queries to include any extra index options from the database, check those index settings. Index settings shown in bold indicate the setting is inherited from the database setting. For details, see "Understanding the Index Option Configuration" on page 147.

5. If you want the word queries to include the root element of the document, even if it is not explicitly included, leave the default of `true` for include document root button. Note that if you set this to `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).

6. Click OK to save any changes you made. The configuration page refreshes with after the changes have been made to the MarkLogic Server configuration.

7. If you want to exclude any elements from word queries, click the Excludes tab.

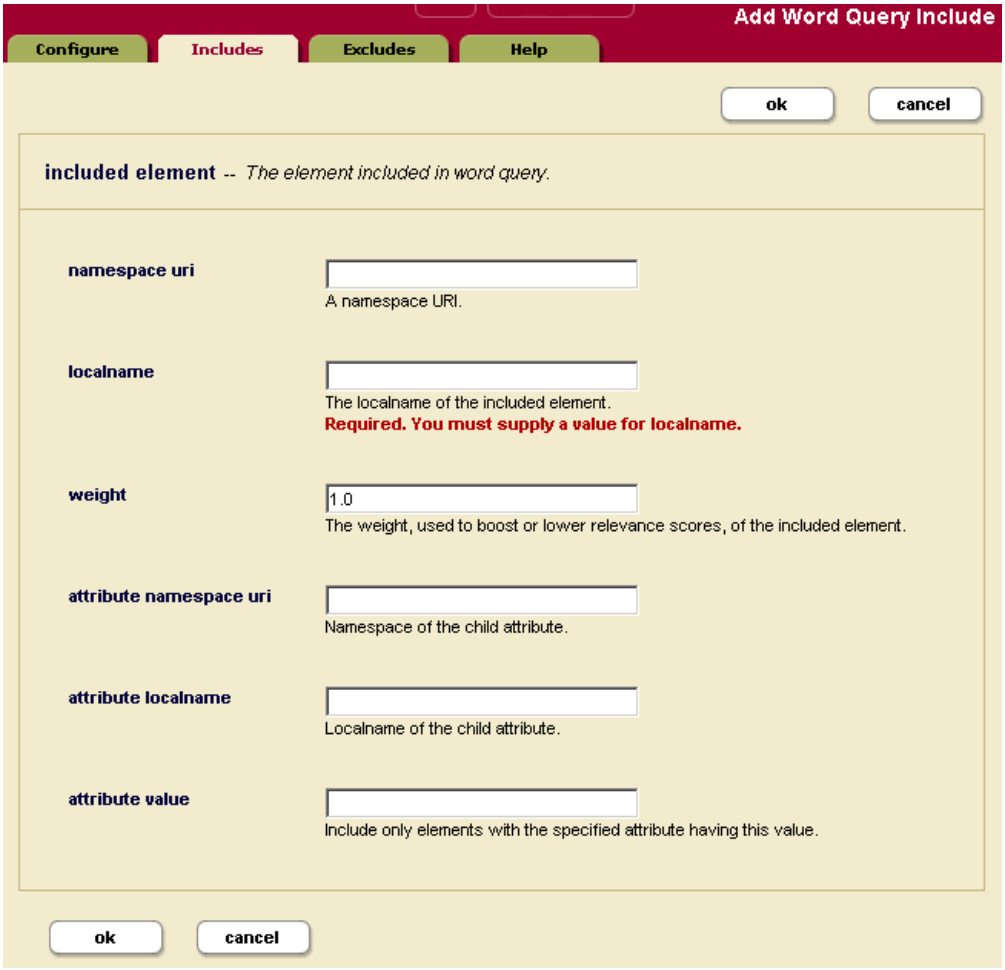
8. Enter the namespace URI (if needed) and the localname for the excluded element.



9. Click OK.

10. Repeat steps 7 through 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 localname for the element to include. If the element is in a namespace, specify the namespace URI for the element to include.

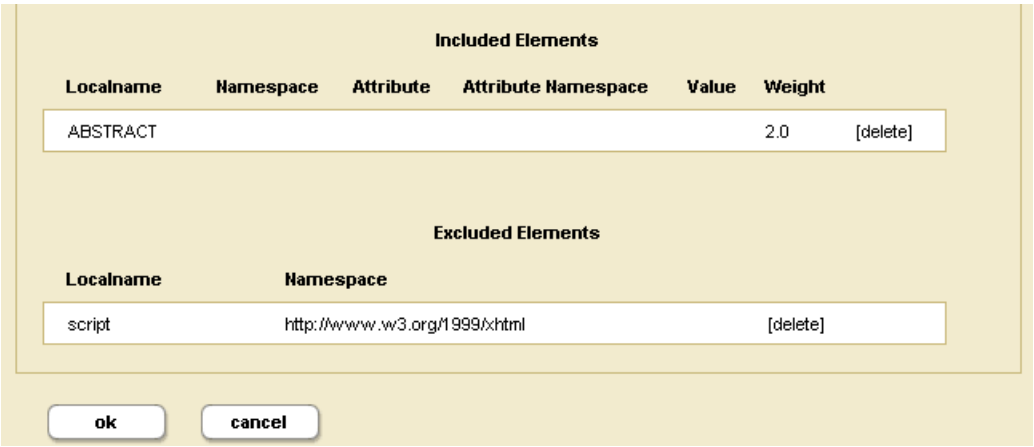
13. [OPTIONAL] If you want to boost or lower the relevance contribution for matches within this element, specify a weight other than the default of 1.0. Weights greater than 1.0 will boost the relevance contribution and weights lower than 1.0 will lower the contribution.

14. [OPTIONAL] If you want to only include elements that have an attribute with a specified value, enter the attribute namespace URI (if needed), the attribute localname, 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.

15. When you have specified everything for this element, click OK.

16. Repeat steps 11 through 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 field configuration page.



16.0 Fields Database Settings

This chapter describes how to 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. The following topics are included in this chapter:

- Overview of Fields
- Understanding Field Configurations
- Field Word Lexicons and Field Value Lexicons
- Configuring Fields

This chapter describes how to use the Admin Interface to create and configure fields. 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*.

16.1 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*.

16.2 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 fields, and you can add extra indexing options for some elements. This section describes the options available in the configuration and includes the following parts:

- Overview of Field Configuration Options
- Understanding What is Included and Excluded in a Field
- Adding a Weight to Boost or Lower the Relevance of an Included Element or Property
- Specifying An Attribute Value for an Included or Excluded Element
- Understanding the Index Option Configuration

16.2.1 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 from each field.
- 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, 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).

16.2.2 Understanding What is Included and Excluded in a Field

You can include and/or exclude elements from a 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.

There are two types of fields, each of which determines where to start including and excluding elements in the document tree:

- Root Fields
- Path Fields

Each is described in the following sections.

16.2.2.1 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.

16.2.2.2 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 "Defining Namespace Prefixes Used in Path Range Indexes and Fields" on page 356.

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.

16.2.2.3 How Field Settings Determine What is 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.

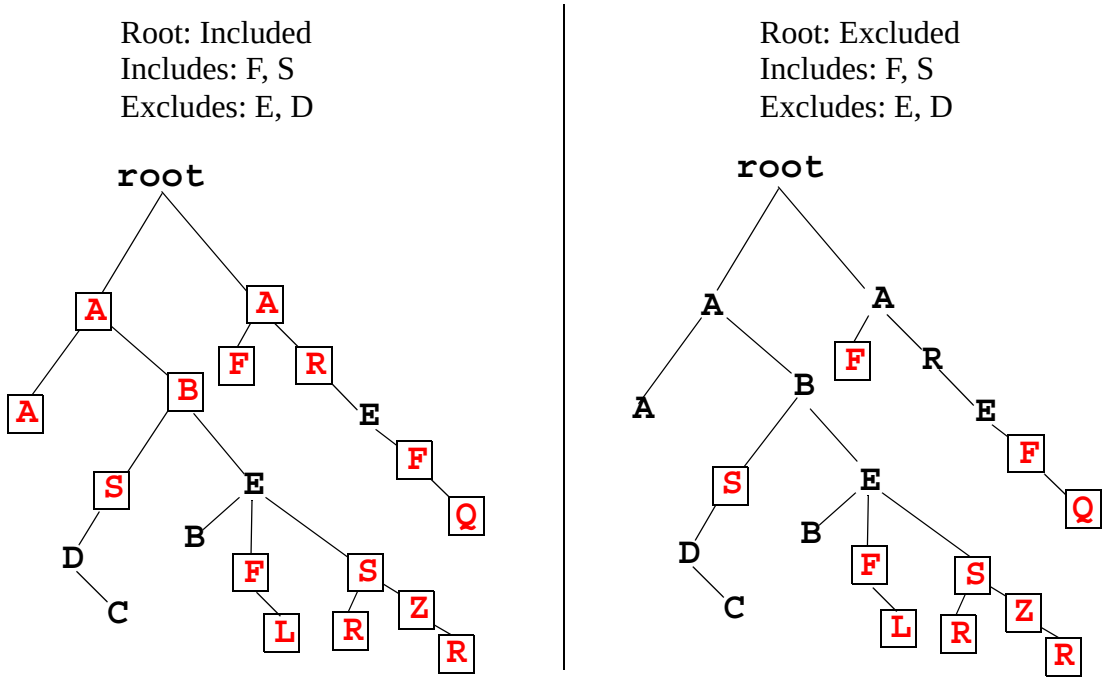
5. If the parent element was not included, MarkLogic Server keeps walking down the tree, not including the text node children, until it encounters an explicitly included element.

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.

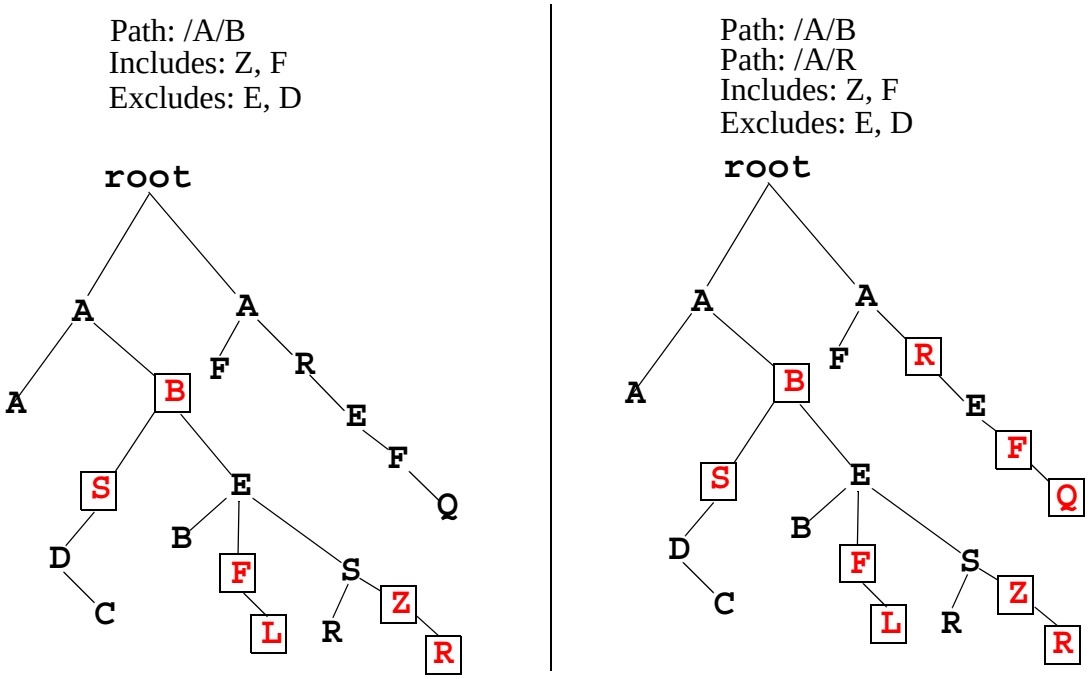
The following figure shows what is included for two possible root field configurations, one with the root node included and one with the root node excluded. Note that the includes and excludes are the same. The lines below the element names represent the text nodes, and the boxed red letters indicates 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.



The lines indicate text nodes, boxed red is included, black is excluded

Notice that the `A`, `B`, and `R` nodes, which is not explicitly included or excluded, sometimes is included and sometimes is not included, depending on the include state of its 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.



The lines indicate text nodes, boxed red is included, black is excluded

16.2.3 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 the chapter 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>
```

16.2.4 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.

16.2.5 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.

16.3 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*.

16.4 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 "Understanding Field Configurations" on page 153. This section includes the following procedures:

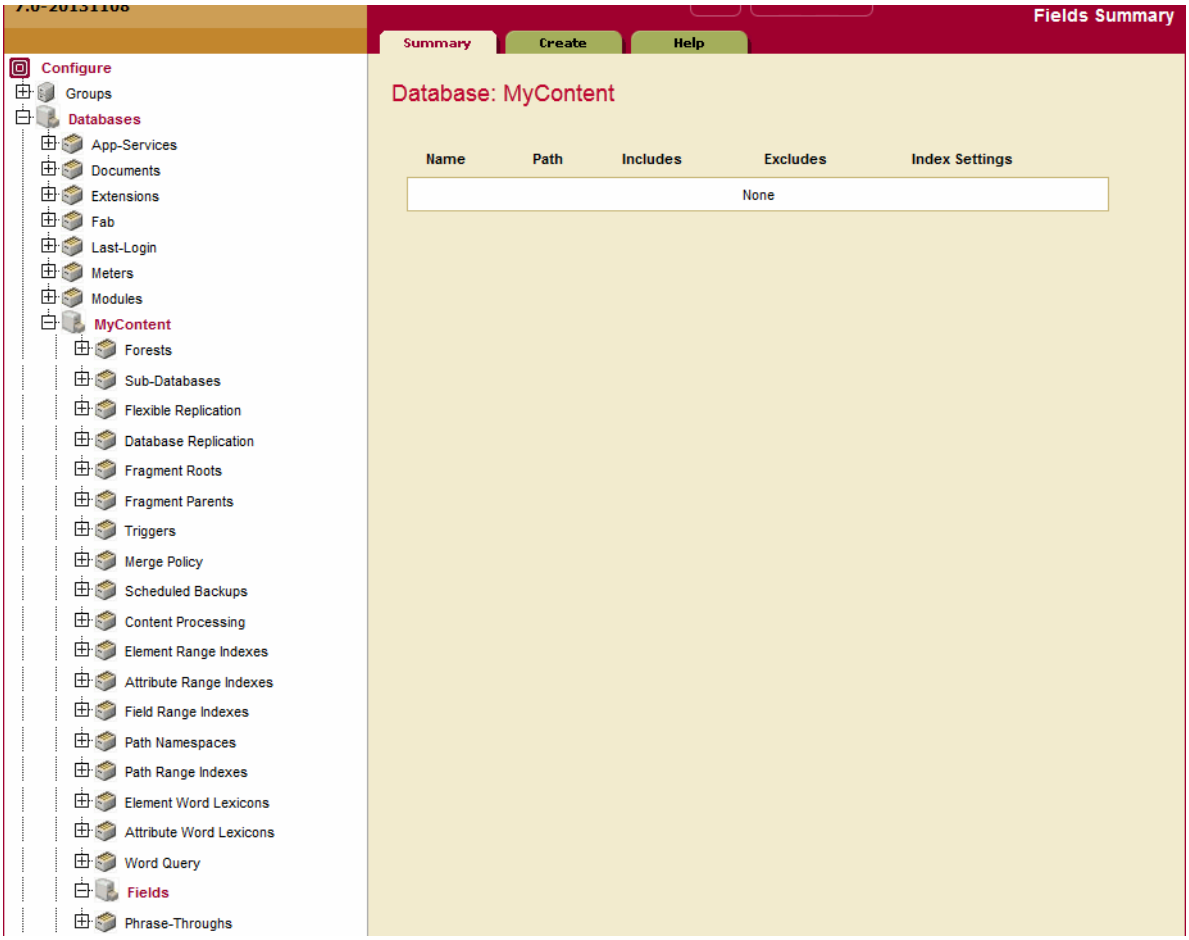
- Configuring a New Field
- Modifying an Existing Field
- Creating a Range Index on a Field

16.4.1 Configuring a New Field

Use the Admin Interface to perform the following steps to add a new field configuration to a database.

1. Navigate to and click the database for which you want to create a field, either from one of the summary tables or in the left tree menu.

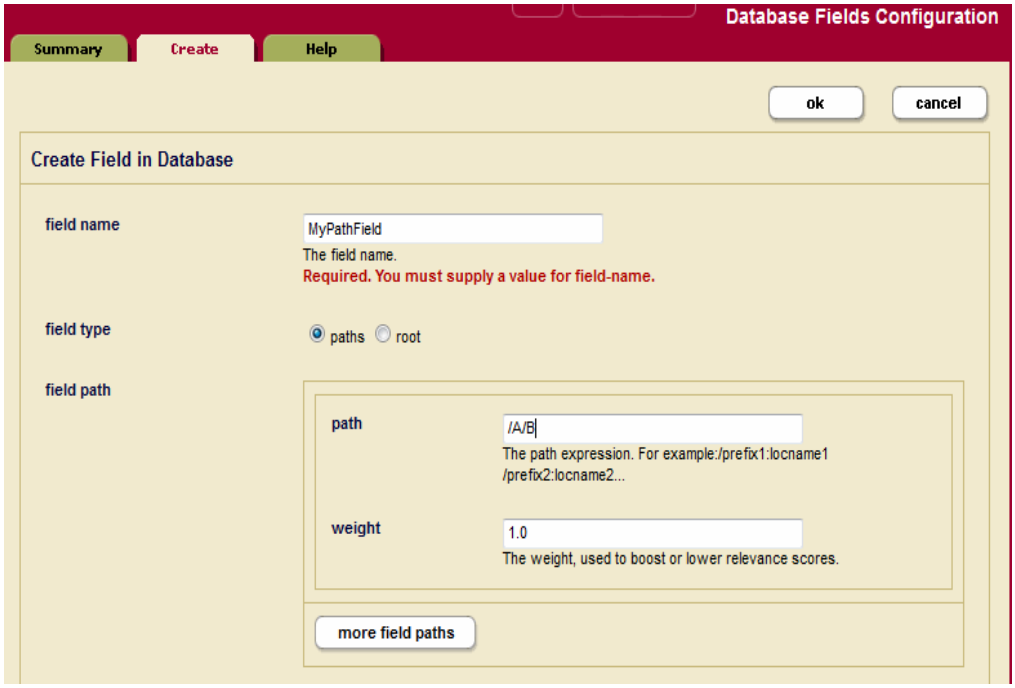
2. Under the database in which you want to create the field, click the Fields link. The Field Summary page appears.



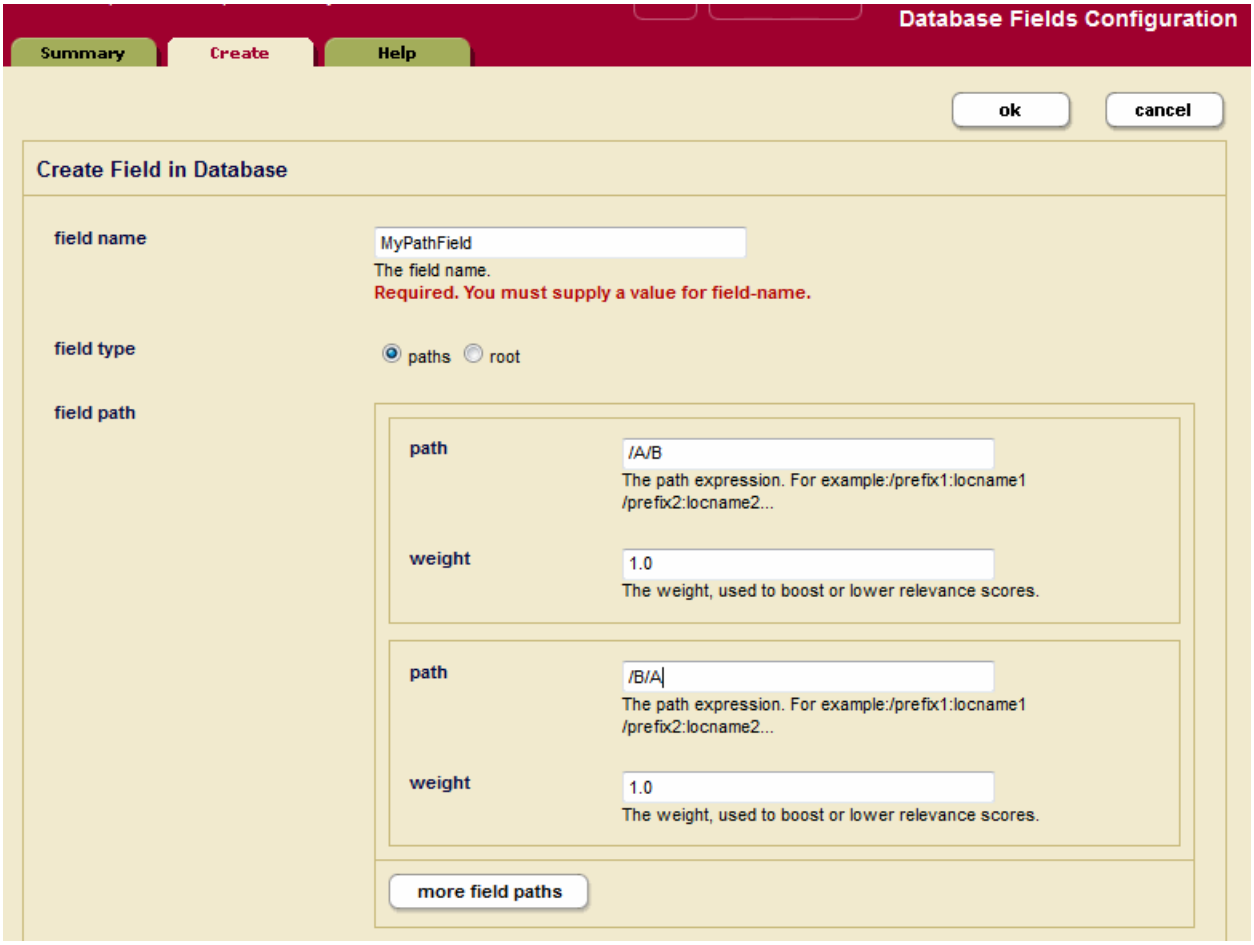
3. Click the Create tab. The Create Field in Database page appears.

4. Enter a name for the field.

5. By default, the field type is path. 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 Items.

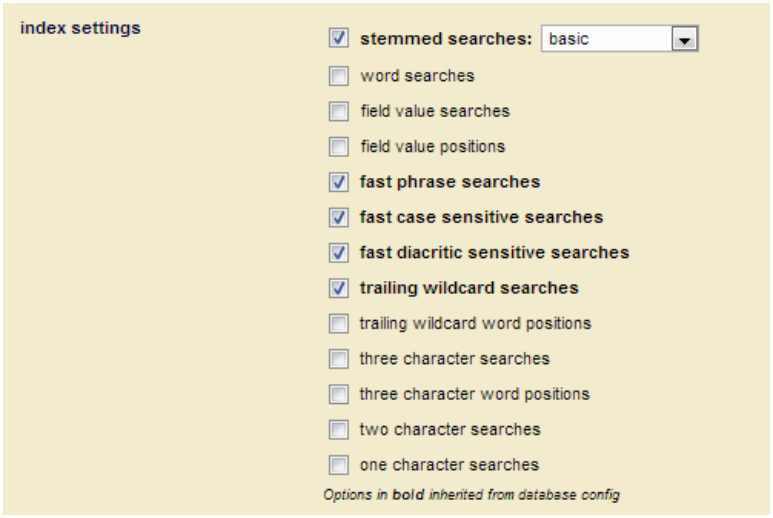


6. Enter as many paths as you need.



7. [OPTIONAL] Create any Field Range Indexes or Tokenizer overrides. You can also go back and add these later.

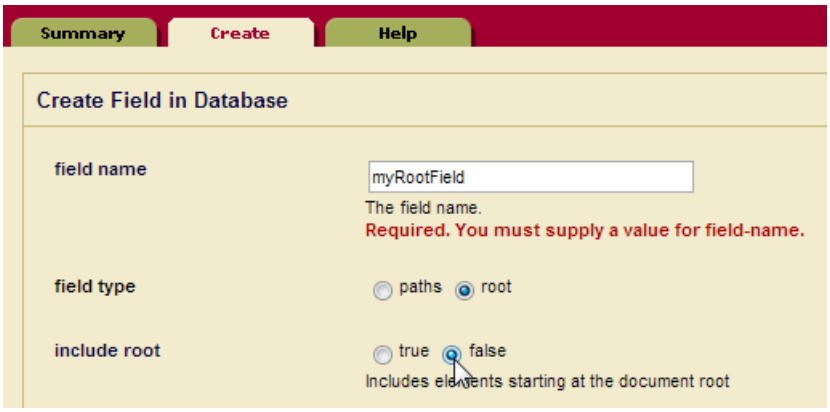
8. If you want the field to include any extra index options from the database, or if you want to remove some index options from the field, check or uncheck those index settings. Index settings shown in bold indicate the setting is inherited from the database setting. You can uncheck an inherited index setting to not inherit the setting from the database-level configuration. For details, see "Understanding the Index Option Configuration" on page 159.



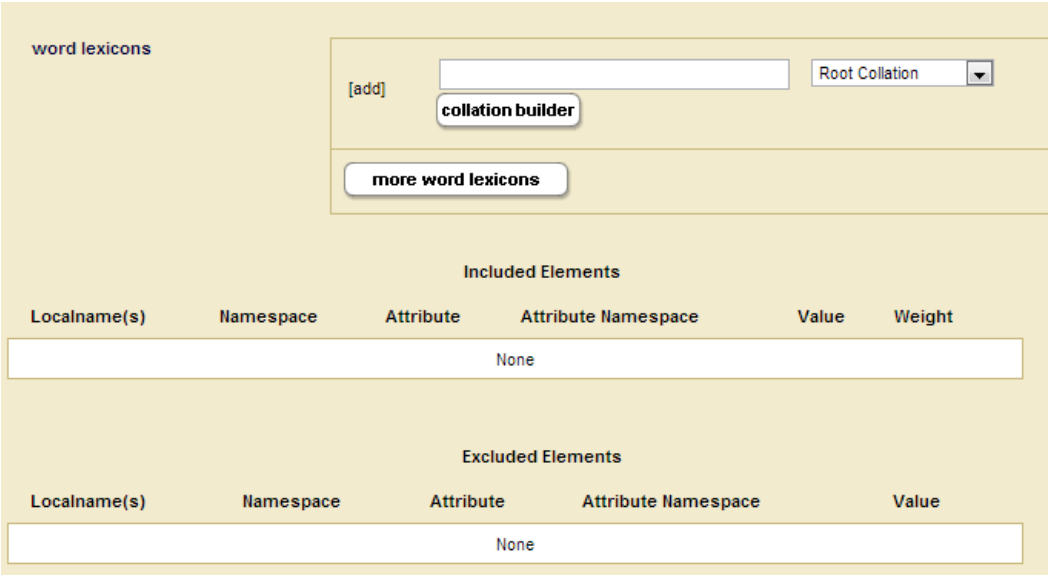
9. Alternately, if creating a root field, set the field type to root. Note that in most cases, a path field will give you everything you need, and you are not likely to need to create a root field.



10. If you want the root field to include the root element of the document, even if it is not explicitly included, click the `true` button for 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.

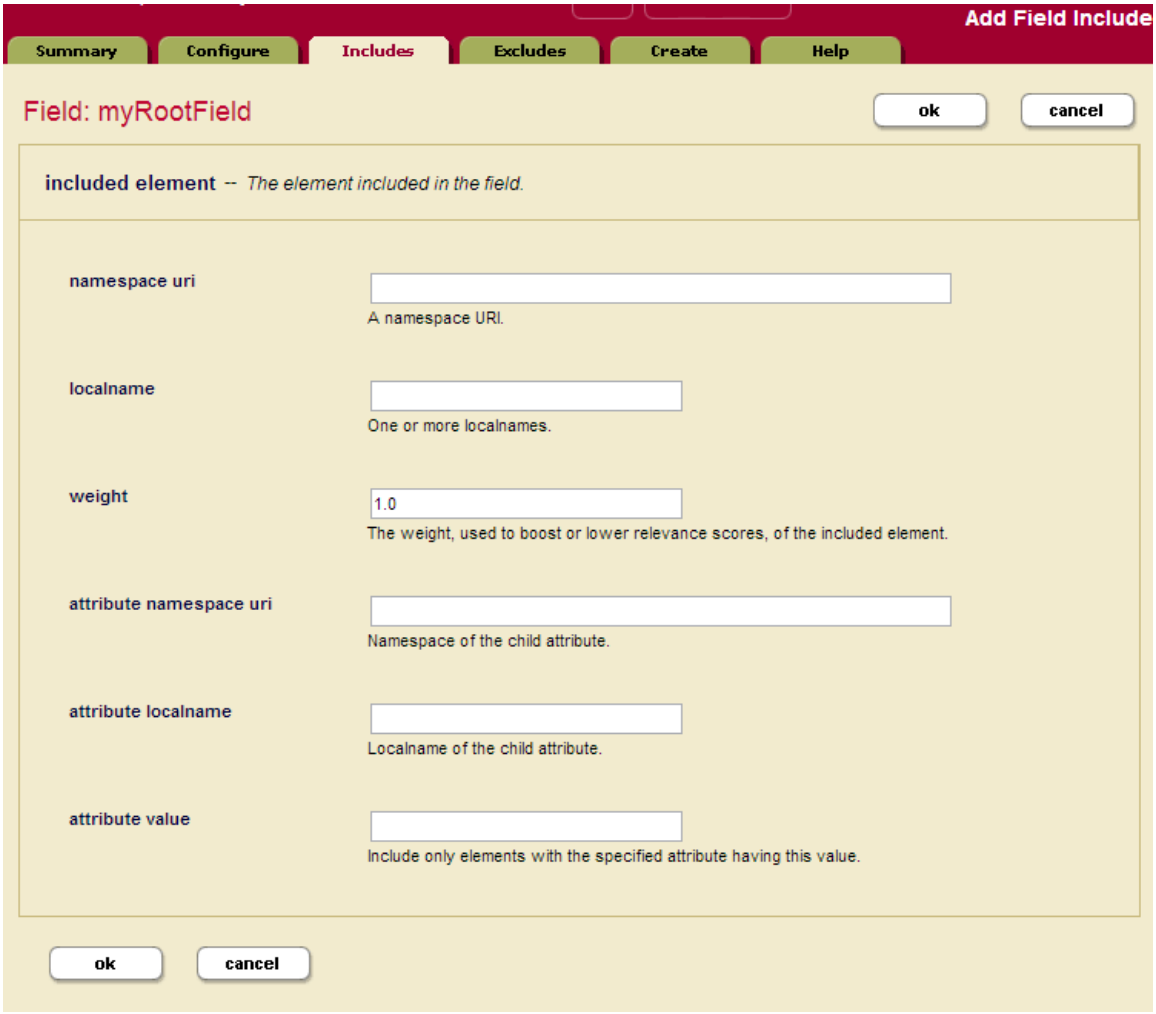


11. Click OK. The configuration page with the field appears, adding the following parts to the bottom of the configuration page:



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 the Language Support in MarkLogic Server chapter 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 localname 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.

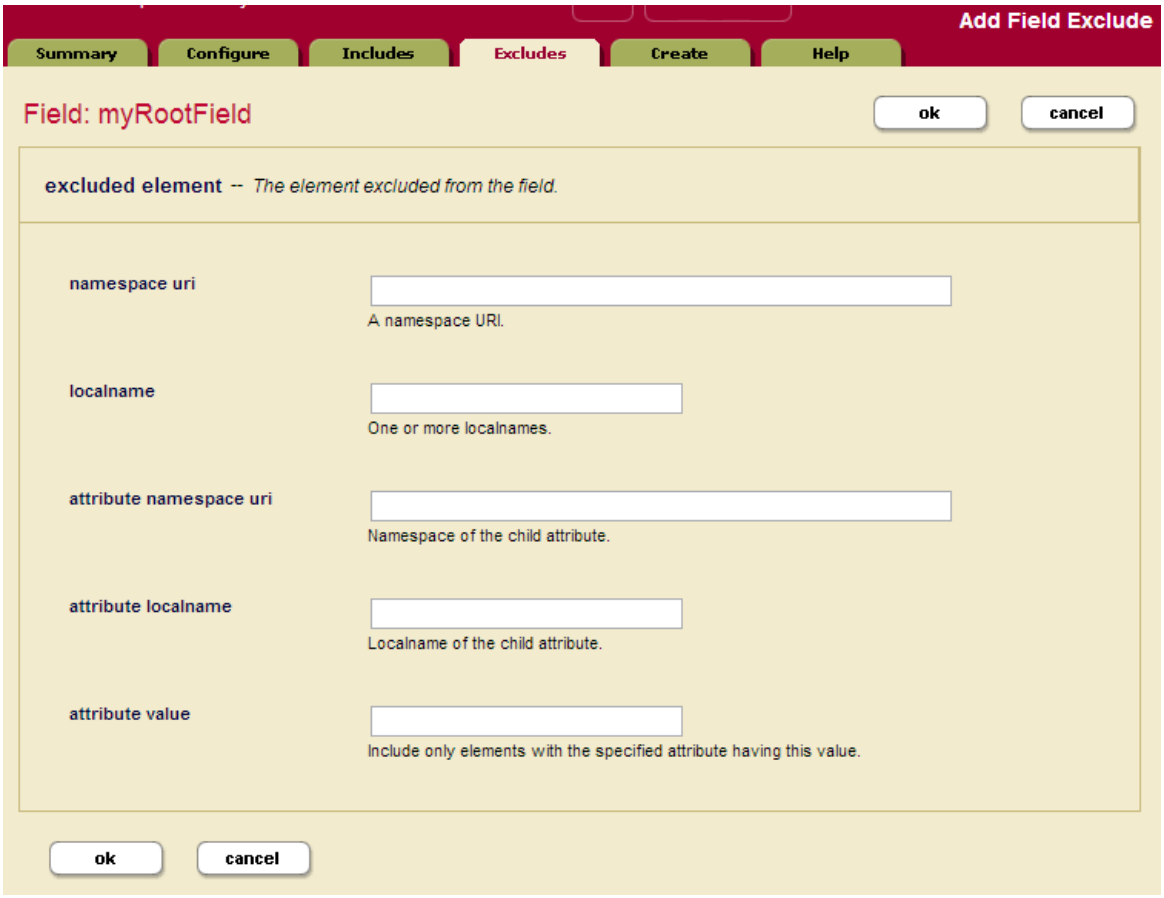
16. [OPTIONAL] If you want to only include elements that have an attribute with a specified value, enter the attribute namespace URI (if needed), the attribute localname, and a value for the attribute. Then only elements containing attributes with the specified value will be included. You must specify the exact value; no wildcard characters are used.

17. When you have specified everything for this element, click OK.

18. Repeat steps 13 through 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 localname for the excluded element.

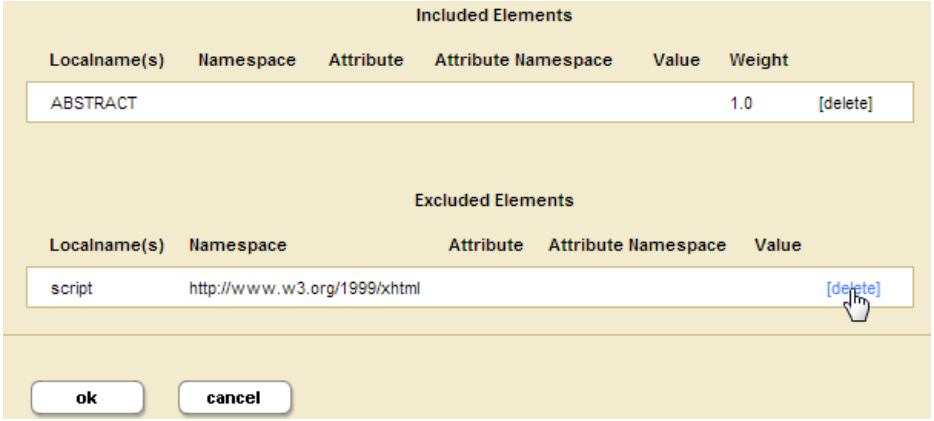


21. [OPTIONAL] If you want to only exclude elements that have an attribute with a specified value, enter the attribute namespace URI (if needed), the attribute localname, and a value for the attribute. Then only elements containing attributes with the specified value will be excluded. You must specify the exact value; no wildcard characters are used.

22. Click OK.

23. Repeat steps 19 through 22 for each element you want to exclude.

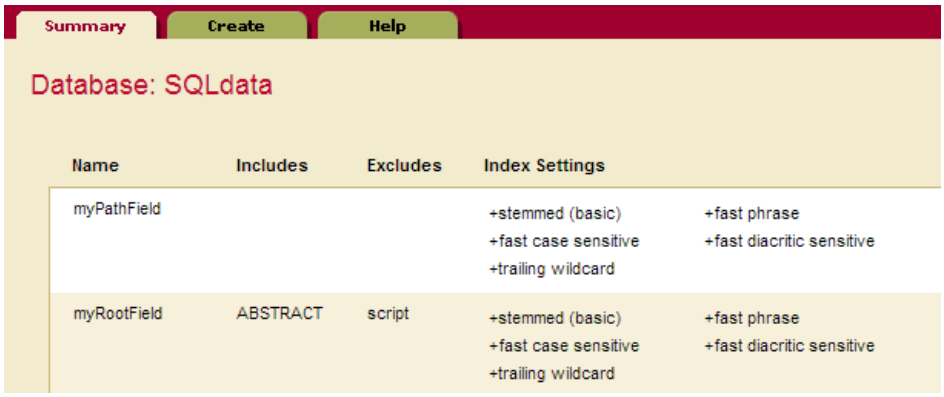
24. You can delete any included or excluded fields from the tables at the bottom of the field configuration page.



16.4.2 Modifying an Existing Field

Perform the following steps to modify an existing field:

1. To modify an existing field, click on the Fields link in the left tree menu. The Fields Summary page appears.



2. Click on the name of the field you want to edit. The Field Configuration page appears.

3. If you want to change any of the settings, make any desired modifications and click OK.

4. The remainder of the procedure is the same as the previous procedure for creating a field, starting with step 12 to create a field word lexicon, and continuing on to add/delete included and excluded elements.

16.4.3 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 “Understanding Range Indexes” on page 341 apply to field range index.

Perform the following steps to create a range index on a field:

1. Navigate to and click the database for which you want to create a field range index, either from one of the summary tables or in the left tree menu.

2. Click Field Range Index in the left tree menu.

3. Click the Add tab. The Add Field Range Indexes page appears.

4. Select the type for the range index.

5. Enter the name of an existing field.

6. Optionally, specify if you want the index to store position data.

7. For Invalid Values, 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.

8. 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.

17.0 Understanding and Controlling Database Merges

This chapter describes database merges and how you can control them. It includes the following sections:

- Overview of Merges: Merges are Good
- Setting Merge Policy
- Blackout Periods for Merges
- Merges and Point-In-Time Queries
- Setting a Negative Merge Timestamp to Preserve Fragments For a Rolling Window of Time
- Monitoring a Merge
- Explicit Merge Commands
- Configuring Merge Policy Rules

17.1 Overview of Merges: Merges are Good

This section provides an overview of merges, and includes the following parts:

- Dynamic and Self-Tuning
- What Happens During a Merge
- Dangers of Disabling Merges
- Merges Will Change Scores

17.1.1 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 “Setting Merge Policy” on page 171.

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.

17.1.2 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.

17.1.3 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 stands 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, contact MarkLogic Customer Support.

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, contact MarkLogic Customer Support, and be sure to monitor the system and make sure the number of stands per forest does not grow too high. For details on setting merge controls, see “Description of Merge Parameters” on page 173 and “Configuring Merge Policy Rules” on page 180.

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 “Making a Forest Delete-Only” on page 279.

17.1.4 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.

17.2 Setting Merge Policy

This section describes the tools you can use to control merges, and has the following parts:

- Overview of the Merge Policy Controls
- Description of Merge Parameters

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 “Making a Forest Delete-Only” on page 279.

17.2.1 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” on page 179):
  - `xdmp:merge()`
  - The merge button in Admin Interface.
- The Admin Interface has controls for cancelling a merge (see “Cancelling a Merge” on page 180).

For more information on how set up your system to better control merges, see “Configuring Merge Policy Rules” on page 180.

17.2.2 Description of Merge Parameters

The following table describes the settings available on the Databases > *db_name* >Merge Policy page of the Admin Interface. These parameters determine when automatic merges occur on a database, as well as other administrative functions.

Database Setting	Description
`merge priority`	Specifies the CPU scheduler priority at which merges should run. The settings are: • `normal` specifies the same CPU scheduler priority as for requests. • `lower` specifies a lower CPU scheduler priority than for requests. Merges always run with normal priority on forests with more than 16 stands.
`merge max size`	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 32GB (32768MB), 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.
`merge min size`	The minimum number of fragments that a stand can contain. Two or more stands with fewer than this number of fragments are automatically merged.
`merge min ratio`	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 “If You Want to Reduce the Number of ‘Large’ Merges” on page 181.

Database Setting	Description
`merge timestamp`	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 `current timestamp` 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*.
`retain until backup`	Specify whether the deleted fragments are retained since the last full or incremental backup. When enabled, `retain until backup` supersedes `merge timestamp`. Deleted fragments are not merged until backups are finished, regardless of the `merge timestamp` setting. Enabling `retain until backup` is same to setting the `merge timestamp` to the timestamp of the last backup. For more information, see "Incremental Backup with Journal Archiving" on page 244.
`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.

17.3 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 and includes the following parts:

- Understanding Merge Blackouts
- Configuring Merge Blackout Periods
- Deleting Merge Blackout Periods

17.3.1 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 “Cancelling a Merge” on page 180. 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.

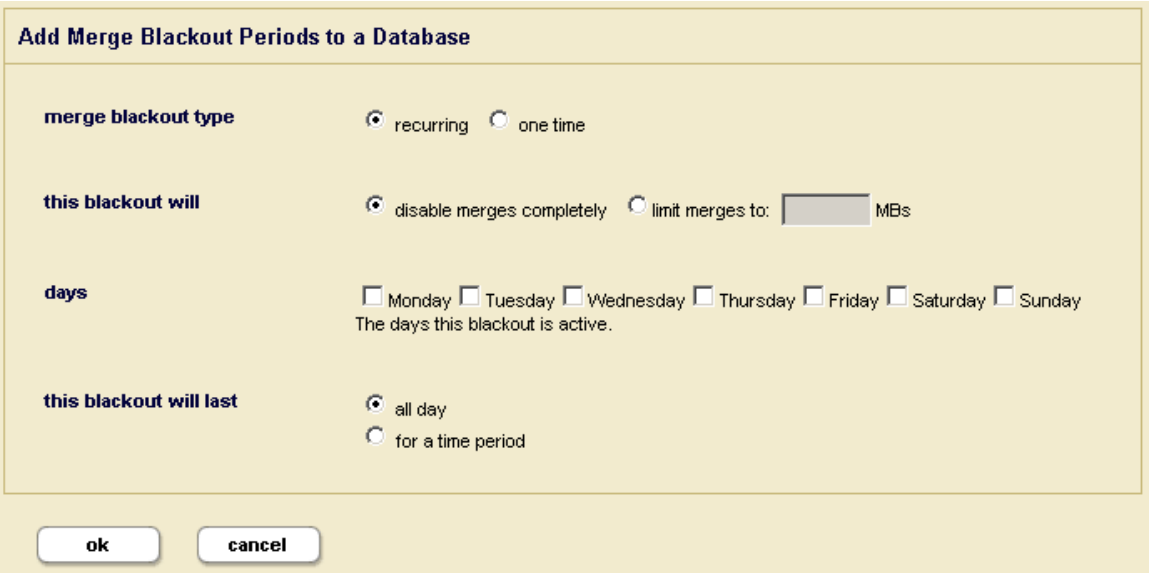
17.3.2 Configuring Merge Blackout Periods

Perform the following to configure merge blackout periods:

1. In the Admin Interface tree menu, click the Databases > *db_name* link, where *db_name* is the name of the database in which you want to specify merge blackout periods.

2. Click the Merge Policy menu item under your database. The Merge Policy Configuration page appears.

3. Click the Create tab. The Add Merge Blackout page appears.



4. 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.

5. Click OK to create the blackout period.

The new blackout period will take effect immediately.

17.3.3 Deleting Merge Blackout Periods

Perform the following to delete a merge blackout period:

1. In the Admin Interface tree menu, click the Databases > *db_name* link, where *db_name* is the name of the database in which you want to delete a merge blackout period.

2. Click the Merge Policy menu item under your database. The Merge Policy Configuration page appears.

3. In the area corresponding to the blackout period you want to delete, click the Delete button.

4. Click OK on the confirmation page to delete the blackout period.

The blackout period is deleted immediately.

17.4 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 chapter on "Point-In-Time Queries" in the *Application Developer's Guide*.

17.5 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	`merge timestamp` Value
5 minutes	`10000000 * 60 * 5`	`-3000000000`
1 hour	`10000000 * 60 * 60`	`-36000000000`
24 hours	`10000000 * 60 * 60 * 24`	`-864000000000`

17.6 Monitoring a Merge

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

- Messages in the ErrorLog.txt File
- Database Status Page

17.6.1 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 where 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.

17.6.2 Database Status Page

You can access the Database Status page by clicking the Databases > *db_name* link in the tree menu, then clicking the Status tab in the Admin Interface. 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 "Cancelling a Merge" on page 180).

17.7 Explicit Merge Commands

This section describes how to manually perform the following operations:

- Manually Initiating a Merge
- Cancelling a Merge

17.7.1 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).

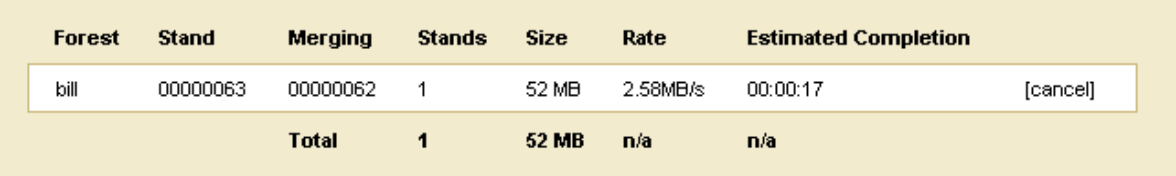
> **Note:** If you have updates occurring on the system while a merge is in progress, the new fragments will not be merged during the active merge operation; they will be merged during a subsequent merge.

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 the last hour; 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 XQuery and XSLT Function Reference*.

17.7.2 Cancelling a Merge

You can cancel a merge in the Database Status page of the Admin Interface (Databases > *db_name* > Status tab). 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).



Forest	Stand	Merging	Stands	Size	Rate	Estimated Completion	
bill	00000063	00000062	1	52 MB	2.58MB/s	00:00:17	[cancel]
		Total	**1**	**52 MB**	**n/a**	**n/a**	

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 the Databases menu item in the Admin Interface.
2. Click the name of the database, either from the tree menu or from the summary page.
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.

17.8 Configuring Merge Policy Rules

By changing some of the merge policy parameters, you can effectively control certain aspects of your merges. The descriptions in “Description of Merge Parameters” on page 173 describes what each parameter does. This section describes some scenarios with suggestions for how to tune the merge control parameters to satisfy the conditions. It includes the following parts:

- Determine the Baseline for Your Merges
- If You Want to Reduce the Number of ‘Large’ Merges
- Other Solutions

17.8.1 Determine 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 about your merges:

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

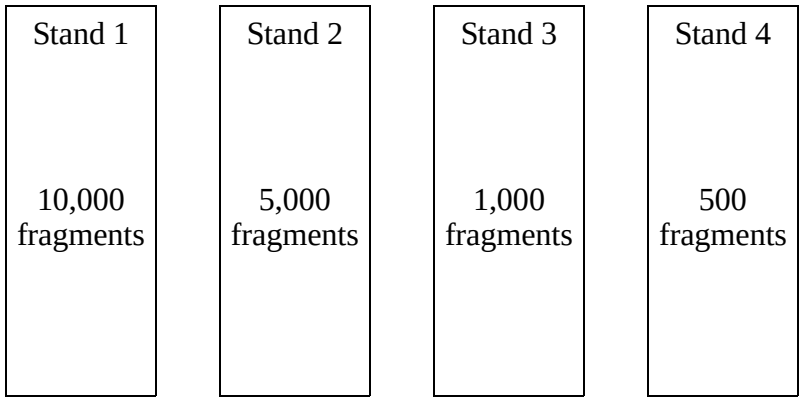
If it turns out that your merges are never taking more than a few minutes to complete, then there is probably no need to change any of your settings.

17.8.2 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:

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

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:

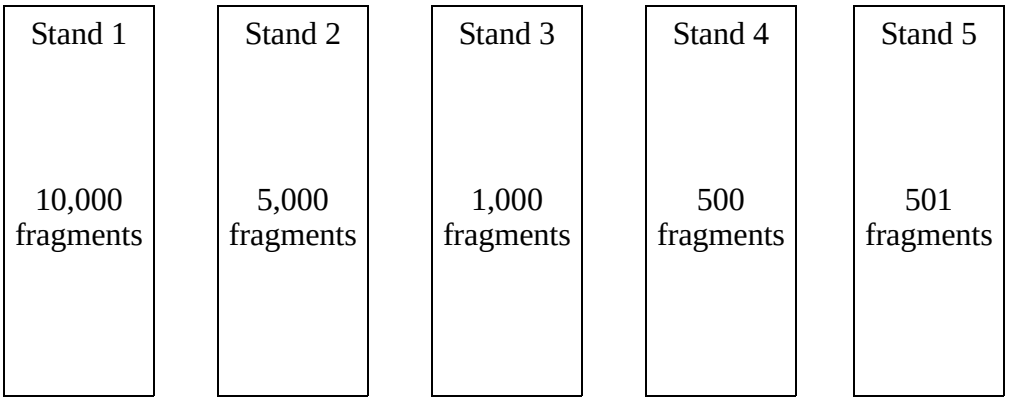
```
5000/(1000 + 500) = 5000/1500 = 3.33
```

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.

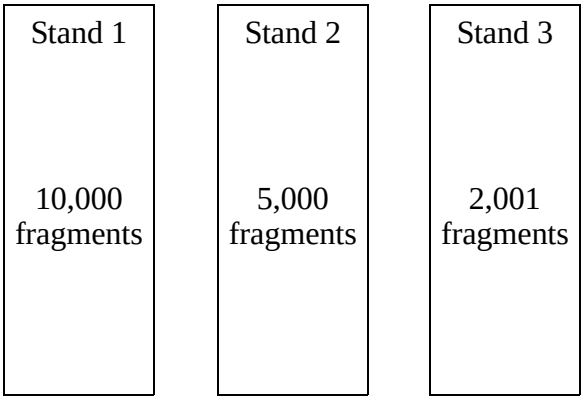
Now, consider that a load is happening during this time and a stand that has 501 fragments is saved into the forest. The result is 5 stands as follows:



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 will now look as follows:



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.

17.8.3 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 256GB, query performance tends to start to decrease while merge times tend to start to increase. If your forests are larger than 256GB, consider breaking them into multiple forests.

18.0 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 chapter describes the database rebalancing mechanism that enables MarkLogic Server to evenly distribute content among the database forests.

This chapter includes the following topics:

- Overview of the Database Rebalancer
- Rebalancer Trigger Events
- Rebalancer Document Assignment Policies
- How the Rebalancer Moves Documents
- Configuring the Rebalancer on a Database
- Configuring the Rebalancer on a Forest
- Retiring a Forest from the Database
- Checking the Rebalancer Status
- How the Rebalancer Interacts with other Database and Forest Settings
- Rebalancer Settings after Upgrading from an Earlier Release

18.1 Overview of the Database Rebalancer

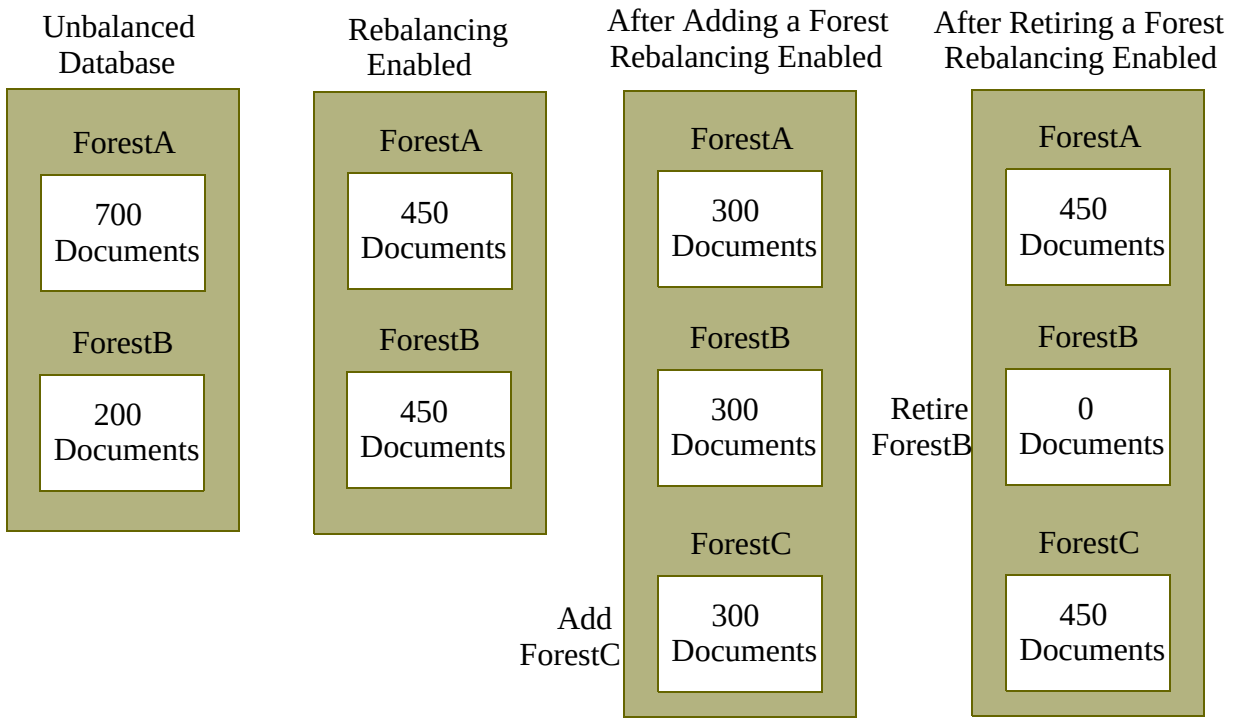
A database rebalancer consists of two parts: an *assignment policy* for data insert and rebalancing and a *rebalancer* for data movement. The rebalancer can be configured with one of several assignment policies, which define what is considered “balanced” for a database. You choose the appropriate policy 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.



18.2 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.

18.3 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.

The four commonly used assignment policies are as follows:

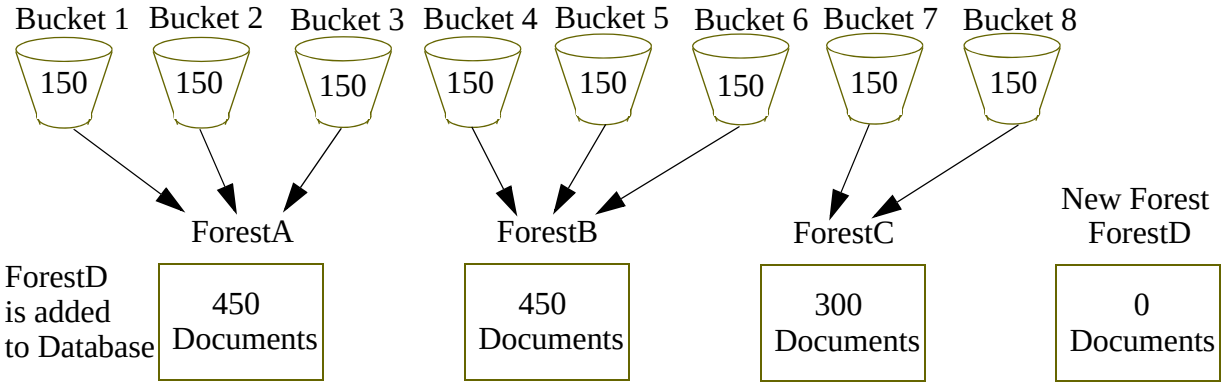
- Bucket Assignment Policy
- Statistical Assignment Policy
- Range Assignment Policy
- Legacy Assignment Policy

18.3.1 Bucket Assignment Policy

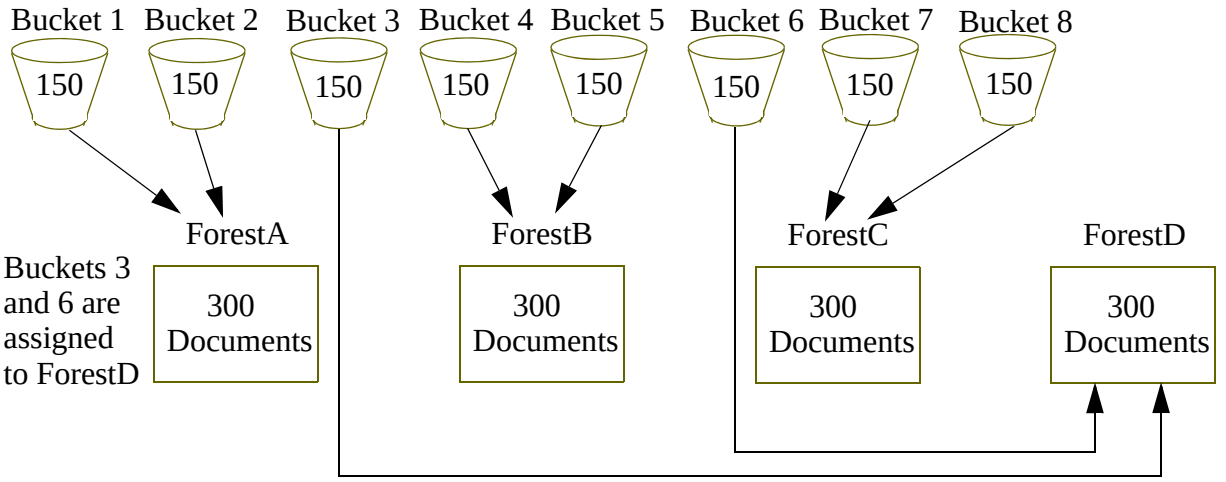
The bucket policy uses the URI of a document to decide which forest the document should be assigned to. The URI is first "mapped" to a bucket then the bucket is "mapped" to a forest. The mapping from a bucket to a forest is kept in memory for fast access. The number of buckets is always 16K, regardless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



Rebalancer Process Begins

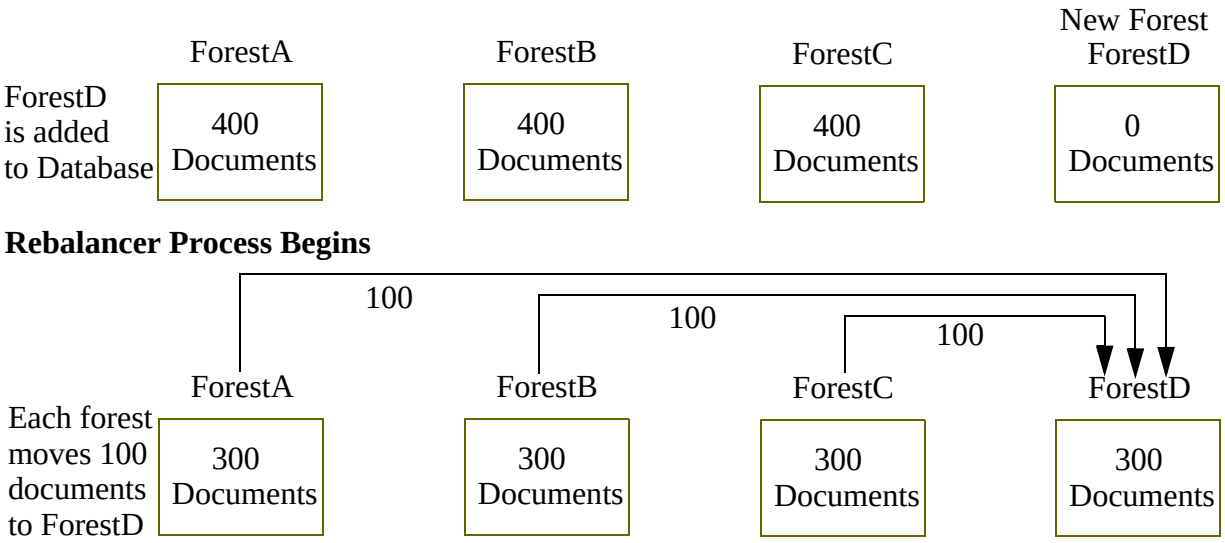


The bucket assignment policy is, in most situations, the most efficient document assignment policy because it is deterministic and it moves the least amount of data of the deterministic assignment policies.

18.3.2 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 number of 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.

Note: 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.

18.3.3 Range Assignment Policy

The range policy is designed for use with the Tiered Storage feature described in "Tiered Storage" on page 203. 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 "Statistical Assignment Policy" on page 189.

> **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.

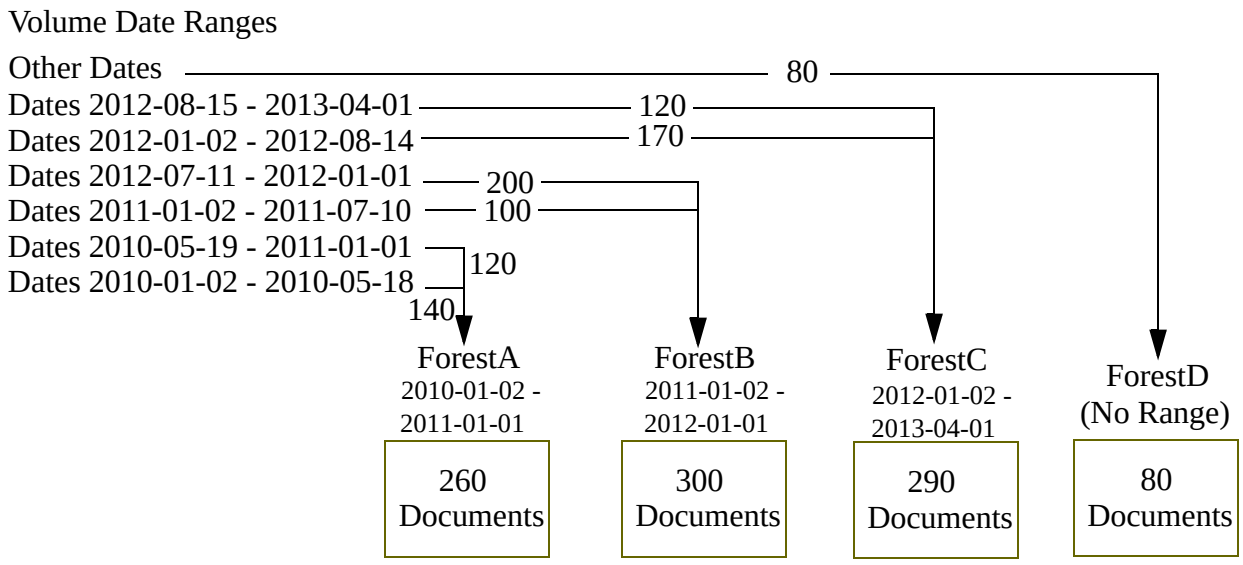
```
xdmp:document-add-properties(
     "example.xml",
     (<creation-date>2010-01-02</creation-date>))
```

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.

Contains SAP BusinessObjects Text Analysis XI from SAP AG. Copyright © 1996-2011. All rights reserved.

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.



Contains SAP BusinessObjects Text Analysis XI from SAP AG. Copyright © 1996-2011. All rights reserved.

18.3.4 Legacy Assignment Policy

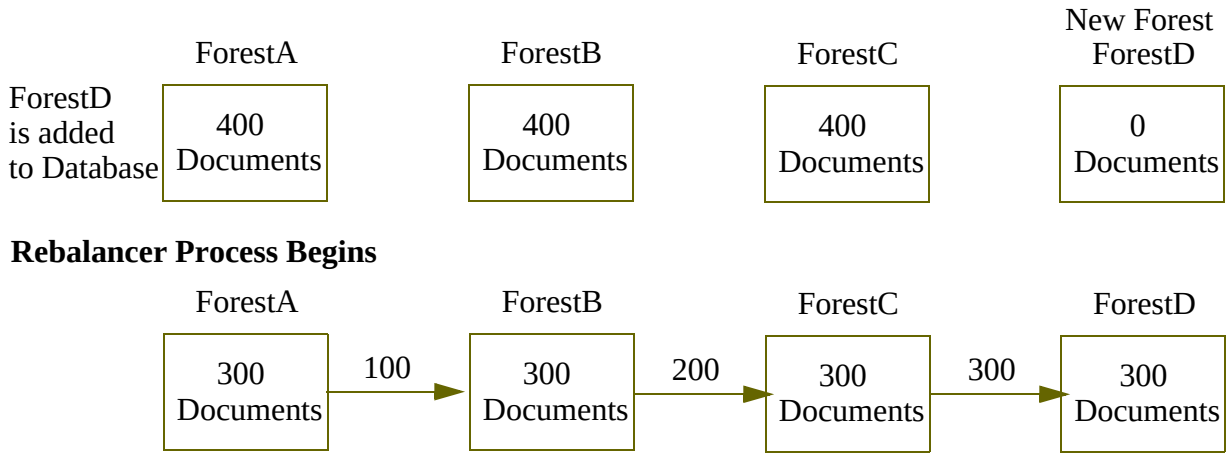
After upgrading to MarkLogic 7.0, existing databases will be configured with the rebalancer disabled and the legacy assignment policy. This is to preserve the expected behavior when new documents are loaded into the database.

> **Note:** Under most circumstances you would not use the legacy policy when the database rebalancer is enabled.

The 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.

18.3.5 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
Statistical	Least	No	No
Range	Less	Yes (Partition key based)	No
Legacy	Most	Yes (URI based)	Yes

18.4 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.

The following sections describe how documents are moved when forests are reconfigured for the database:

- How Data is Moved when a Forest is Attached to the Database
- How Data is Moved when a Forest is Retired from the Database

18.4.1 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.

18.4.2 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.

18.5 Configuring the Rebalancer on a Database

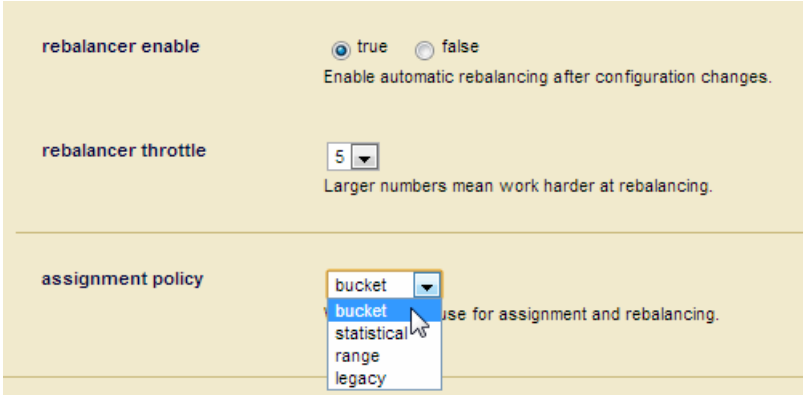
You can configure and monitor the rebalancing process through the Admin Interface or the Admin APIs.

To configure the rebalancer on a database, complete the following procedure:

1. Click the Databases icon on the left tree menu.
2. Decide which database for which you want to configure the rebalancer.
3. Click the database name, either on the tree menu or the summary page.

   The Database Configuration page displays.
4. Scroll down the Database Configuration page to the assignment policy and set the Rebalancer Enable to `true`.

5. From the assignment policy pull-down menu, select the assignment policy. For details on the available rebalancer assignment policies, see “Rebalancer Document Assignment Policies” on page 187.



6. From the rebalancer throttle pull-down menu, select the rebalancer throttle setting. For details on the rebalancer throttle, see “How the Rebalancer Moves Documents” on page 193.

7. Click OK.

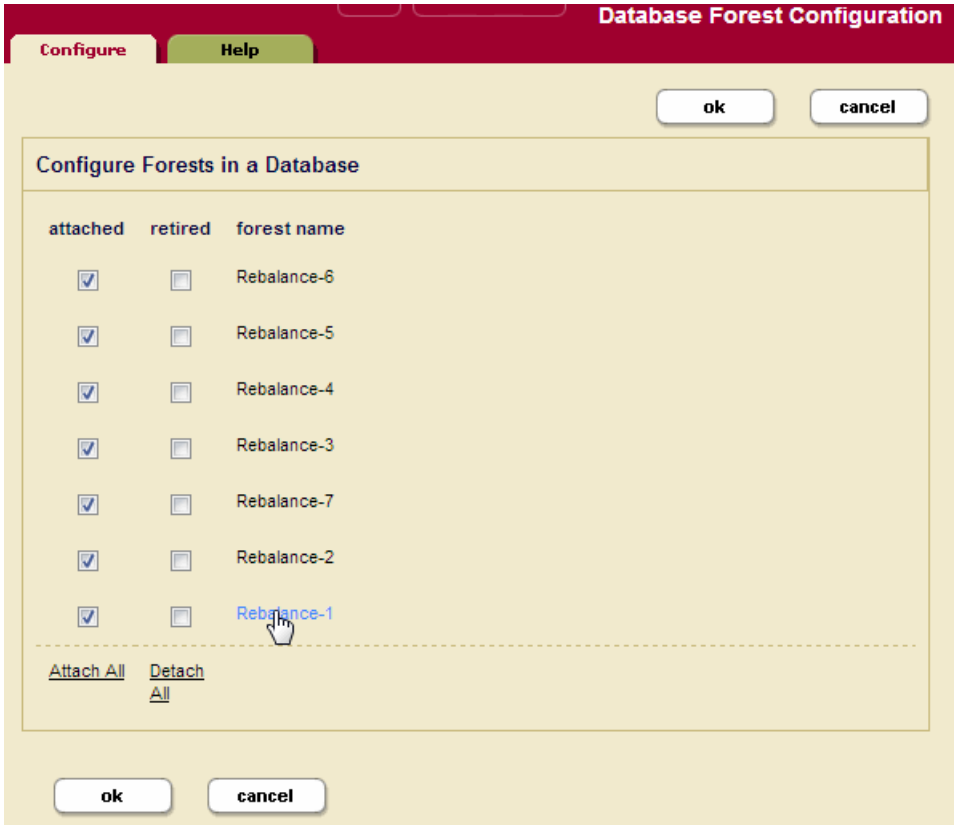
18.6 Configuring the Rebalancer on a Forest

In addition to enabling and disabling on the database level, as described in “Configuring the Rebalancer on a Database” on page 194, 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.

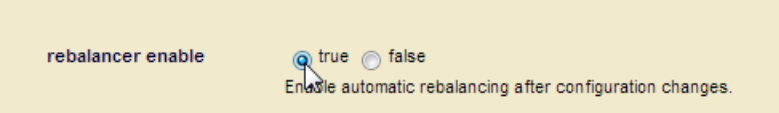
> **Note:** The rebalancer is enabled on each new forest by default.

To configure the rebalancer on a forest, complete the following procedure:

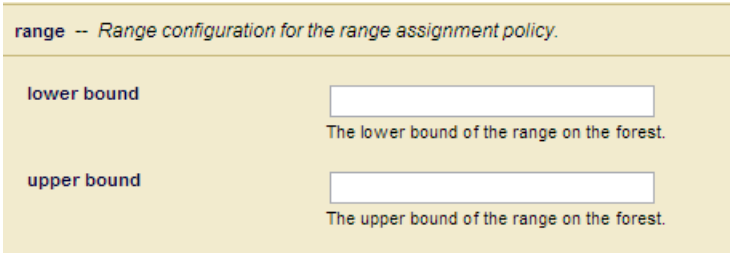
1. Click the Databases icon on the left-tree menu.
2. Select the database for which you want to configure the forest.
3. Click the database name, either on the tree menu or the summary page.
4. In the left-tree menu under the database name, select Forests.
5. In the Database Forest Configuration page, select the forest for which you want to enable or disable the rebalancer.



6. In the Forest Configuration page, scroll down to Rebalancer Enable and set to `true` to enable the rebalancer or `false` disable the rebalancer.



7. 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.



8. Click OK.

18.7 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 "How Data is Moved when a Forest is Retired from the Database" on page 194. 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, complete the following procedure:

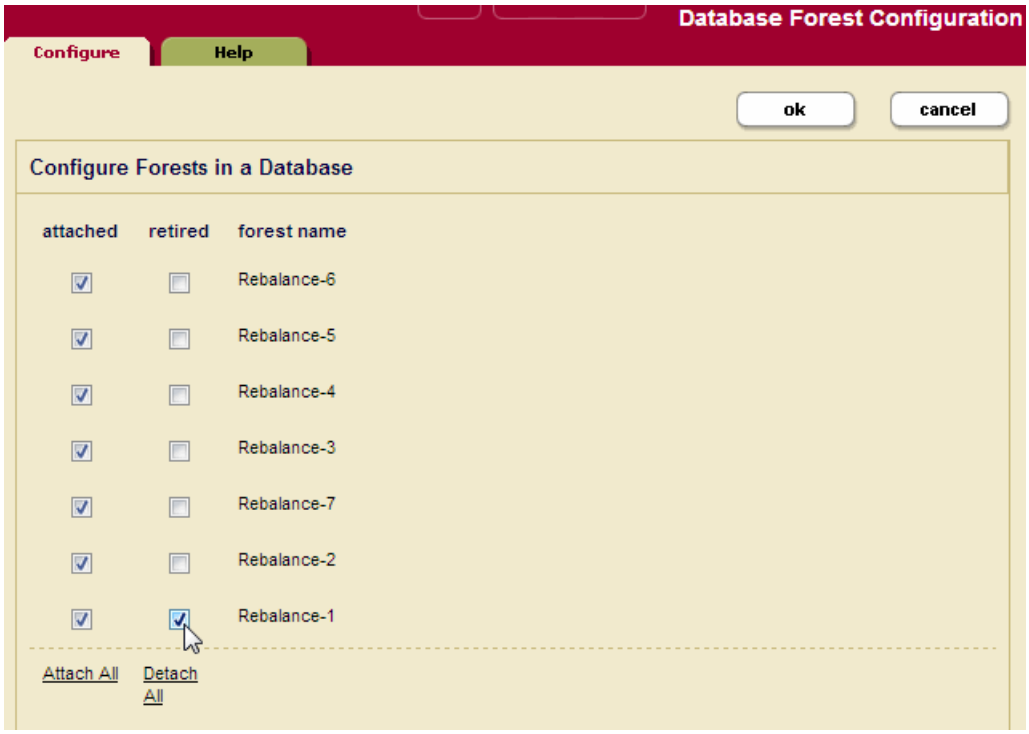
1. Click the Databases icon on the left-tree menu.

2. Decide which database for which you want to retire a forest.

3. Click the database name, either on the tree menu or the summary page.

4. In the left-tree menu under the database name, select Forests.

5. In the Database Forest Configuration page, 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 forest Attached box checked.



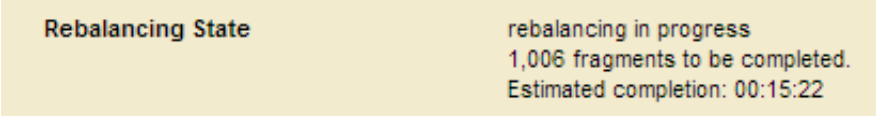
6. Click OK. The documents in the retired forest will be evenly redistributed to the other forests in the database.

7. 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 “Attaching and/or Detaching Forests to/from a Database” on page 135.

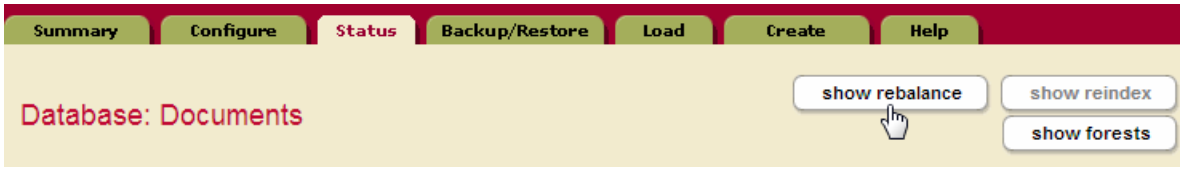
18.8 Checking the Rebalancer Status

When the rebalancer is enabled on the database, you can check the state of the rebalancer, along with an estimated completion time, on the Database Status page.

For example, if the database is rebalancing documents, you will see status similar to the following:



When the rebalancer is disabled on the database, you can click on the Show Rebalance button at the top of the Database Status page to view the number of fragments that are pending rebalancing.



This will display a table like the following toward the middle of the Database Status page:

Destination	Rebalancing Fragments Remaining
1 -- 10	0
11 -- 20	0
21 -- 30	0
31 -- 40	0
41 -- 100	0
101 -- 200	0
201 -- 300	0
301 -- 400	16
DEFAULT	1,227
401 -- 500	99

18.9 How the Rebalancer Interacts with other Database and Forest Settings

This section describes how the database rebalacer interacts with other database and forest settings. The topics are:

- Database Replication
- Restoring a Database from a Backup
- Tiered Storage
- Fast Locking
- Delete-only and Read-only Forests

18.9.1 Database Replication

If you have configured a database for database replication and that database is enabled for rebalancing with the bucket or legacy 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.

18.9.2 Restoring a Database from a Backup

If you have a database enabled for database rebalancing with the bucket or legacy 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 Bucket or Legacy 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.

18.9.3 Tiered Storage

The range assignment policy described in “Range Assignment Policy” on page 190 is designed to support tiered storage. For details on tiered storage, see “Tiered Storage” on page 203.

18.9.4 Fast Locking

Fast locking works with both the legacy policy and the 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.

18.9.5 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.
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.
Range	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.

18.10 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.

19.0 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 chapter 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:** You must be have a license for Tiered Storage in order to use this feature in production.

This chapter contains the following topics:

- Terms Used in this Chapter
- Overview of Tiered Storage
- Partitions, Partition Keys, and Partition Ranges
- Configuring a Database to Participate in Tiered Storage
- Defining a Partition Key
- Creating Partitions
- Overview of the Tiered Storage REST API
- Common Forest and Partition Operations
- Sub-databases and Super-databases
- Partitions with Forest-Level Failover

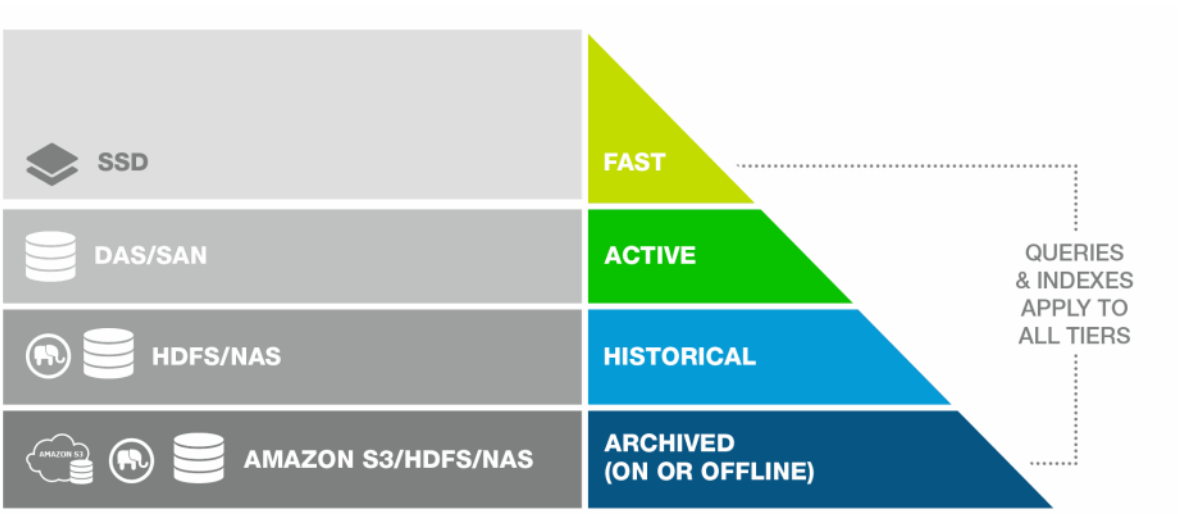
> **Note:** If you are storing your MarkLogic data in an Amazon Elastic Compute Cloud (EC2) S3 bucket, Tiered Storage operations are not supported for forests residing in S3 buckets in the us-east-1 region.

19.1 Terms Used in this Chapter

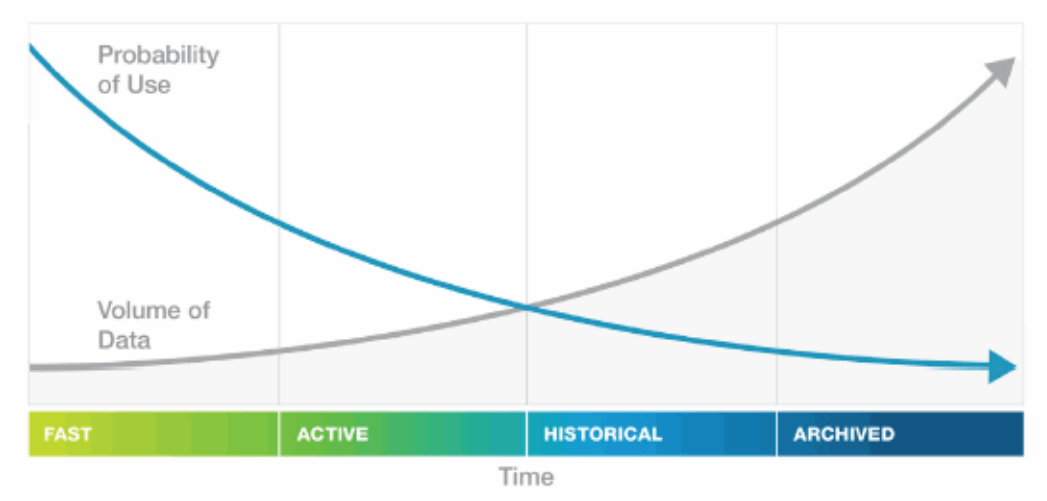
- A *Partition* is a set of forests sharing the same name prefix and same partition range definition. Typically forests in a partition share the same type of storage and configuration such as updates allowed, availability, and enabled status. Partitions are based on forest naming conventions. A forest's 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 *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 partitions in the database. The partition key is a database-level setting.
- A *Partition Range* defines a range of values for a partition. Documents with a partition key value that fall within the range specified for a partition are stored in that partition.
- A *Default Partition* is a partition with no defined range. Documents that have no partition key or a partition key value that does not fall into any of the partition ranges 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.

19.2 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.

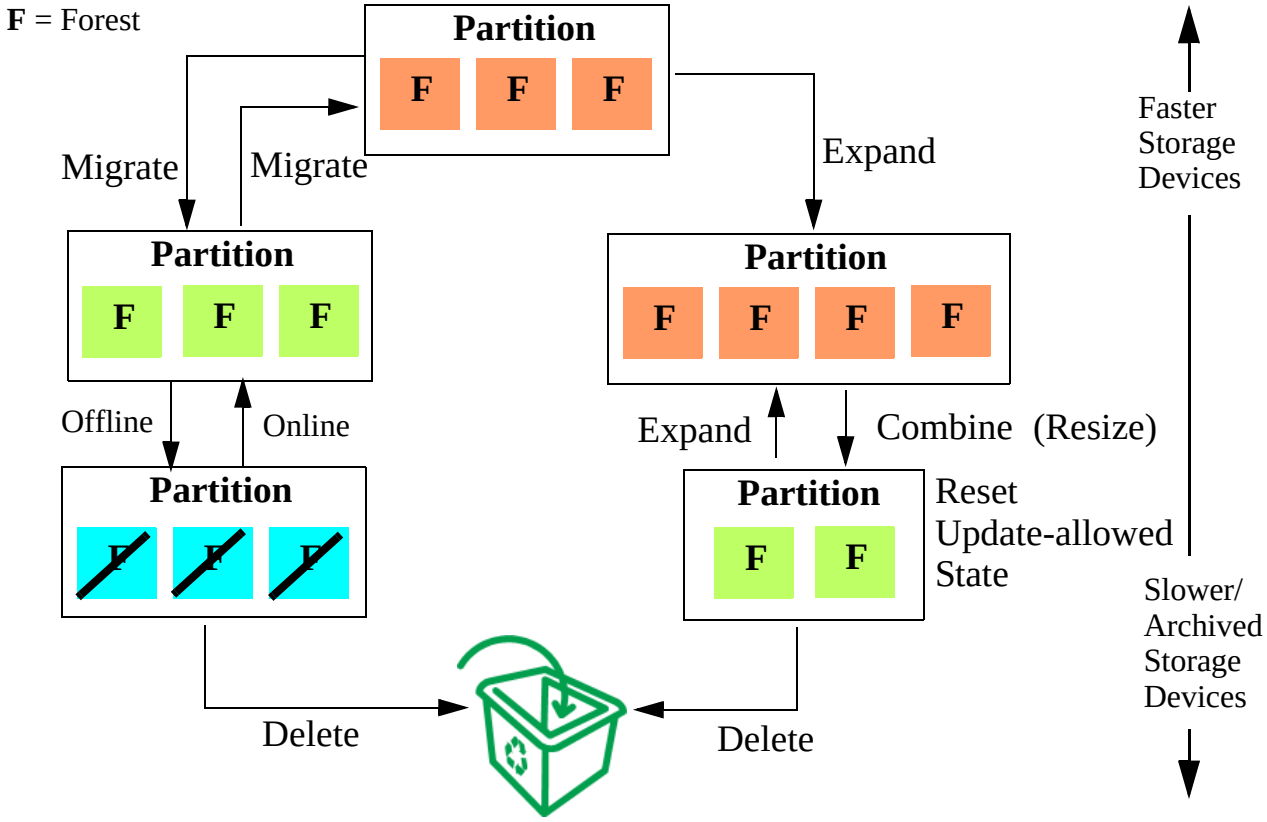


As data ages and becomes less updated and queried, it can be migrated to less expensive and more densely-packed storage devices to make room for newer, more frequently accessed and updated data, as illustrated in the graph below.



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.

19.3 Partitions, Partition Keys, and Partition Ranges

MarkLogic Server tiered storage manages data in *partitions*. Each partition consists of a group of database forests that share the same name prefix and the same *partition range*.

> **Note:** When deploying forests in a cluster, you should align forests and forest replicas across hosts for parellelization 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.

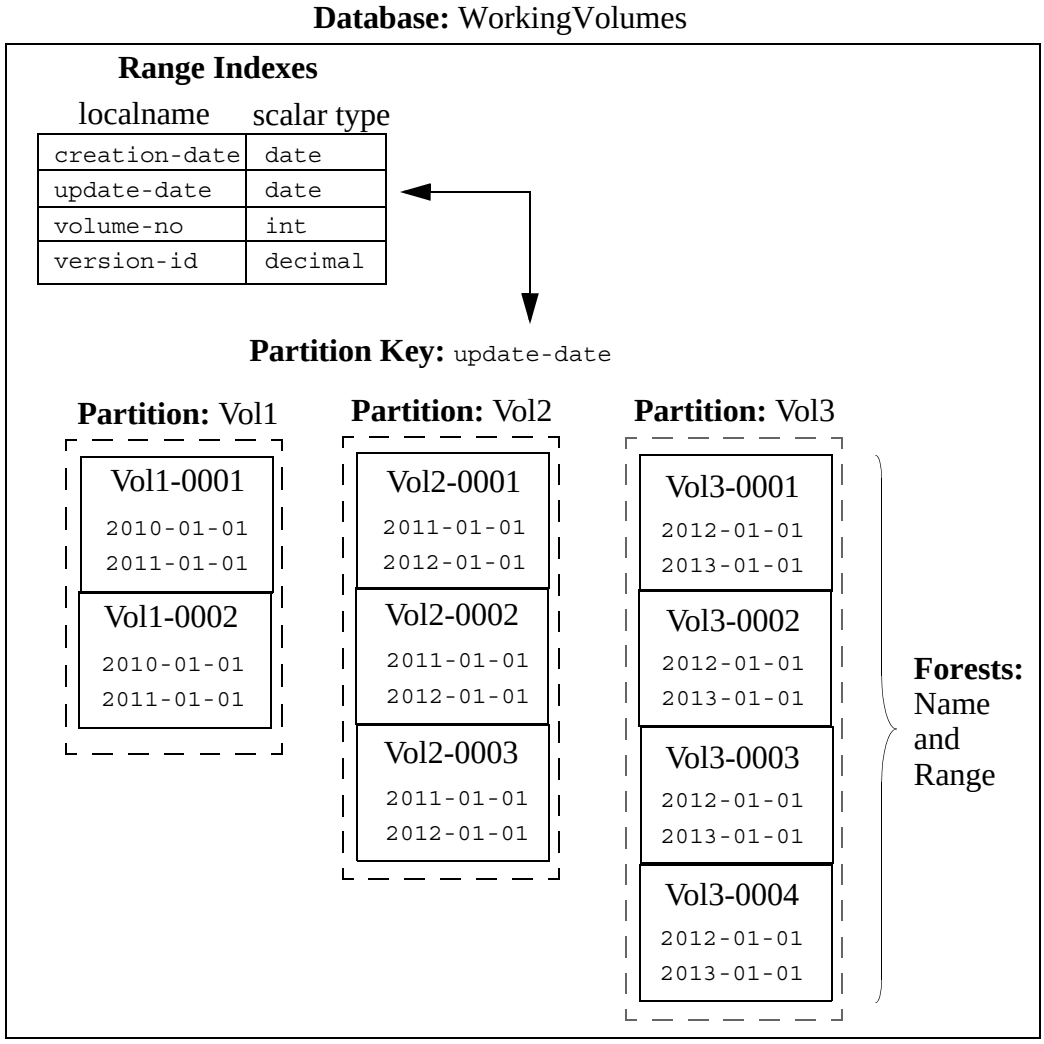
For example, you have a database, named `WorkingVolumes`, that contains nine forests that are grouped into three partitions. Among the range indexes in the `WorkingVolumes` database is an element range index for the `update-date` element with a type of `date`. The `WorkingVolumes` database has its partition 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 (*prefix-name*)	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.

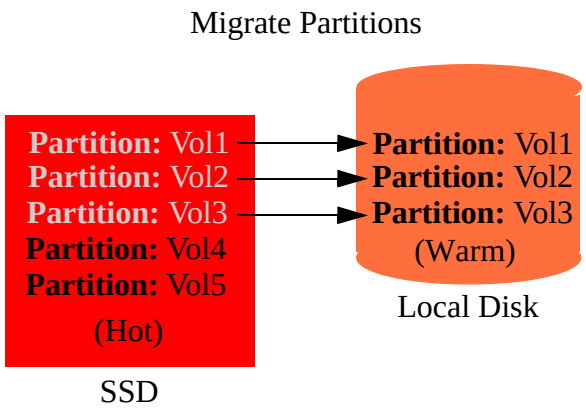
In this example, a document with an `update-date` element value of `2011-05-22` would be stored in one of the forests in the `Vol2` partition. Should the `update-date` element value in the document get updated to `2012-01-02` or later, the document will be automatically moved to the `Vol3` partition. How the documents are redistributed among the partitions is handled by the database rebalancer, as described in “Range Assignment Policy” on page 190.

Below is an illustration of the `WorkingVolumes` database, showing its range indexes, partition key, and its partitions and forests.

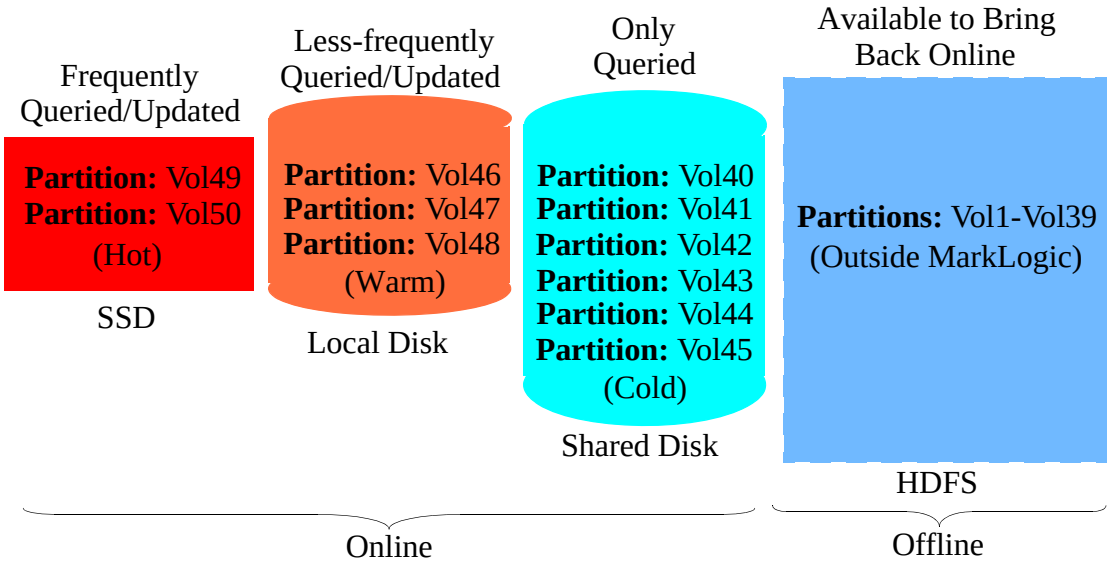


Contains SAP BusinessObjects Text Analysis XI from SAP AG. Copyright © 1996-2011. All rights reserved.

In 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.

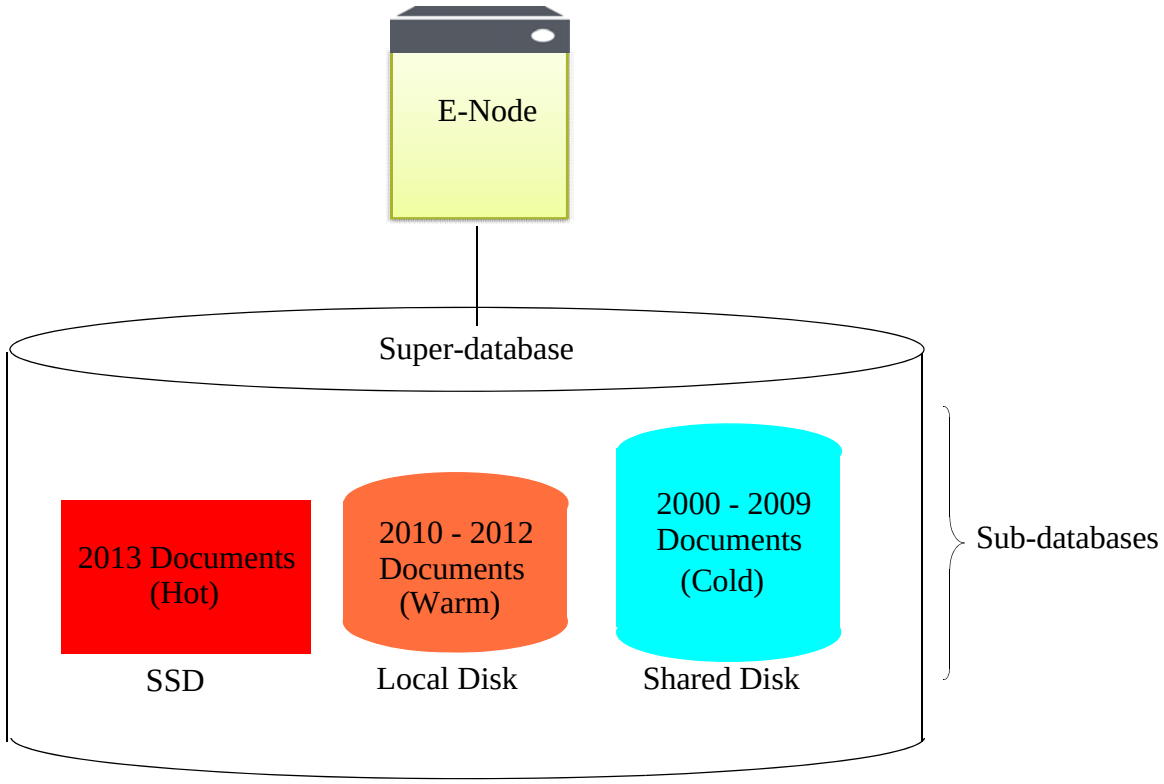


After years of data growth, the volumes of documents grow to 50. After migrating between storage tiers, the partitions are eventually distributed among the storage tiers, as shown below.



Contains SAP BusinessObjects Text Analysis XI from SAP AG. Copyright © 1996-2011. All rights reserved.

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.



19.4 Configuring a Database to Participate in Tiered Storage

If a database is to participate in a tiered storage scheme, it must have the following set:

- 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 "Range Indexes and Lexicons" on page 340
- A partition key, as described in "Defining a Partition Key" on page 211
- Partitions, as described in "Creating Partitions" on page 213

Warning All of the forests in a database configured for tiered storage must be part of a partition.

For details on how to configure the database rebalancer with the range assignment policy, see the sections "Range Assignment Policy" on page 190, "Configuring the Rebalancer on a Database" on page 194, and "Configuring the Rebalancer on a Forest" on page 196.

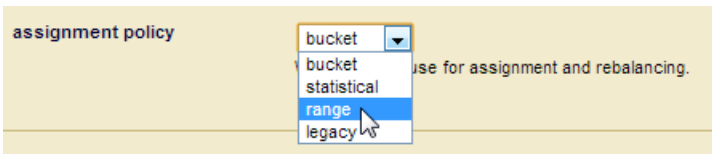
19.5 Defining a Partition Key

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:

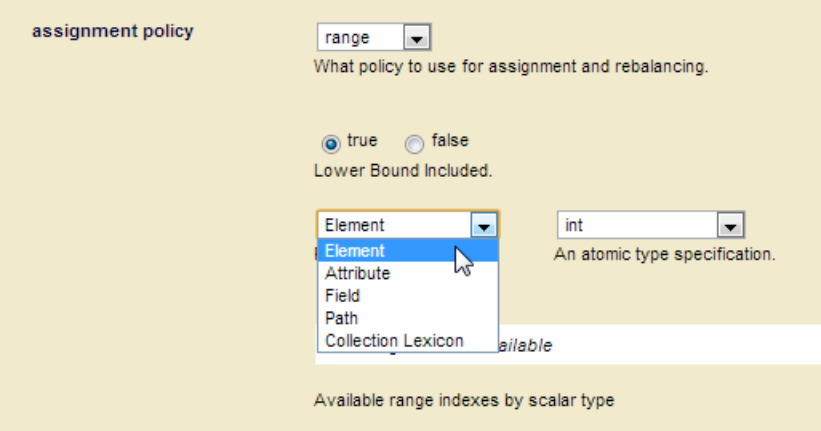
1. Create an element range index, named `update-date`, on the database of type `date`. The details on how to create an element range index are described in "Defining Element Range Indexes" on page 351.

2. In the Admin UI, open the configuration page for the database, set the assignment policy to range. Additional settings appear under the assignment policy.

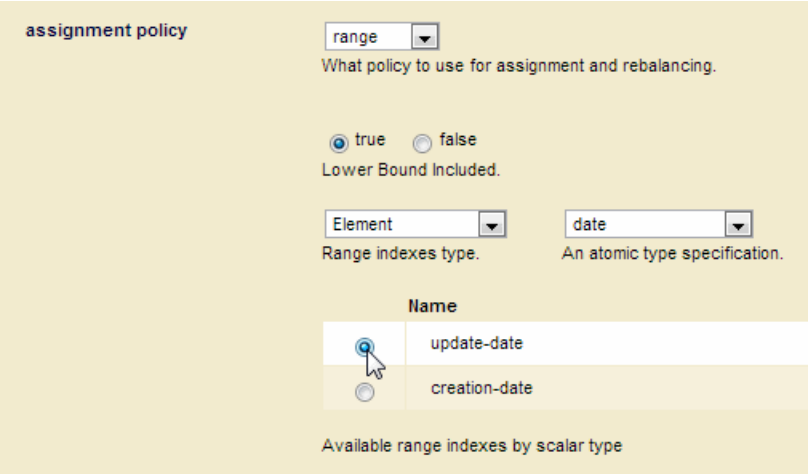


3. Set the `Lower Bound Included` to `true` if you want to include documents with a partition key value that matches the lower bound value and exclude documents that match the upper bound value. Set the `Lower Bound Included` to `false`, if you want to exclude documents with a partition key value that 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.

4. Note the type and scalar type of the range index, field, or collection lexicon you want to use as your partition key. In this example, we use an `Element` range index with a scalar type of `date`. Set the index and scalar types in the drop down menus to list the matching range indexes, fields, or collection lexicons set for the database.



5. Select the range index, field, or collection lexicon you want to use as your partition key, which is `update-date` in this example.



19.6 Creating Partitions

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.

There are two ways to create a partition:

- Creating a Partition with New Forests
- Creating a Partition from Existing Forests

19.6.1 Creating a Partition with New Forests

You can use the `POST:/manage/v2/databases/{id|name}/partitions` REST resource address to create a new partition with empty forests. When creating a partition, you specify the partition range and the number of forests to be created for the partition. You can also specify that the partition be created for multiple hosts, in which case the specified number of forests will created on each host.

For example, the following creates a 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 “Partitions with Forest-Level Failover” on page 231.

19.6.2 Creating a Partition from Existing Forests

You can create a partition from existing forests simply by renaming the forests so that they adhere to a 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 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 partition. Default partitions store the documents that have partition key values that do not fit into any of the ranges set for the other partitions.

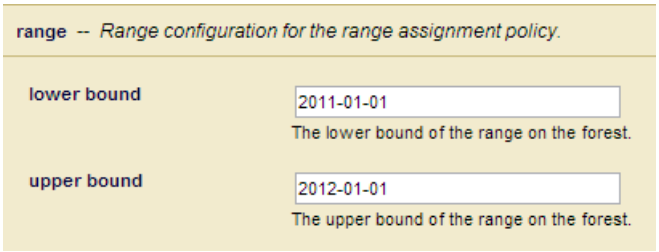
For example, to rename the `1-2011` forest to `2011-1` and set the range to `2011-01-01` - `2012-01-01`, do the following:

1. Open the Forest Configuration page in the Admin UI, as described in “Creating a Forest” on page 275.

2. In the forest name field, change the name from `1-2011` to `2011-1`:



3. In the range section of the Forest Configuration page, set the lower bound value to `2011-01-01` and the upper bound value to `2012-01-01`:



4. 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.

19.7 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 set-up 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 chapter.

The topics in this section are:

- Asynchronous Operations
- Privileges
- /manage/v2/databases/{id|name}/partitions
- /manage/v2/databases/{id|name}/partitions/{name}
- /manage/v2/databases/{id|name}/partitions/{name}/properties
- /manage/v2/databases/{id|$name}/sub-databases
- /manage/v2/databases/{id|name}/sub-databases/{id|name}
- /manage/v2/databases/{id|name}/super-databases
- /manage/v2/databases/{id|name}/super-databases/{id|name}
- /manage/v2/forests
- /manage/v2/forests/{id|name}
- /manage/v2/forests/{id|name}/properties

19.7.1 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 the 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/868180999119846221 4?view=process-status
```

and your host is `MyHost`, you can view the status of your operation using the following URL:

```
http://MyHost:8002/manage/v2/tickets/868180999119846221 4?view=process-
status
```

Note: Historical ticket information can always be accessed by viewing the ticket default view.

19.7.2 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.

19.7.3 /manage/v2/databases/{id|name}/partitions

Method	Description	Parameters	XQuery Equivalent
GET	Gets a list of partitions on the database	format? (json | xml)	`tieredstorage:database-partitions`
POST	Add a partition to the database	format? (json | xml)	`tieredstorage:partition-create`

For examples, see:

- Viewing Partitions
- Creating Partitions

19.7.4 /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)	`tieredstorage:partition-forests`
DELETE	Deletes the partition	delete-data? (true|false)	`tieredstorage:partition-delete`

PUT	Invokes one of the following operations on the partition:
	• resize (asynchronous)
	• transfer (synchronous)
	• migrate (asynchronous)
		format? (json | xml)	`tieredstorage:partition-resize`
			`tieredstorage:partition-transfer`
			`tieredstorage:partition-migrate`

For examples, see:

- Deleting Partitions
- Resizing Partitions
- Transferring Partitions between Databases
- Migrating Forests and Partitions

19.7.5 /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 partition properties (updates-allowed, online | offline)	format? (json | xml)	`tieredstorage:partition-set-availability`
			
			`tieredstorage:partition-set-updates-allowed`

For examples, see:

- Taking Forests and Partitions Online and Offline
- Setting the Updates-allowed State on Partitions

19.7.6 /manage/v2/databases/{id|$name}/sub-databases

Method	Description	Parameters	XQuery Equivalent
GET	Gets a list of sub-databases associated with the specified database.	format? (json | xml)	`admin:database-sub-databases`
POST	Creates a new database attached as a sub-database.	format? (json | xml)	`tieredstorage:database-create-sub-database`

For an example, see:

- Creating Sub-databases

19.7.7 /manage/v2/databases/{id|name}/sub-databases/{id|name}

Method	Description	Parameters	XQuery Equivalent
GET	Gets a summary of the sub-database, including a list of sibling sub-databases if any.	format? (json | xml)	`admin:database-sub-databases`
DELETE	Detaches the sub-database from the specified (super) database and deletes the sub-database.		`tieredstorage:database-delete-sub-database`

For an example, see:

- Viewing Super-databases and Sub-databases

19.7.8 /manage/v2/databases/{id|name}/super-databases

Method	Description	Parameters	XQuery Equivalent
GET	Gets a list of super-databases associated with the specified database.	format? (json | xml)	`admin:database-super-databases`
POST	Creates a new database defined as super-database of the specified database.	format? (json | xml)	`tieredstorage:database-create-super-database`

For an example, see:

- Creating Super-databases

19.7.9 /manage/v2/databases/{id|name}/super-databases/{id|name}

Method	Description	Parameters	XQuery Equivalent
GET	Gets a summary of the super-database, including a count of sub-databases.	format? (json | xml)	`admin:database-super-databases`
DELETE	Detaches the super-database from the specified (super) database and deletes the super-database.		`tieredstorage:database-delete-super-database`

For an example, see:

- Viewing Super-databases and Sub-databases

19.7.10 /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:
	• forest-combine
	• forest-migrate
	These operations are asynchronous
		format? (json | xml)	`tieredstorage:forest-combine`
			
			`tieredstorage:forest-migrate`

For examples, see:

- Migrating Forests and Partitions
- Combining Forests

19.7.11 /manage/v2/forests/{id|name}

Method	Description	Parameters	XQuery Equivalent
GET	Gets a summary of the forest.	format? (json | xml) view	`admin:forest-get-*`

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:

- Retiring Forests

19.7.12 /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`

19.8 Common Forest and Partition Operations

This section describes the following partition operations:

- Viewing Partitions
- Migrating Forests and Partitions
- Resizing Partitions
- Transferring Partitions between Databases
- Combining Forests
- Retiring Forests
- Taking Forests and Partitions Online and Offline
- Setting the Updates-allowed State on Partitions
- Deleting Partitions

Some of these operations operate asynchronously and immediately return a ticket number that you can use to check the status of the operation. For example, if the following is returned:

```
<link><kindref>process-status</kindref><uriref>/manage/v2/tickets/467851692005
7381194?view=process-status</uriref></link>
```

You can check the status of the operation by entering a resource address like the following:

```
http://MyHost:8002/manage/v2/tickets/467851692005738119 4?view=process-
status
```

For details on asynchronous processes, see “Asynchronous Operations” on page 216.

19.8.1 Viewing Partitions

You can return all of the information on a partition. For example, to return the details of the `2011` partition on the `Documents` database, do the following:

```
curl -X GET --anyauth --user admin:admin --header \
"Content-Type:application/xml" \
http://MyHost:8002/manage/v2/databases/Documents/partitions/2011
```

19.8.2 Migrating Forests and Partitions

Forests and partitions can be migrated from one storage device to another. For example, a partition on an SSD has aged to the point where is it less frequently queried and can be moved to a slower, less expensive, storage device to make room for a more frequently queried partition.

For example, the `2011` partition on the `Documents` database is mounted on a local disk on the host, `MyHost`. To migrate the `2011` partition to the `/warm-storage` data directory mounted on a shared disk on the host, `OurHost`, do the following:

```
$ 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`, do the following:

```
$ 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.

19.8.3 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` partition up to five forests, do the following:

```
$ 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 “Migrating Forests and Partitions” on page 224.

Note: If you resize partitions for databases configured for database replication, first resize the replica partitions before resizing the master partitions.

19.8.4 Transferring Partitions between Databases

You can use the `PUT:/manage/v2/databases/{id|name}/partitions/{name}` resource address to move a partition from one database to another. For example, to transfer the `2011` partition from the `Documents1` database to the `Documents2` database, do the following:

```
$ cat transfer-partition.xml
<transfer xmlns="http://marklogic.com/manage">
  <destination-database>Documents2</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/Documents1/partitions/2011'
```

19.8.5 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`, do the following:

```
$ 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 “Migrating Forests and Partitions” on page 224.

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.

Note: 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.

19.8.6 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 "How Data is Moved when a Forest is Retired from the Database" on page 194.

For example, to retire the forest, `2011`, from the `Documents` database, do the following:

```
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
```

19.8.7 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` partition in the `Documents` 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/Documents/partitions/2011/prop
erties'
```

19.8.8 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. **Note:** Resizing a read-only partition to fewer forests preserves its original forests.
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 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/prop
erties'
```

19.8.9 Deleting Partitions

You can delete a partition, along with all its forests. For example, to delete the 2011 partition from the `Documents` database, do the following:

```
$ curl --anyauth --user user:password -X DELETE \
-H 'Content-type: application/xml' \
'http://MyHost:8002/manage/v2/databases/Documents/partitions/2011'
```

19.9 Sub-databases and Super-databases

Multiple databases can be grouped into a super-database in order to allow a single query to be done across multiple databases distributed on different storage tiers. Databases contained in a super-database are called sub-databases. A sub-database can be either `active` (online) or `archive` (offline), as specified by the `kind` element.

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.

Note: Sub-databases and their super-databases must have the same index settings. Otherwise, queries will not work.

Note: 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.

The operations described in this section are:

- Creating Sub-databases
- Creating Super-databases
- Viewing Super-databases and Sub-databases

19.9.1 Creating Sub-databases

You can call the `POST:/manage/v2/databases/{id|name}/sub-databases` resource address to create a sub-database. Creating a sub-database for an existing database automatically designates the existing database as a super-database. The new sub-database inherits all of the settings from the super-database.

For example, to define the `subdb1` database as an `active` sub-database of the `superdb1` database, do the following:

```
$ cat create-subdatabase.xml
<sub-database xmlns="http://marklogic.com/manage">
  <database-name>subdb1</database-name>
  <kind>active</kind>
</sub-database>

$ curl --anyauth --user user:password -X POST \
-d@"./create-subdatabase.xml" -H 'Content-type: application/xml' \
'http://MyHost:8002/manage/v2/databases/superdb1/sub-databases'
```

19.9.2 Creating Super-databases

You can call the `POST:/manage/v2/databases/{id|name}/super-databases` resource address to create a super-database. Creating a super-database for an existing database automatically designates the existing database as a sub-database. The new super-database inherits all of the settings from the sub-database.

For example, to define the `superdb1` database as a super-database containing the `subdb1` sub-database, do the following:

```
$ cat create-superdatabase.xml
<super-database xmlns="http://marklogic.com/manage">
  <database-name>superdb1</database-name>
</super-database>

$ curl --anyauth --user user:password -X POST \
-d@"./create-superdatabase.xml" -H 'Content-type: application/xml' \
'http://MyHost:8002/manage/v2/databases/subdb1/super-databases'
```

19.9.3 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'
```

19.10 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 chapters 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 partition and configure the forests for local-disk failover between `MyHost1`, `MyHost2`, and `MyHost3`, do the following.

```
$ 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 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.

20.0 Backing Up and Restoring a Database

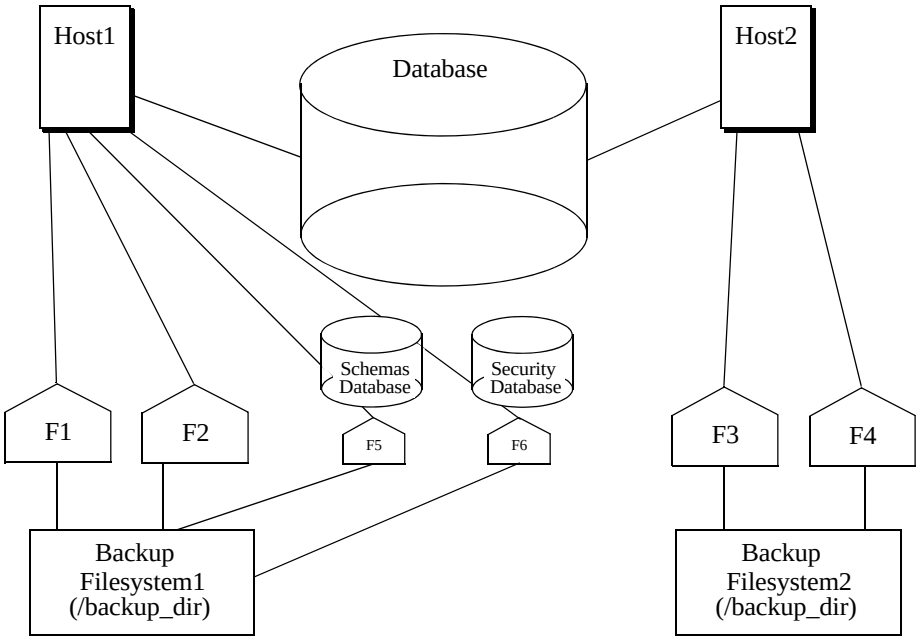
MarkLogic Server provides a facility to make a consistent backup of a database. This section describes the backup and restore architecture and provides procedures for backing up and restoring a database. The following topics are included:

- Backup and Restore Overview
- Backing Up Databases with Journal Archiving
- Incremental Backup
- Backing Up a Database
- Restoring a Database without Journal Archiving
- Restoring Databases with Journal Archiving
- Restoring from an Incremental Backup with Journal Archiving
- Backing up and Restoring a Database Following Local Disk Failover

20.1 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.



20.1.1 Consistent, Database-Level Backup

By default, when you back up a database you backup everything associated with it, including the following:

- 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 backup any individual forests that you choose, choosing only the ones you need to backup. 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 backup 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 backup or restore. The choices you make depend on the amount of change in your system and your unique backup and restore requirements.

20.1.2 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 “Backing Up a Database” on page 245 and “Restoring a Database without Journal Archiving” on page 251.

20.1.3 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.

20.1.4 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 must be readable and writable by the user running MarkLogic Server (by default `daemon` on UNIX and the local System user on Windows). 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 contain one backup. The following 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` directory on September 1, 2004, a directory structure similar to the following is created:

```
/space/backups
   20040901-1/
      *.xml
      BackupTag.txt
      Forests/
         Documents/
            Label
            000001e1/
            Journals/
         Schemas/
            Label
            000001e1/
            Journals/
         Security/
            Label
            000001e1/
            Journals/
         Triggers/
            Label
            000001e1/
            Journals/
```

Incremental backups are stored in the directory under the full backup. In this first example, the backup directory (backup-dir) is `/space/backup` and the incremental backup directory (incremental-dir) is not used:

```
/space/backups
        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/backup` and the incremental backup directory (incremental-dir) is `/space/incremental`.

```
/space/backups
        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.

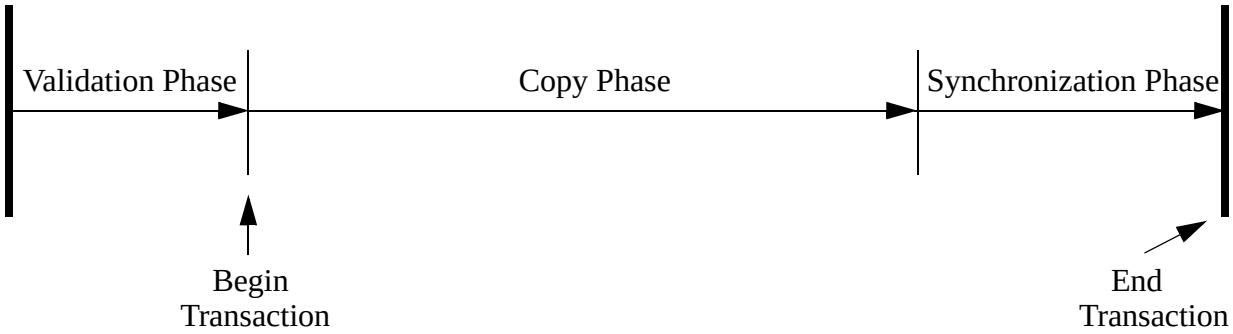
> **Note:** Once you restore an incremental backup, you can no longer use the previous full backup location for ongoing incremental backups. After the restore, you need to make a fresh full backup and use the full backup location for ongoing incremental backups. This means that after restore of an incremental backup, scheduled backups need to be updated to use the fresh full backup location.

20.1.5 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:



20.1.5.1 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.

20.1.5.2 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.

20.1.5.3 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 "Appendix A: 'Hot' versus 'Cold' Admin Tasks" on page 414.

20.1.6 Notes about Backup and Restore Operations

This section provides notes and restrictions about backing up and restoring MarkLogic Server databases.

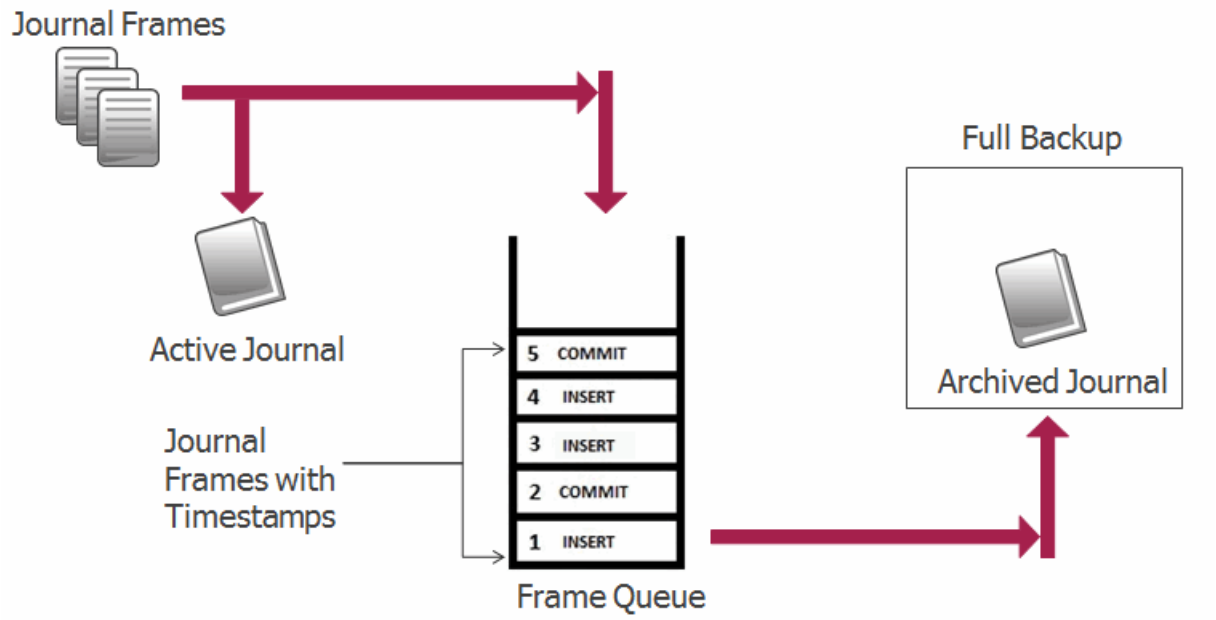
- The backup files are platform specific—backups on a given platform should only be restored onto the same platform. This is true for both database and forest backups.
- You can restore an individual forest using a database backup by unchecking all forests except the one you want to restore on the Confirm Restore screen (see step 10 in "Restoring a Database without Journal Archiving" on page 251).
- 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 "Restoring a Forest" on page 286.
- 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 `xdmp: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 (`/etc/init.d/MarkLogic` 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.

20.2 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.

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 reenable 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.

20.3 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 the hour or the minute) because an incremental backup takes less time to do than a full backup.

To enable an incremental backup, set Incremental backup to `true` while initiating or scheduling a backup. See "Backing Up a Database" on page 245 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 "Backing Up Databases with Journal Archiving" on page 241 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.

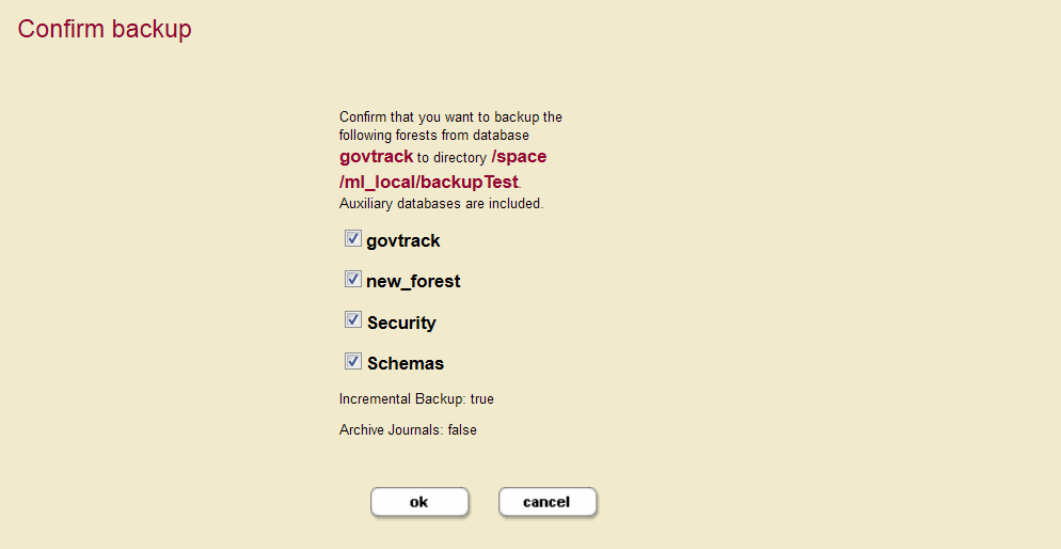
Incremental backup and journal archiving both provide disaster recovery. Incremental backup uses less disk space than journal archiving, and incremental backup is faster than using journal archiving.

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 "Restoring from an Incremental Backup with Journal Archiving" on page 258 for details.

20.3.1 Incremental Backup of New Forest

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.

After you attach a new forest to your database, it will be included in the list of forests to be backed up in the Confirm backup step (Step 11 in "Backing Up a Database Immediately" on page 245).



Select the forest to include it in the backup and click ok. See "Backing Up a Database" on page 245 for more information.

20.4 Incremental Backup with Journal Archiving

Incremental backup improves restore both time and space 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, we suggest the following:

1. Configure a scheduled full backup at some coarse granularity (for example, weekly) and enable journal archiving

2. Configure a scheduled incremental backup as some finer granularity (for example, hourly), 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 "Setting Merge Policy" on page 171 or `admin:database-set-retain-until-backup` for details.

This configuration means that journal archives are only needed for the most recent hour, and the older ones are purged once there is an incremental backup that covers that hour. 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.

20.5 Backing Up a Database

You can either initiate a database backup immediately or you can schedule a backup to occur in the future with the following procedures:

- Backing Up a Database Immediately
- Scheduling a Database Backup

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.

20.5.1 Backing Up a Database Immediately

Perform the following steps to initiate a database backup:

1. Log into the Admin Interface as a user with the `admin` role.

2. Click the Databases link in the left menu of the Admin Interface.

3. Click the database name for the database you want to back up, either from the tree menu or on the summary page.

4. Click the Backup/Restore tab. The Backup/Restore screen appears.

5. Enter the directory to which you want the database backed up in the Backup to directory field.

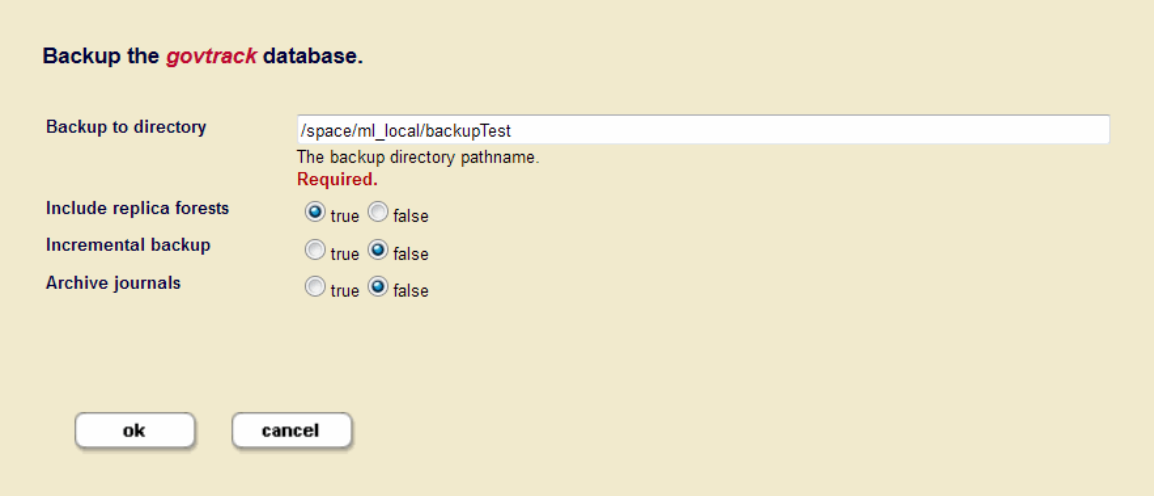
   Note: The backup directory path must exist on all hosts that serve any forests in the database. The directory you specified can be an operating system mounted directory path, it can be an HDFS path, or it can be an S3 path. For details on using HDFS and S3 storage in MarkLogic, see Disk Storage Considerations 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 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*.

7. Set Incremental backup to true to create an incremental backup. The default is a full backup (false).

8. 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 “Backing Up Databases with Journal Archiving” on page 241.

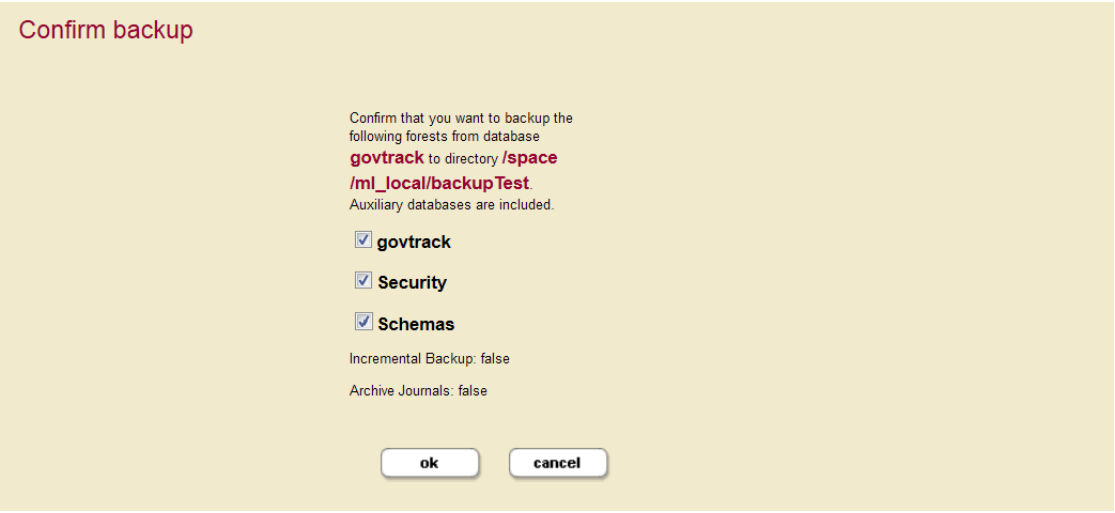
   Note: If Journal Archiving is enabled, you cannot include auxiliary forests, as they should have their own separate backups.



9. Click OK.

10. 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.

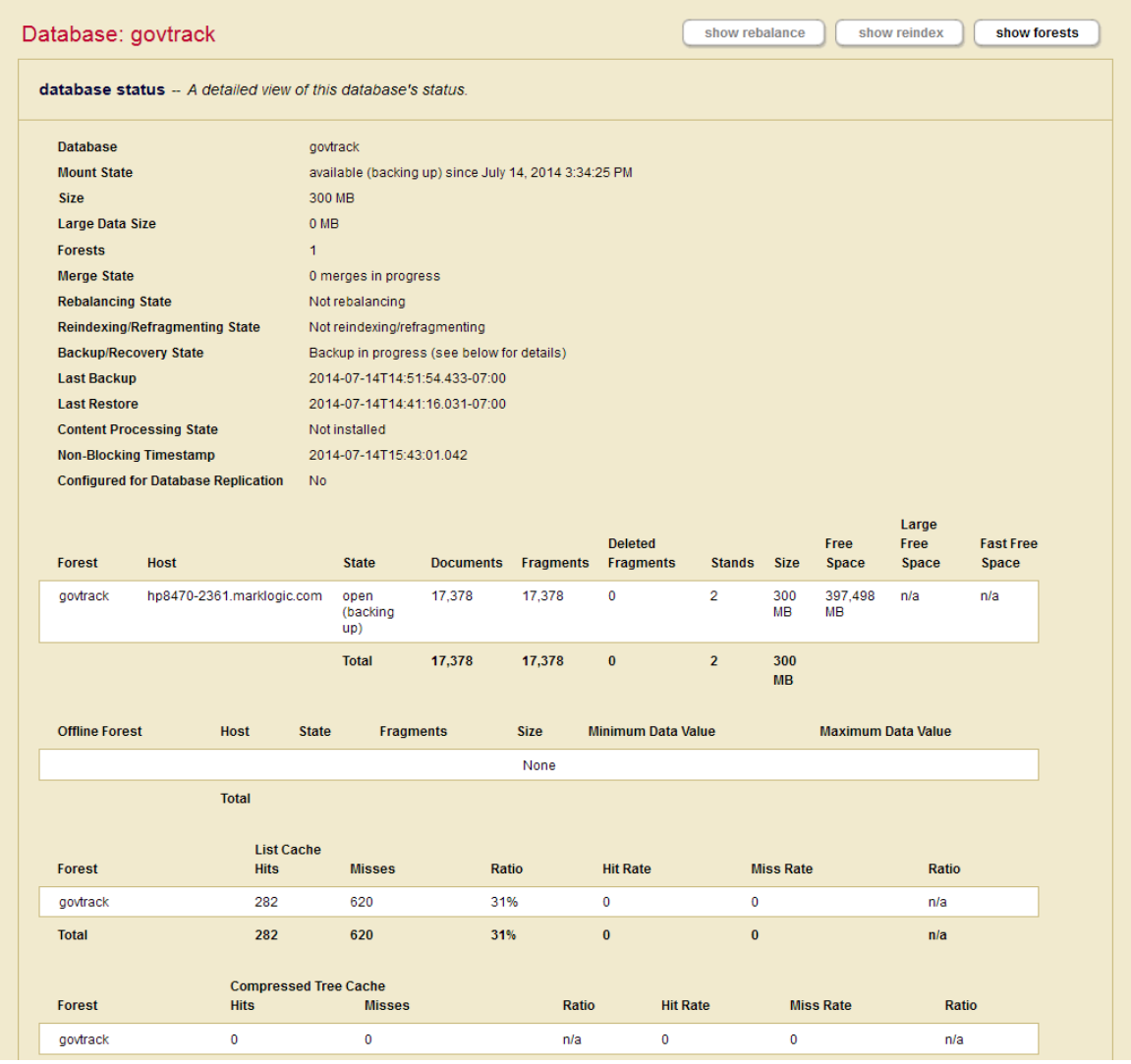
11. The Confirm backup screen appears and lists all the forest selected for back up.



12. 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.

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



Database: govtrack

show rebalance | show reindex | show forests

database status -- *A detailed view of this database's status.*

Database	govtrack
Mount State	available (backing up) since July 14, 2014 3:34:25 PM
Size	300 MB
Large Data Size	0 MB
Forests	1
Merge State	0 merges in progress
Rebalancing State	Not rebalancing
Reindexing/Refragmenting State	Not reindexing/refragmenting
Backup/Recovery State	Backup in progress (see below for details)
Last Backup	2014-07-14T14:51:54.433-07:00
Last Restore	2014-07-14T14:41:16.031-07:00
Content Processing State	Not installed
Non-Blocking Timestamp	2014-07-14T15:43:01.042
Configured for Database Replication	No

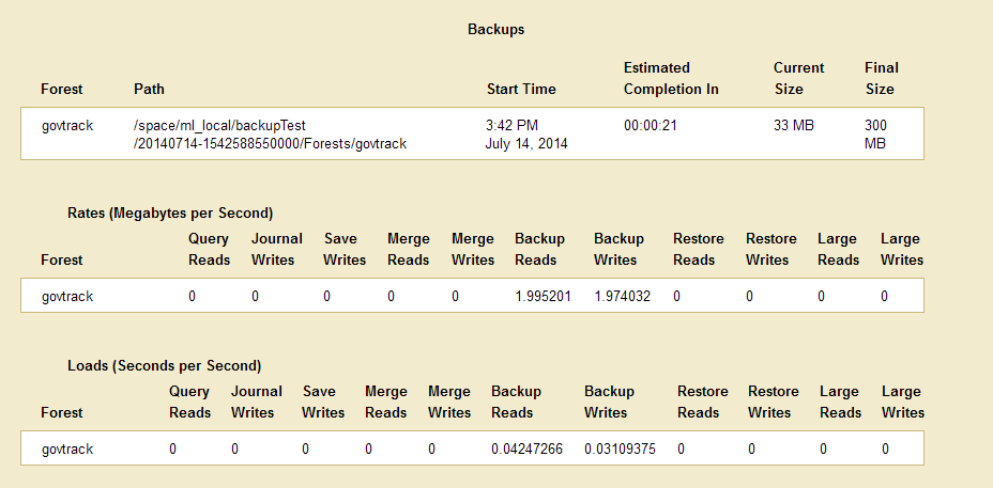
Forest	Host	State	Documents	Fragments	Deleted Fragments	Stands	Size	Free Space	Large Free Space	Fast Free Space
govtrack	hp8470-2361.marklogic.com	open (backing up)	17,378	17,378	0	2	300 MB	397,498 MB	n/a	n/a
		Total	**17,378**	**17,378**	**0**	**2**	**300 MB**			

Offline Forest	Host	State	Fragments	Size	Minimum Data Value	Maximum Data Value
				None		
	Total					

Forest	List Cache Hits	Misses	Ratio	Hit Rate	Miss Rate	Ratio
govtrack	282	620	31%	0	0	n/a
Total	**282**	**620**	**31%**	**0**	**0**	**n/a**

Forest	Compressed Tree Cache Hits	Misses	Ratio	Hit Rate	Miss Rate	Ratio
govtrack	0	0	n/a	0	0	n/a

14. 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.



Backups

Forest	Path	Start Time	Estimated Completion In	Current Size	Final Size
govtrack	/space/ml_local/backupTest /20140714-1542588550000/Forests/govtrack	3:42 PM July 14, 2014	00:00:21	33 MB	300 MB

Rates (Megabytes per Second)

Forest	Query Reads	Journal Writes	Save Writes	Merge Reads	Merge Writes	Backup Reads	Backup Writes	Restore Reads	Restore Writes	Large Reads	Large Writes
govtrack	0	0	0	0	0	1.995201	1.974032	0	0	0	0

Loads (Seconds per Second)

Forest	Query Reads	Journal Writes	Save Writes	Merge Reads	Merge Writes	Backup Reads	Backup Writes	Restore Reads	Restore Writes	Large Reads	Large Writes
govtrack	0	0	0	0	0	0.04247266	0.03109375	0	0	0	0

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.

20.5.2 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, perform the following steps using the Admin Interface:

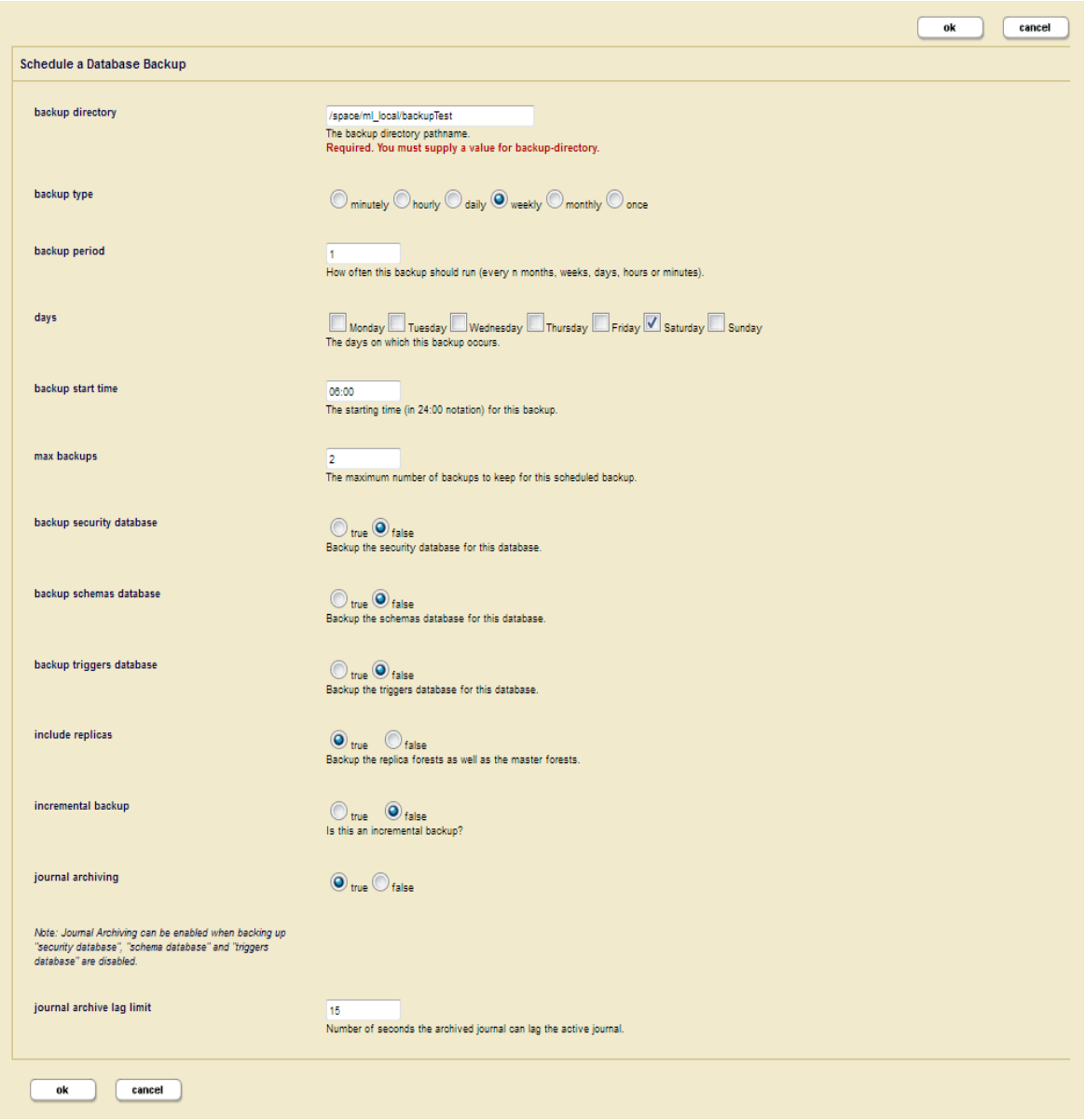
1. Click the Databases icon on the left tree menu.

2. Select the database for which you want to schedule a backup, either on the tree menu or from the Database Summary page. The Database Configuration page appears.

3. Click the Scheduled Backup link in the tree menu for the database. The Scheduled Backup Configuration page appears.

4. On the Scheduled Backup Configuration page, you can delete any existing scheduled backups if you no longer need them.

5. Click the Create tab. The Schedule a Database Backup page appears:



6. 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. Choose a scheduled or one-time for the backup type:

    - For minutely, enter how many minutes between each backup.
    - For 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 “Backing Up Databases with Journal Archiving” on page 241.

    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.

20.6 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 “Restoring Databases with Journal Archiving” on page 254.

To restore an entire database from a backup, perform the following steps:

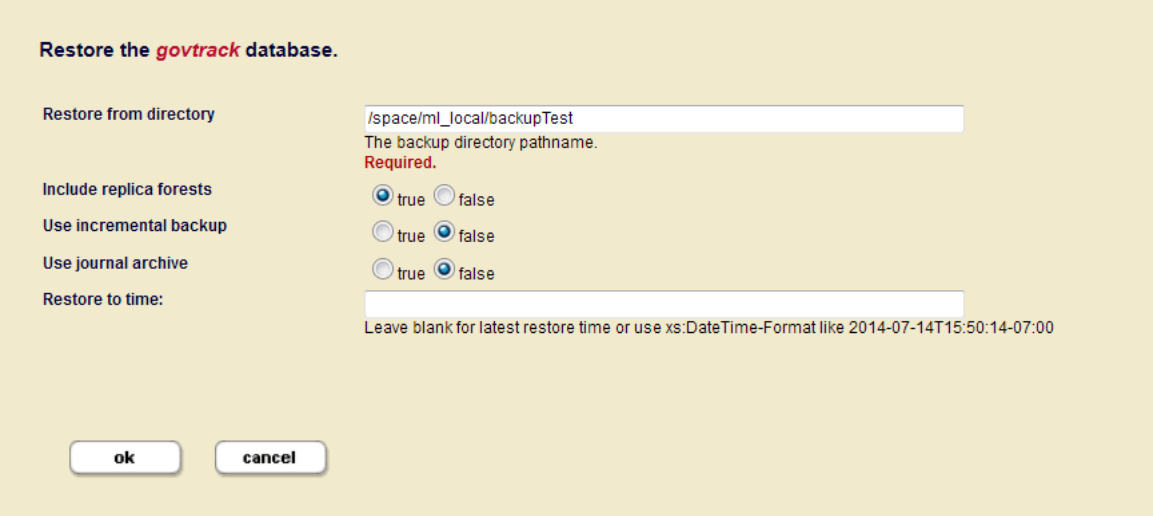
1. Log into the Admin Interface as a user with the `admin` role.
2. Click the Databases link in the left menu of the Admin Interface.
3. Click the database name for the database you want to restore, either on the tree menu or on the summary page. This database should have the same configuration settings (index options, fragmentation, range indexes) as the one that was backed up.
4. Click the Backup/Restore tab. The Backup/Restore screen appears.
5. Enter the directory in which the back up exists in the Restore from directory field.

   > **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 “Backup Directory Structure” on page 236.

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. If you want to restore an incremental back up, set Use incremental backup to `true`.

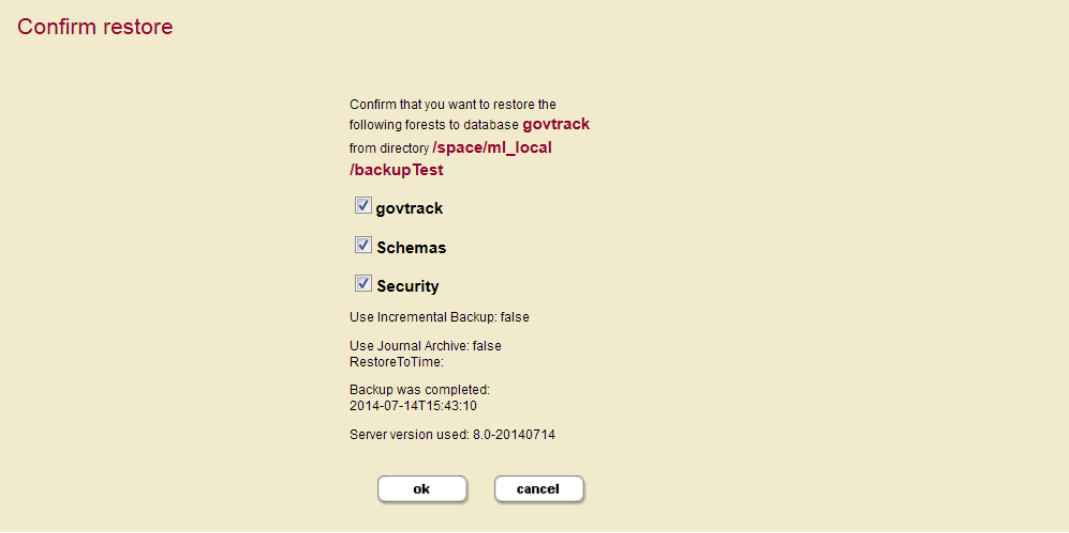
   > **Note:** If you restore from an incremental backup, you can’t use the previous full backup location for ongoing incremental backups. You need to make a fresh full backup after the restore and use the full backup location for the ongoing incremental backups. After doing a restore from an incremental backup, any scheduled backups will need to be updated to use the new full backup location.

8. Leave Journal Archiving false.



9. Click OK.

10. The Confirm restore screen appears and lists all the forest selected for restoring.



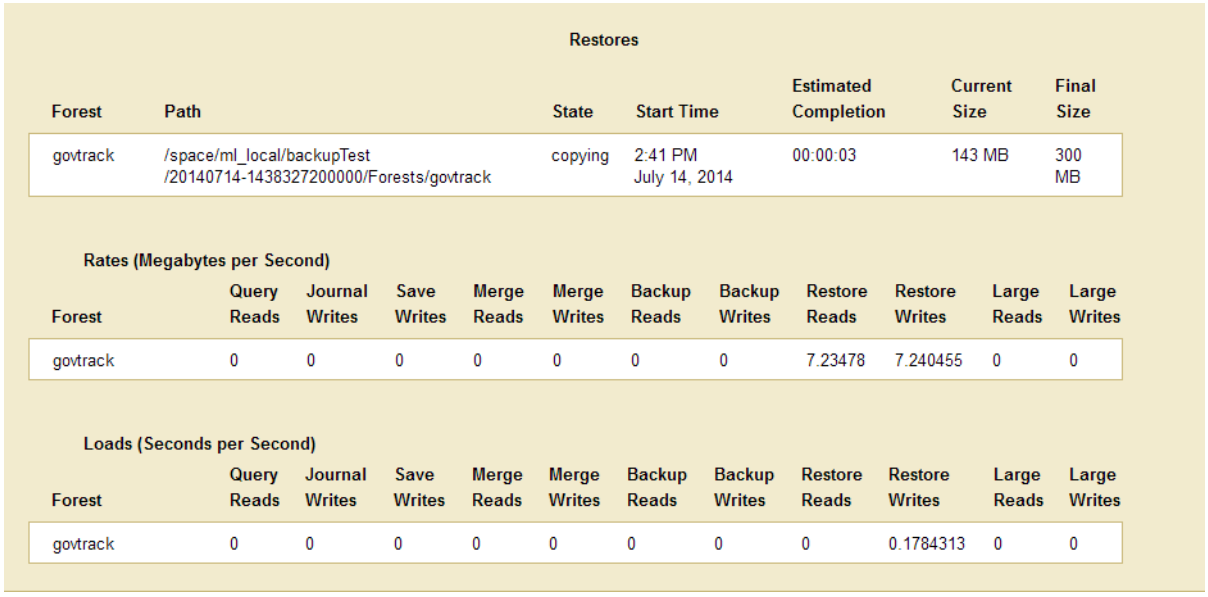
The Confirm restore screen also lists the date the backup was performed and the server version used for the backup you selected.

11. 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.

12. 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.



Restores

Forest	Path	State	Start Time	Estimated Completion	Current Size	Final Size
govtrack	/space/ml_local/backupTest /20140714-1438327200000/Forests/govtrack	copying	2:41 PM July 14, 2014	00:00:03	143 MB	300 MB

Rates (Megabytes per Second)

Forest	Query Reads	Journal Writes	Save Writes	Merge Reads	Merge Writes	Backup Reads	Backup Writes	Restore Reads	Restore Writes	Large Reads	Large Writes
govtrack	0	0	0	0	0	0	0	7.23478	7.240455	0	0

Loads (Seconds per Second)

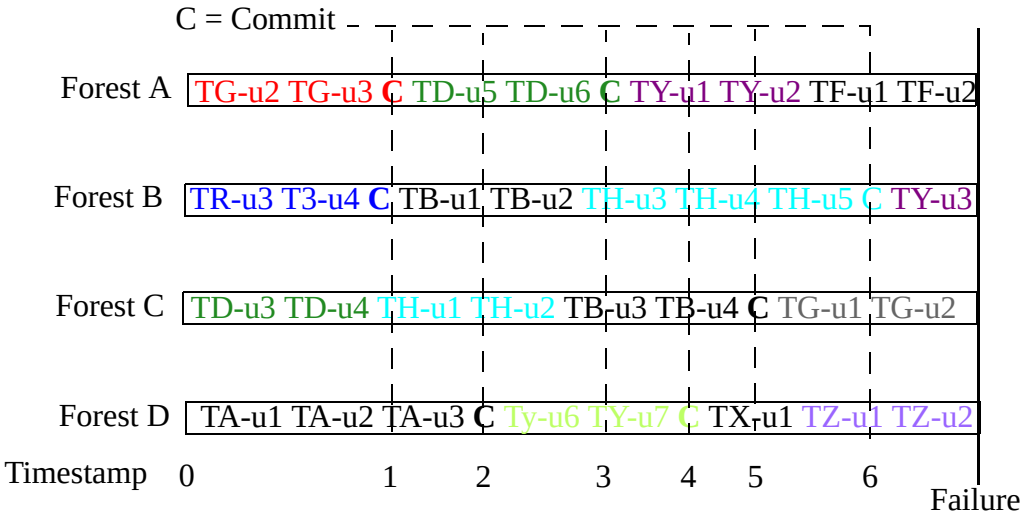
Forest	Query Reads	Journal Writes	Save Writes	Merge Reads	Merge Writes	Backup Reads	Backup Writes	Restore Reads	Restore Writes	Large Reads	Large Writes
govtrack	0	0	0	0	0	0	0	0	0.1784313	0	0

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.

20.7 Restoring Databases with Journal Archiving

After you restore a database with Journal Archiving enabled, each forest will likely have comitted 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 "Restoring to the Safe Timestamp" on page 255.
- Restore data at a specific timestamp. Follow the procedure described in "Restoring to a Specific Timestamp" on page 257.
- Restore data at a specific timestamp based on the state of some sample documents. Follow the procedure described in "Restoring Based on Sample Documents" on page 257.

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.

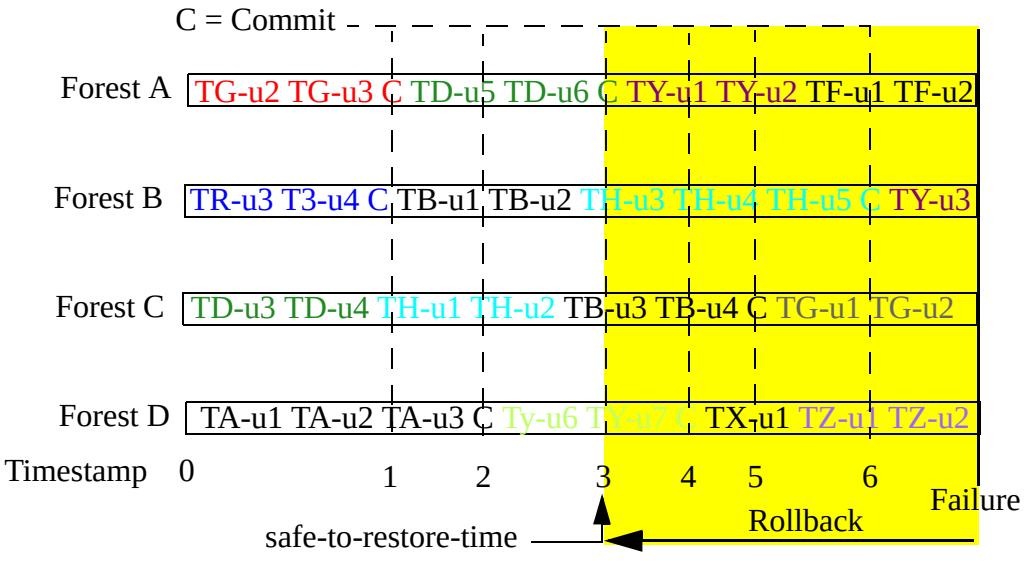
Note: If you are using XA distributed transaction processing, a restore to a point in time may revive some XA transactions that were prepared before the target restore time, and committed/aborted after that time. For details on how to identify XA transactions, see Heuristically Completing a MarkLogic Server Transaction in the *XCC Developer's Guide*

Note: 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.

20.7.1 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 `xdmp: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.

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.

20.7.2 Restoring to a Specific Timestamp

The following procedure describes how to restore a database to a specific timestamp using the XQuery API.

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.

20.7.3 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.

The following procedure describes how to restore to the state of some documents using the XQuery API.

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 chapter 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("2011-09-13T10:50:21.201832-07:00")))
```

6. Use `admin:database-set-merge-timestamp` function to set the merge timestamp back to the value you saved in Step 1.

20.8 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.

1. Log into the Admin Interface as a user with the `admin` role. Click the Databases link in the left menu of the Admin Interface.

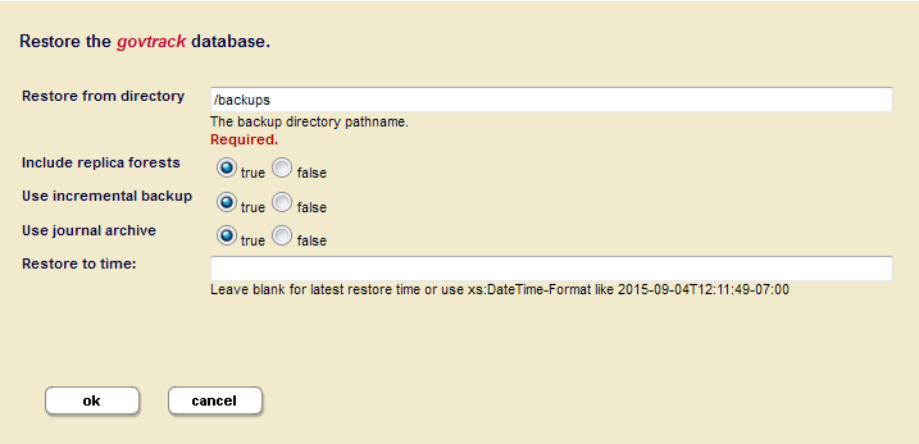
2. Click the database name for the database you want to restore, either on the tree menu or on the summary page. This database should have the same configuration settings (index options, fragmentation, range indexes) as the one that was backed up.

   Note: For journal archiving, you need either the the timestamp for the restore target or the current timestamp. This example uses a blank field (latest restore time/current timestamp) for the restore target.

3. Click the Backup/Restore tab. The Backup/Restore screen appears. 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 “Backup Directory Structure” on page 236.

4. 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*.



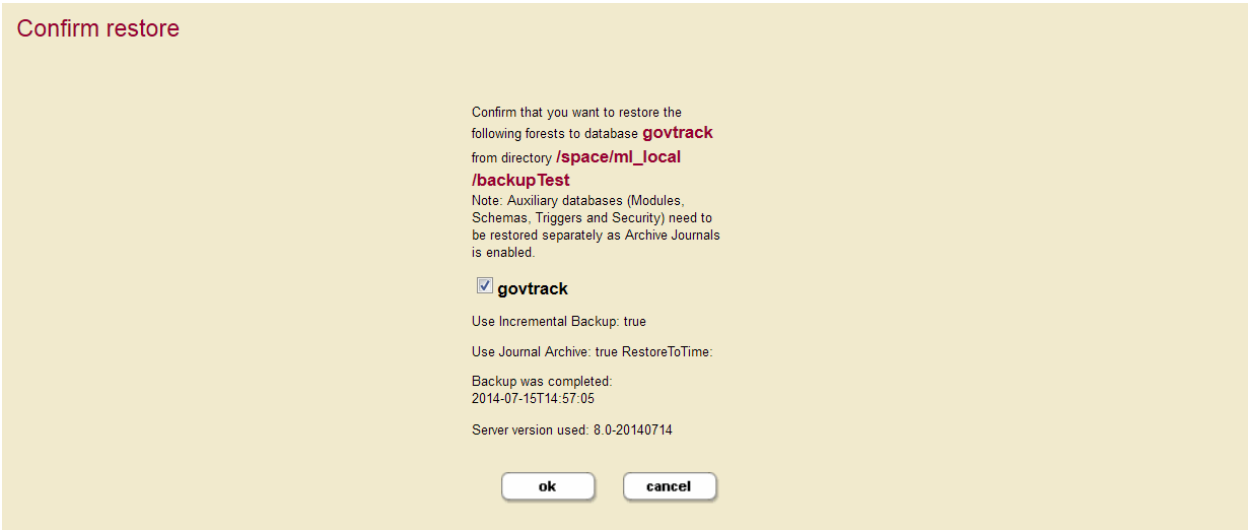
5. Set Use incremental backup to true. Set Journal Archiving to true. Leave the Restore to time blank or enter a time in `xs:DataTime-Format`.

Note: For Journal archive 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.

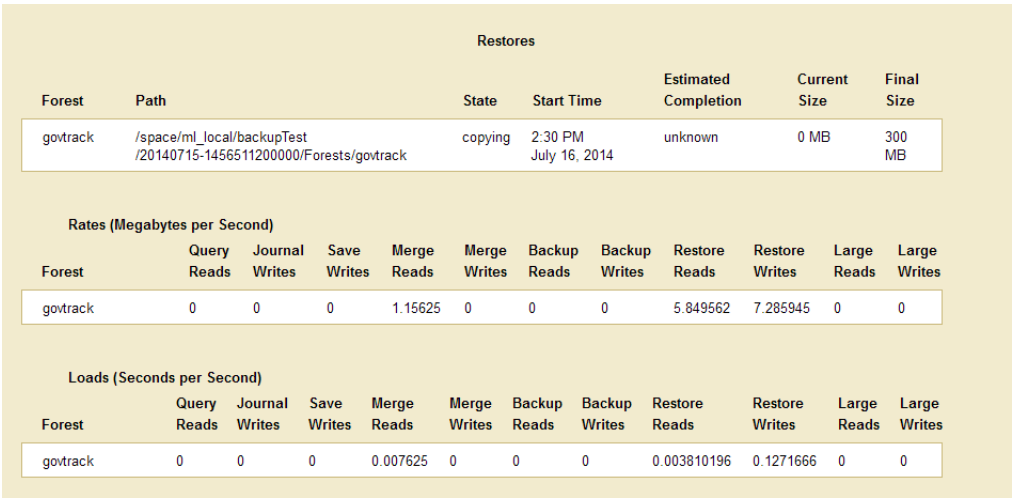
When restoring a backup with journal archiving enabled, be sure to change the merge timestamp from 0 to a non-zero value. Using zero for the merge timestamp will result in an error when restoring with journal archiving and restore-to-time set to zero. The merge timestamp must be set to a non-zero value.

6. Click OK to begin the restore process.

7. 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.



Restores

Forest	Path	State	Start Time	Estimated Completion	Current Size	Final Size
govtrack	/space/ml_local/backupTest /20140715-1456511200000/Forests/govtrack	copying	2:30 PM July 16, 2014	unknown	0 MB	300 MB

Rates (Megabytes per Second)

Forest	Query Reads	Journal Writes	Save Writes	Merge Reads	Merge Writes	Backup Reads	Backup Writes	Restore Reads	Restore Writes	Large Reads	Large Writes
govtrack	0	0	0	1.15625	0	0	0	5.849562	7.285945	0	0

Loads (Seconds per Second)

Forest	Query Reads	Journal Writes	Save Writes	Merge Reads	Merge Writes	Backup Reads	Backup Writes	Restore Reads	Restore Writes	Large Reads	Large Writes
govtrack	0	0	0	0.007625	0	0	0	0.003810196	0.1271666	0	0

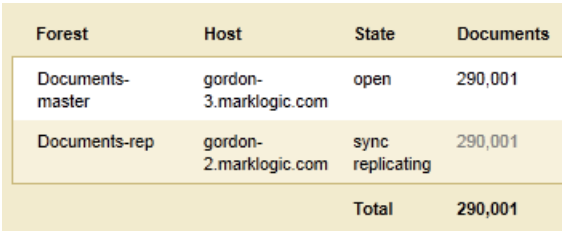
When the process is complete, the Restores table entry will disappear.

20.9 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 chapter in the *Scalability, Availability, and Failover Guide*.

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	290,001
Documents-rep	gordon-2.marklogic.com	sync replicating	290,001
		Total	290,001

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.

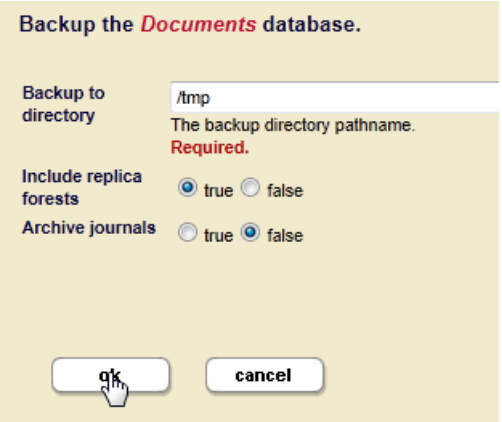


Forest	Host	State	Documents
Documents-master - This forest has an error, is disabled, or is ca			
Documents-rep	gordon-2.marklogic.com	open	290,001
		Total	290,001

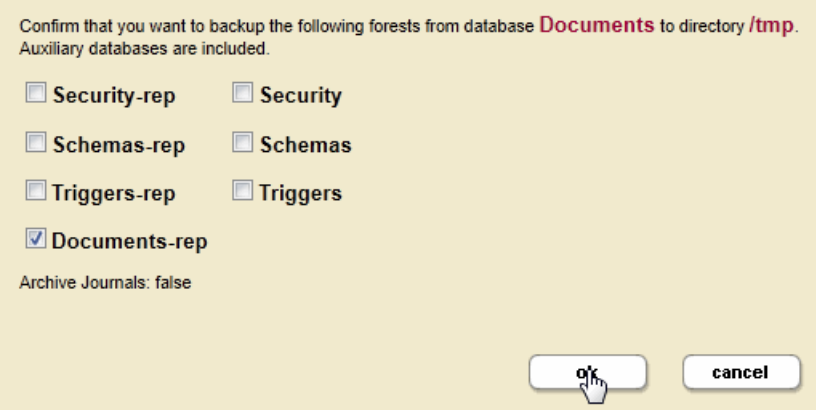
To back up the `Documents-rep` forest, do the following.

Note: Both the backup and restore procedures must be done on the host that contains the `Documents-rep` forest.

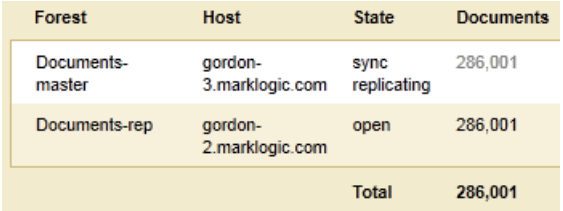
1. On the host machine that contains the `Documents-rep` forest, backup the `Documents` database. Leave Include Replica Forests set to true.



2. Select only the `Documents-rep` forest for backup.



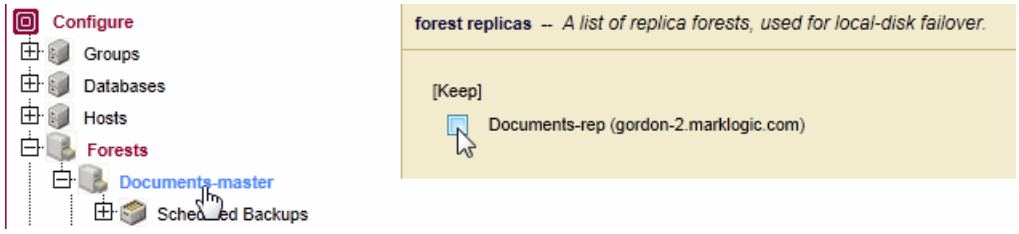
3. Once the host containing the `Documents-master` forest is restored, the `Documents-master` forest becomes the replica forest and receives replicated updates from the `Documents-rep` forest.



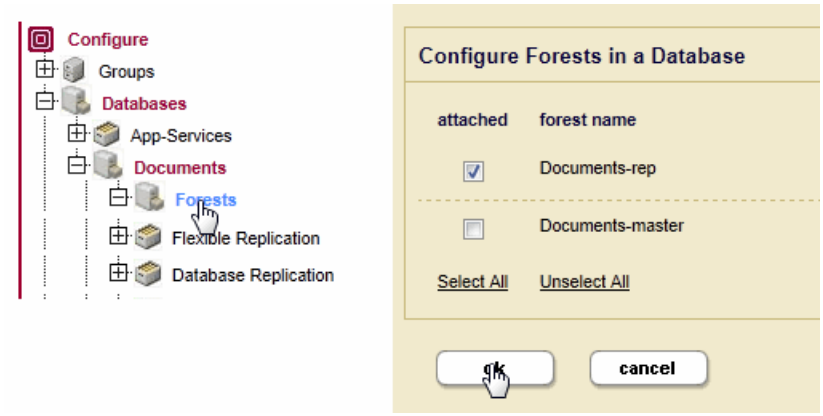
Forest	Host	State	Documents
Documents-master	gordon-3.marklogic.com	sync replicating	286,001
Documents-rep	gordon-2.marklogic.com	open	286,001
		Total	286,001

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.

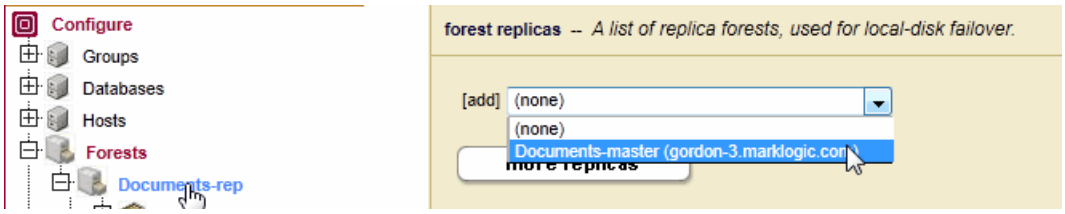
1. In the configuration page `Documents-master` forest, disable replication to `Documents-rep` forest.



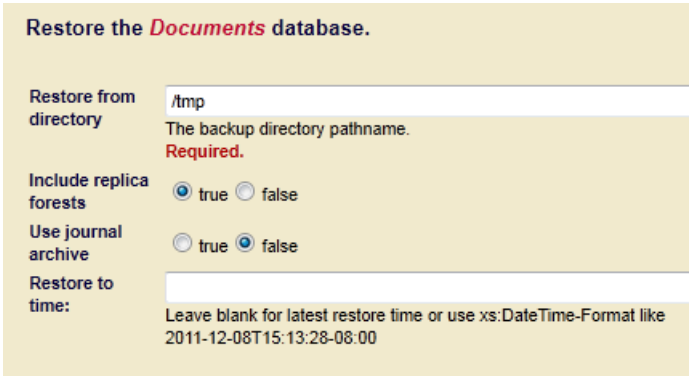
2. Navigate to the Forests configuration page for the `Documents` database. Unattach the `Documents-master` forest and attach the `Documents-rep` forest.



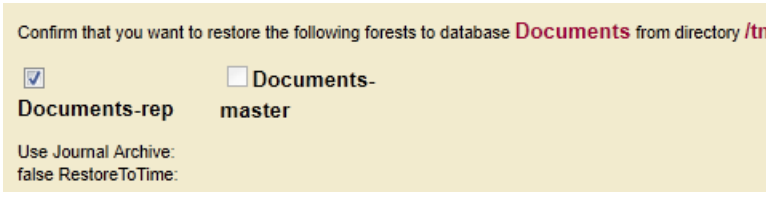
3. Navigate to the configuration page for the `Documents-rep` forest and select the `Documents-master` forest for local-disk failover.



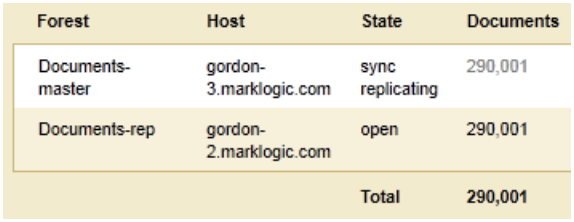
4. On the host containing the `Documents-rep` forest, confirm that the forest is in the `open` state and restore the `Documents` database from the backup taken after the failover.



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	290,001
Documents-rep	gordon-2.marklogic.com	open	290,001
		Total	290,001

21.0 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 the chapters "Groups" on page 30 and "Databases" on page 120 for more details on hosts as they relate to groups and databases.

A host is managed from both the Group and Hosts configuration screens. Use the following procedures to administer your hosts

- Adding a Host to a Cluster
- Changing the Group of the Host
- Shutting Down or Restarting a Host
- Clearing a Forest on a Host
- Deleting a Forest on a Host
- Leaving the Cluster
- Changing the License Key For a Host
- Rolling Back a Transaction

This chapter describes how to use the Admin Interface to manage hosts. For details on how to manage hosts programmatically, see 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*.

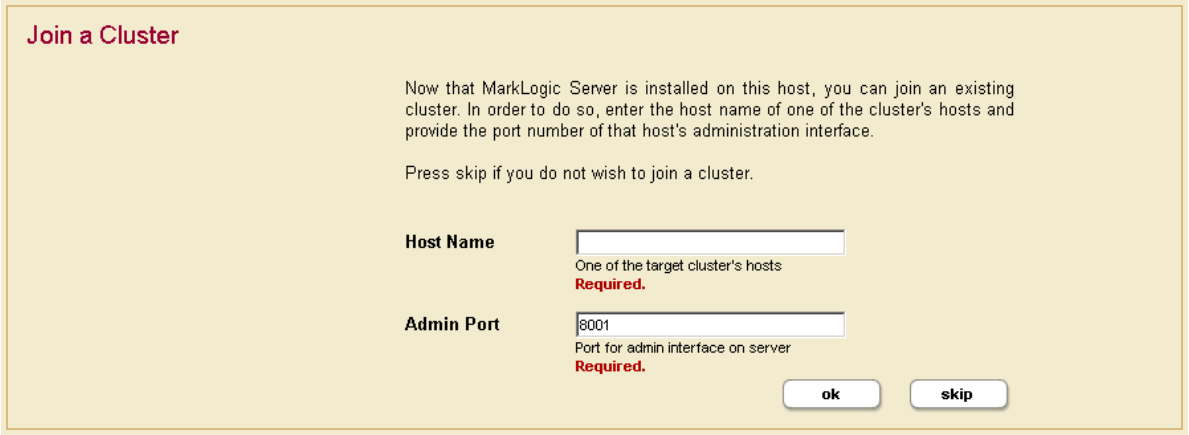
To add a host to a cluster, perform the following steps using the Admin Interface:

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, perform the following steps using the Admin Interface:

1. Click the Hosts icon in the left frame.
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, perform the following steps:

1. Click the Hosts icon on the left tree menu.
2. Click the name of the host you want to shut down or restart, either on the tree menu or the summary page.
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.

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, perform the following steps using the Admin Interface:

1. Click the Hosts icon on the left tree menu.
2. Click the name of the host which contains the forest you want to clear, either on the tree menu or the summary page.
3. Click the Forests icon 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, perform the following steps to delete a forest from a host.

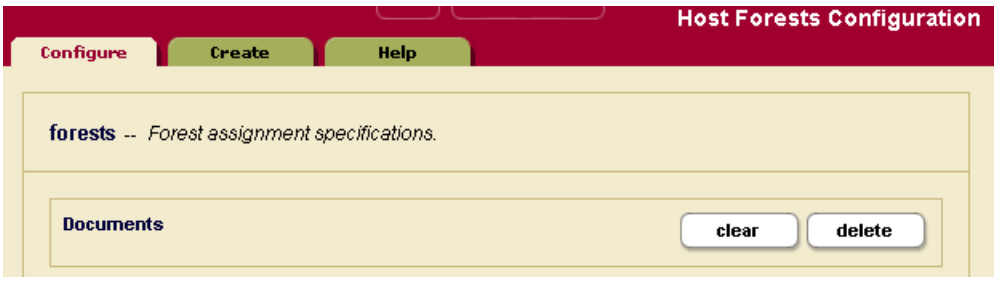
1. Click the Hosts icon on the left tree menu.

2. Click the name of the host which contains the forest you want to delete, either on the tree menu or the summary page.

3. Click the Forests icon under the selected host.

4. Click on 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

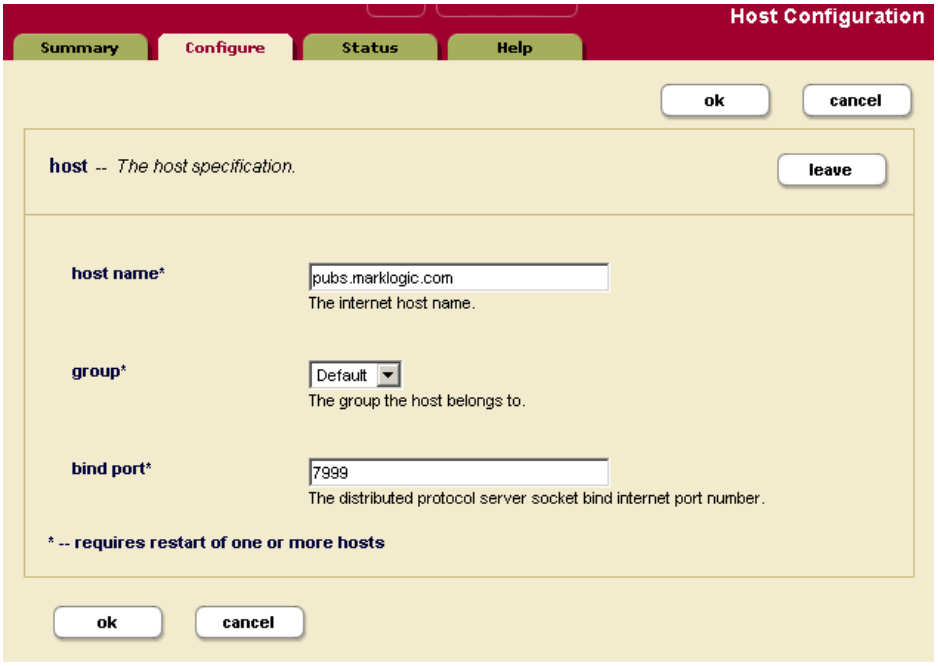
A host has to leave a cluster first to be moved to another cluster. Leaving a cluster is also a way to switch a host from a single host environment to a multi-host environment or vice versa. A host cannot leave a cluster if there are still forests assigned to it or if it has any foreign clusters associated 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.

Perform the following steps to make a host leave a cluster:

1. Access the Admin Interface from any host in the cluster.

2. Click on the Hosts icon in the left frame.

3. Click on the name of the host you want to remove from the cluster. The host configuration screen appears:



4. Click on the Leave button.

5. Click OK to confirm leaving the cluster.

6. The host restarts to load the new configuration.

7. Click OK to self-install initial databases and application servers.

8. You will be prompted to join a cluster.

9. To join another cluster, enter the name of one of the hosts in that cluster and click OK. Otherwise, click Skip.

10. Set up an admin user name and password if prompted.

11. Log in with the admin user name and password if prompted.

You should see the Admin Interface.

21.7 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, perform the following steps using the Admin Interface:

1. Click the Hosts icon on the left tree menu.

2. Click the name of the host in which contains you want to change the license key, either on the tree menu or the summary page. The Host Configuration page appears.

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.

21.8 Rolling Back a Transaction

Use Host Status page of the Admin Interface to rollback stalled or long-running transactions, discarding any updates made by the transaction. The Host Status page includes a list of transactions which have started but not yet completed.

> **Note:** To rollback the MarkLogic Server portion of a prepared XA transaction, see "Rolling Back a Prepared XA Transaction Branch" on page 290.

To rollback a transaction using the Admin Interface, complete the following steps:

1. Open the Admin Interface in your browser by navigating to `http://yourhost:8001`.

2. Click Hosts in the left tree menu. The tree expands to show available hosts.

3. Click the name of the target host in the left tree menu. The configuration view for the host appears.

4. Click the Status tab at the top of the page. The status view for the host appears.

5. 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.

6. Click `[rollback]` next to a transaction to initiate a transaction rollback. For example:



   A confirmation dialog appears.

7. Click OK to confirm the rollback. The host status page appears.

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”.

22.0 Forests

This section describes forests in the MarkLogic Server, and includes the following sections:

- Understanding Forests
- Creating a Forest
- Making a Forest Delete-Only
- Making a Forest Read-Only
- Attaching and Detaching Forests Using the Forest Summary Page
- Making Backups of a Forest
- Restoring a Forest
- Rolling Back a Forest to a Point In Time
- Merging a Forest
- Clearing a Forest
- Disabling a Forest
- Deleting a Forest from a Host
- Rolling Back a Prepared XA Transaction Branch

This chapter 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 comprised of XML, JSON, binary, and/or text fragments, stored in document order. 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 "Understanding and Controlling Database Merges" on page 169 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 "Making a Forest Delete-Only" on page 279 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 "Making a Forest Read-Only" on page 280 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 "Making a Forest Read-Only" on page 280 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, complete the following procedure:

1. Click the Forests icon in the left frame.

2. Click the Create tab at the top right. The Create Forest page displays:



3. Enter the name of your forest in the Forest Name textbox. Each forest name must be unique.

4. Select the host on which you want the forest to be created.

5. Enter the path to the Data Directory, which specifies where the forest data is stored. 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 the following 9 characters: `\ * ? / : < > | "` . Additionally, the name cannot begin or end with a space or a dot (.). 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.

   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*.

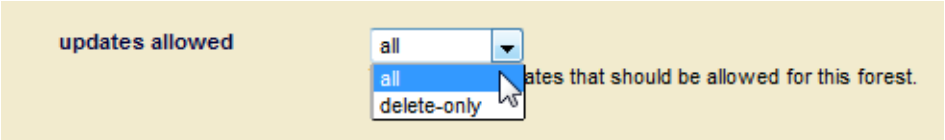
   The Forests directory is either a fully-qualified pathname or is relative to the Forests directory, set at installation time based on the directory in which MarkLogic Server is installed. The following table shows the default location Forest directory for each platform:

Platform	Program Directory
Microsoft Windows	`C:\Program Files\MarkLogic\Data\Forests`
Red Hat Linux	`/var/opt/MarkLogic/Forests`
Sun Solaris	`/var/opt/MARKlogic/Forests`
Mac OS X	`~/MarkLogic/Forests`

6. If you want to specify a different directory to store large objects (such as large binary documents), specify a Large Data Directory. If you do not specify a large data directory, the Data Directory is used. For details on binary file support, see Working With Binary Documents in the *Application Developer's Guide*.

7. 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*.

8. 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 “Making a Forest Delete-Only” on page 279 for details.



Note: The Read-Only update types described in “Making a Forest Read-Only” on page 280 can be set in the Configure page of an existing forest.

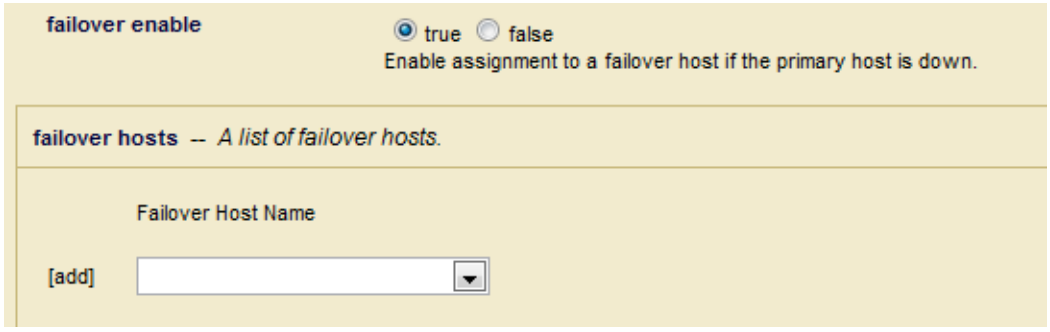
9. In the Availability field, select online to make the forest data available to tiered-storage or offline to make the data unavailable. For details on tiered storage, see “Tiered Storage” on page 203.

10. 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 “Database Rebalancing” on page 185.

11. 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 “Range Assignment Policy” on page 190.

12. 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*.

13. Select the Failover Host from the Failover Host Name drop down menu:



14. 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()) )
```

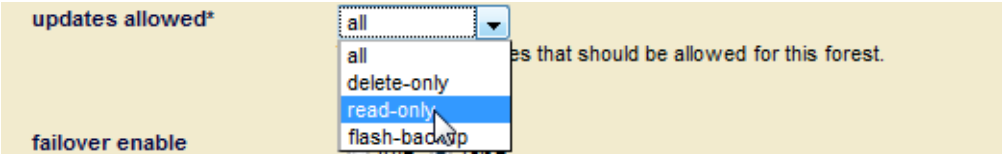
Note: You can only move a document from a delete-only forest to a forest that allows updates using an API that takes forest IDs, and then by explicitly setting the forest IDs to include one or more forests that allow updates. The node-level update built-in functions (`xdmp:node-replace`, `xdmp:node-insert-child`, 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 “Understanding and Controlling Database Merges” on page 169.

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.

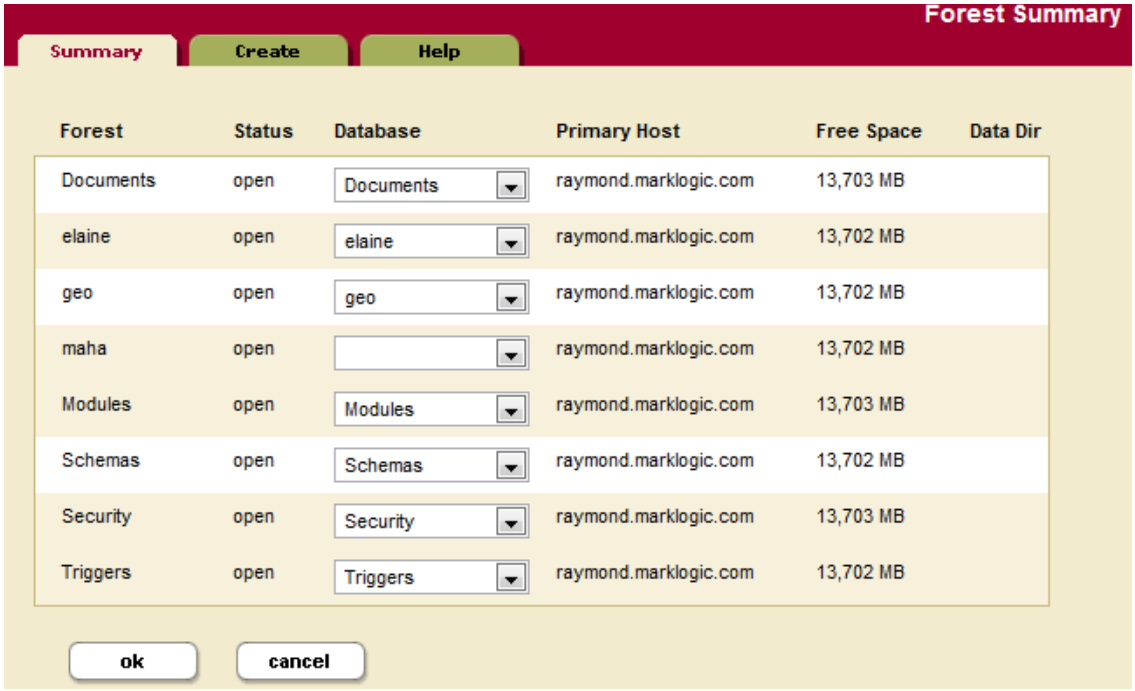
- If a forest is set to read-only or flash-backup, an insert, update, or delete transaction will either generate an exception (in the case of `read-only`) or retried later (in the case of `flash-backup`).
- You cannot clear, restore, or fully delete the forest. However, you can delete the forest configuration, as described in “Deleting a Forest from a Host” on page 289.
- 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 "Attaching and/or Detaching Forests to/from a Database" on page 135.

Perform the following steps using the Admin Interface to attach or detach one or more forests to or from a database:

1. Click the Forests icon on the left tree menu. The Forest Summary page appears.



2. For each forest whose database assignment you want to change, select the name of the new database assignment.

   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`.

3. After you have made your selections, click OK to save the forest assignment changes.

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 “Backing Up and Restoring a Database” on page 234.

> **Note:** Forest backups do not provide a journal archive feature, as described for database backups in “Backing Up and Restoring a Database” on page 234. However, you can manually invoke the `xdmp: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, and includes the following parts:

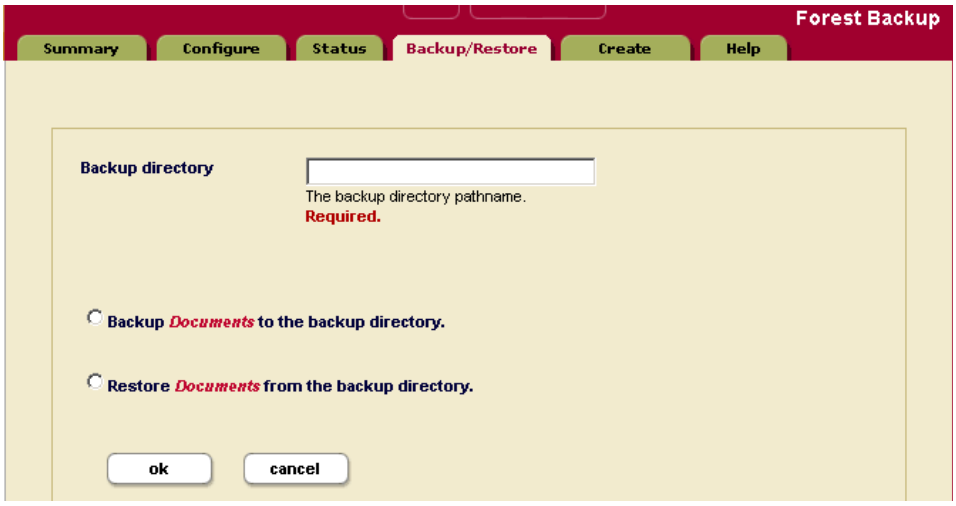
- Backing Up a Forest
- Scheduling a Forest Backup

22.6.1 Backing Up a Forest

To initiate a forest backup using the Admin Interface, complete the following procedure:

1. Click the Forests icon on the left tree menu.
2. Decide which forest to back up.
3. Click the icon for this forest name.

4. Click the Backup/Restore tab at the top right. The Forest Backup screen appears.



5. 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.

6. Select Backup.

7. Click OK.

8. 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, perform the following steps using the Admin Interface:

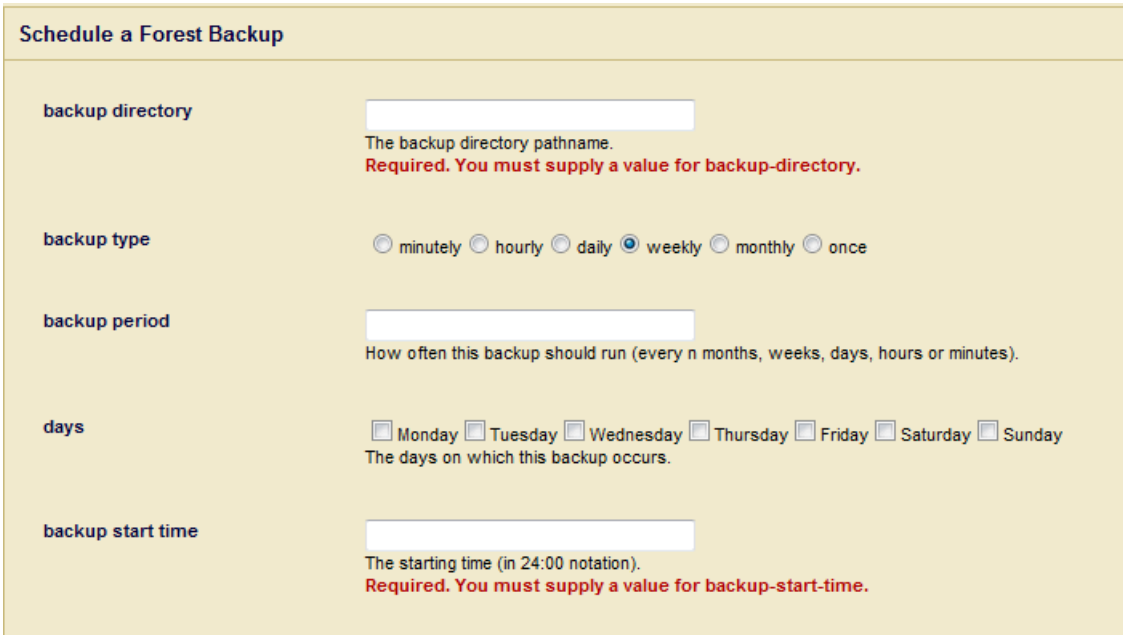
1. Click the Forests icon on the left tree menu.

2. Select the forest for which you want to schedule a backup, either from the tree menu or from the Forest Summary page. The Forest Configuration page appears.

3. Click the Scheduled Backup link in the tree menu for the forest. The Scheduled Backup Configuration page appears.

4. On the Scheduled Backup Configuration page, you can delete any existing scheduled backups if you no longer need them.

5. Click the Create tab. The Create Scheduled Backups page appears



6. 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. Choose a scheduled or one-time for the backup type:

    - For minutely, enter how many minutes between each backup.
    - For 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. 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 made earlier either using the Admin Interface. Backups are restored at the forest granularity only.

To restore a forest from a backup made previously, complete the following procedure:

1. Click the Forests icon on the left tree menu.

2. Decide which forest to restore.

3. Click the icon for this forest name.

4. Click the Backup/Restore tab on the top right.

5. Enter the name of the directory that contains the backup copy of the forest.

6. Select Restore.

7. Click OK.

    A confirmation message displays.

8. 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.

Warning When performing restores on the Windows platform, ensure that no users have the Forests or Data directories (or any subdirectories within them) open while the restore process is executing.

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 *MarkLogic XQuery and XSLT Function Reference*.

22.9 Merging a Forest

You can merge the forest data using the Admin Interface. As described in “Understanding and Controlling Database Merges” on page 169, 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, complete the following procedure:

1. Click the Forests icon on the left tree menu.
2. Decide which forest you want to merge.
3. Click the forest name, either on the tree menu or the summary page.

   The Forest Configuration page displays.
4. Click the Merge button on the Forest Configuration page.

   A confirmation message displays.
5. 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, complete the following procedure:

1. Click the Forests icon on the left tree menu.
2. Decide which forest you want to clear.
3. Click the forest name, either on the tree menu or the summary page.

   The Forest Configuration page displays.
4. Click the Clear button on the Forest Configuration page.

   A confirmation message displays.
5. 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, complete the following procedure:

1. Click the Forests icon on the left tree menu.
2. Decide which forest you want to disable.
3. Click the forest name, either on the tree menu or the summary page.

   The Forest Configuration page displays.
4. Click the Disable button on the Forest Configuration page.

   A confirmation message displays.
5. Confirm that you want to disable the forest by clicking Disable.

22.12 Deleting a Forest from a Host

You can use the Admin Interface to delete a forest. The are two levels of forest deletion:

- Delete configuration only, which removes the forest configuration information, but preserves the document data.
- Full Delete, which completely removes the document data and the configuration information for the forest.

  Note: The forest cannot be deleted if it is still attached to a database. Also, you can delete the configuration information on a Read-Only or Flash-Backup forest, but you cannot do a Full Delete on such forests.

To delete a forest, complete the following procedure:

1. Click the Forests icon on the left tree menu.
2. Decide which forest to delete.
3. Click the forest name, either on the tree menu or the summary page.

   The Forest Configuration page displays.
4. Click the Delete button on the Forest Configuration page.

   A confirmation message displays.
5. Select either Configuration Only to delete only the configuration information, or Full Delete to delete the configuration information and the document data.
6. 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 the *XCC Developer's Guide*.

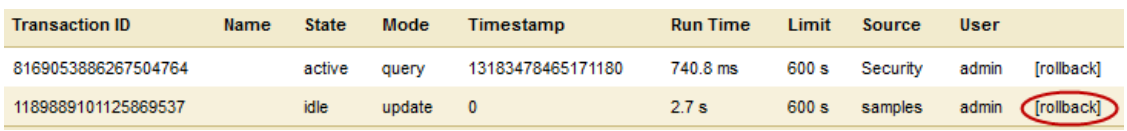
> **Note:** Heuristic completion bypasses the Transaction Manager and the Two Phase Commit process, so it can lead to data integrity problems. Use heuristic completion only as a last resort.

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 "Rolling Back a Transaction" on page 271.

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 using the Admin Interface, follow these steps:

1. Open the Admin Interface in your browser by navigating to `http://yourhost:8001`.
2. Click Forests on the left tree menu. The forest summary page appears.
3. Click the name of the coordinating forest. The Forest Status page appears.
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:



Transaction ID	Name	State	Mode	Timestamp	Run Time	Limit	Source	User	
8169053886267504764		active	query	13183478465171180	740.8 ms	600 s	Security	admin	[rollback]
1189889101125869537		idle	update	0	2.7 s	600 s	samples	admin	[rollback]

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 the *XCC Developer's Guide*.

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:



23.0 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. For background information on understanding the security model in MarkLogic Server, see *Understanding and Using Security Guide*. This section describes administration tasks related to security, and includes the following sections:

- Security Entities
- Users
- Roles
- Execute Privileges
- URI Privileges
- Amps
- Protected Collections
- Realm

This chapter 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*.

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 or remove documents from a 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” on page 78.

- Certificate Template

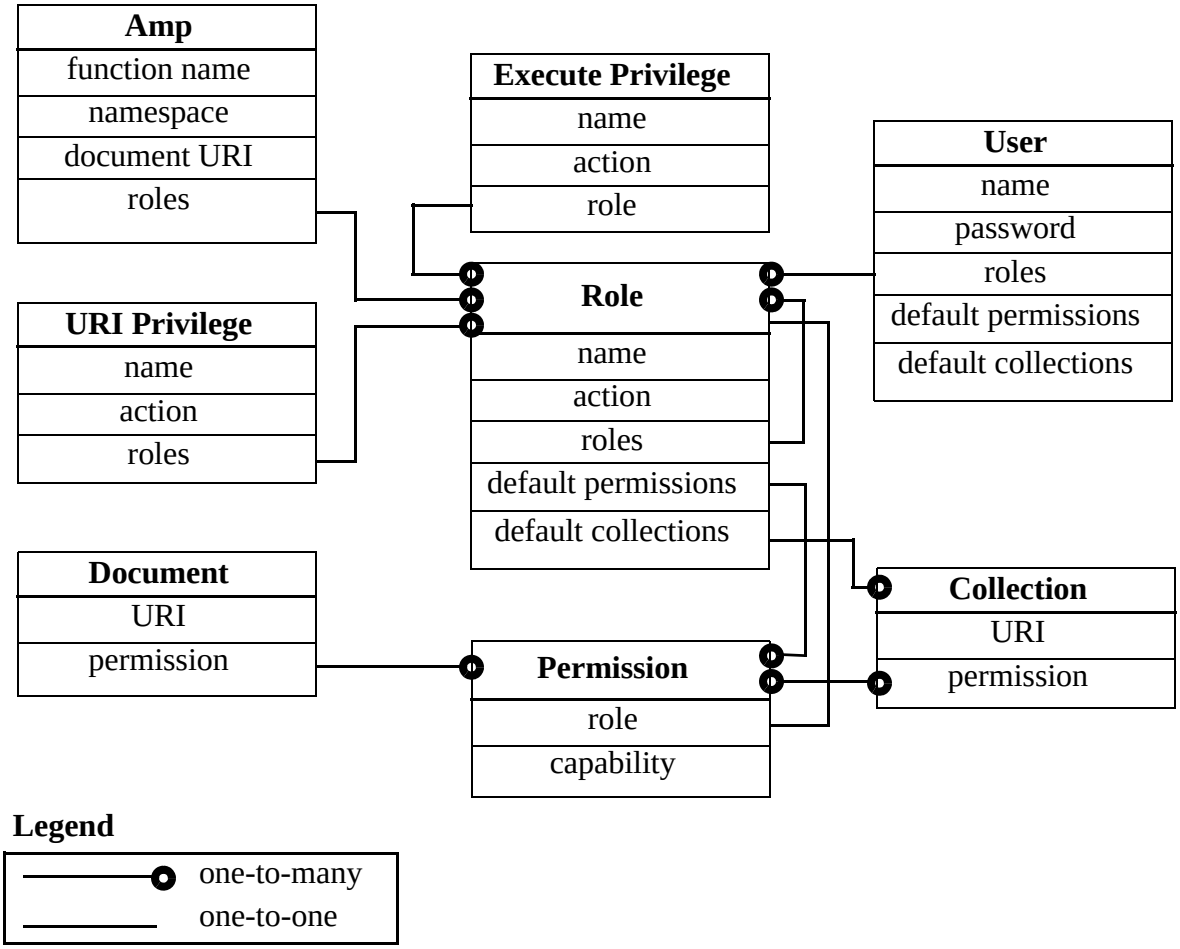
  A *certificate template* is a MarkLogic construct that is used to generate certificate requests for the various hosts in a cluster. A certificate template is used by the SSL (Secure Sockets Layer) security standard to provide encrypted protection between browsers and App Servers. The template defines 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” on page 78.

- External Authentication

  An *External Authentication Configuration Object* is used to configure MarkLogic Server for external authentication by LDAP or Kerberos. An external authentication configuration object specifies which authentication protocol and authorization scheme to use, along with any other parameters necessary for LDAP authentication. For details on external authentication with MarkLogic Server, see the External Authentication (LDAP and Kerberos) chapter in the *Understanding and Using Security Guide*.

- Security Entity Relationships

  The following diagram illustrates the relationships between the different entities in the MarkLogic Server security model.



The remaining sections of this chapter 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 *MarkLogic XQuery and XSLT Function Reference*.

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.

Use the following procedures to create, manage and maintain users:

- Creating a User
- Viewing a User Configuration
- Modifying a User Configuration
- Deleting a User

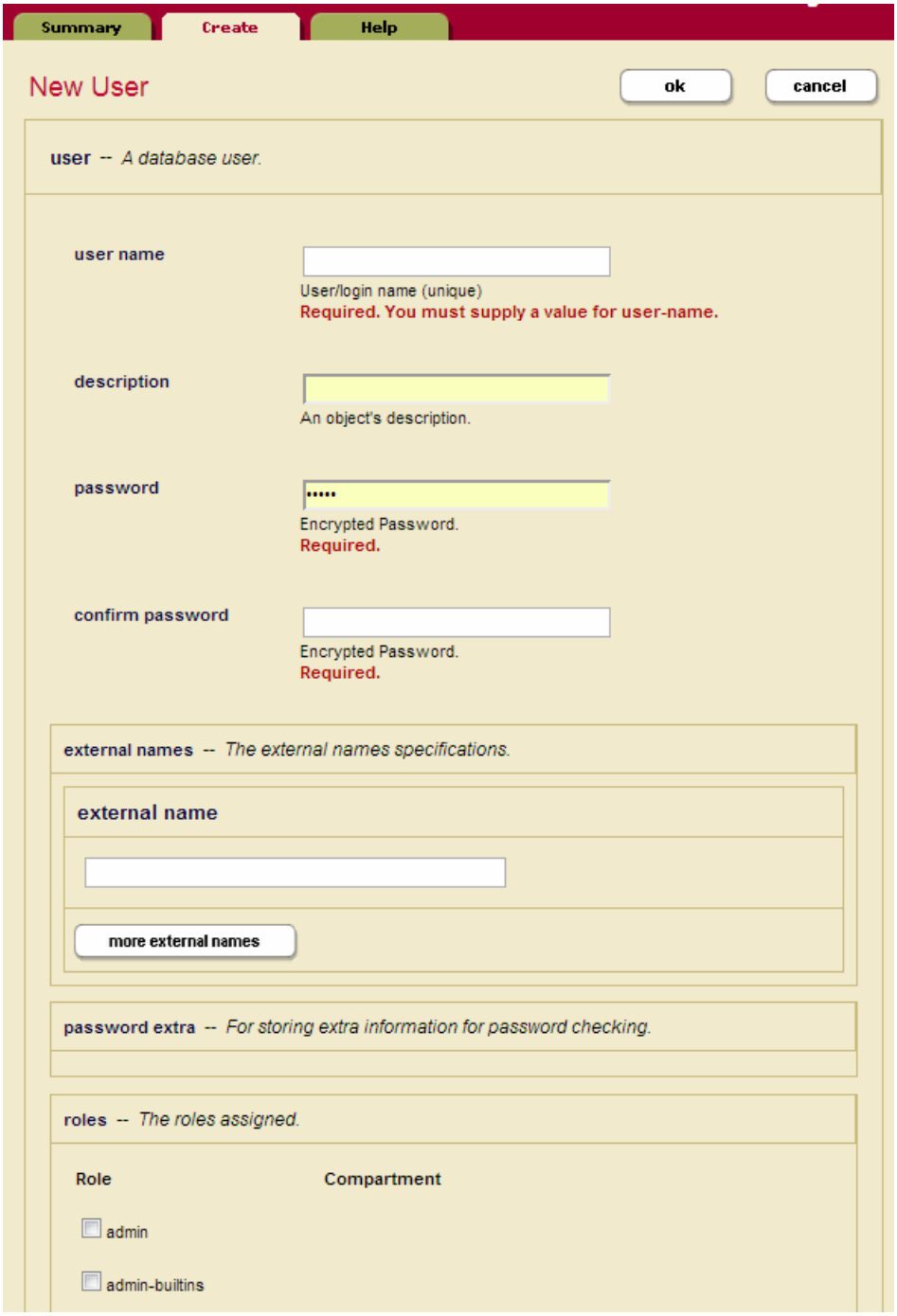
23.2.1 Creating a User

Follow these steps to create a user:

1. Click the Security icon in the left tree menu.

2. Click the Users icon.

3. Click the Create tab. The User Configuration page appears:



4. Enter a name for the user in the username field.

5. Enter the description for the user (optional).

6. Enter a password for the user.

7. Re-enter the password to confirm it.

8. 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 the External Authentication (LDAP and Kerberos) chapter in the *Understanding and Using Security Guide*.

9. Under the roles section, check the roles to assign the user.

10. Create default permissions for this user (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 user, you can do so on the next screen after you click OK.

11. Create default collections for this user (optional). 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.

12. 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

Perform the following steps to view a user's configuration:

1. Click the Security icon in the left tree menu.

2. Click the Users icon.

3. Locate the name of the user whose settings you want to view, either on the tree menu or on the summary page.

4. Click the name. The user configuration page appears where you can view the user's configuration:



23.2.3 Modifying a User Configuration

Perform the following steps to modify the configuration for a user:

1. For the user to which you want to modify, view that user's configuration as described in "Viewing a User Configuration" on page 298.

2. 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 *Understanding and Using Security Guide*.

3. 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

Perform the following steps to delete a user from the security database:

1. Click the Security icon in the left tree menu.

2. Click the Users icon.

3. Locate the user you want to delete, either on the tree menu or on the summary page.

4. Click the user name.

5. Click on the Delete button.

6. Click OK to confirm deleting the user.

The user is permanently deleted from the security database.

23.3 Roles

MarkLogic Server implements a role-base 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 *Understanding and Using Security Guide*.

Use the following procedures to create, manage and maintain roles:

- Creating a Role
- Viewing a Role
- Modifying a Role Configuration
- Deleting a Role

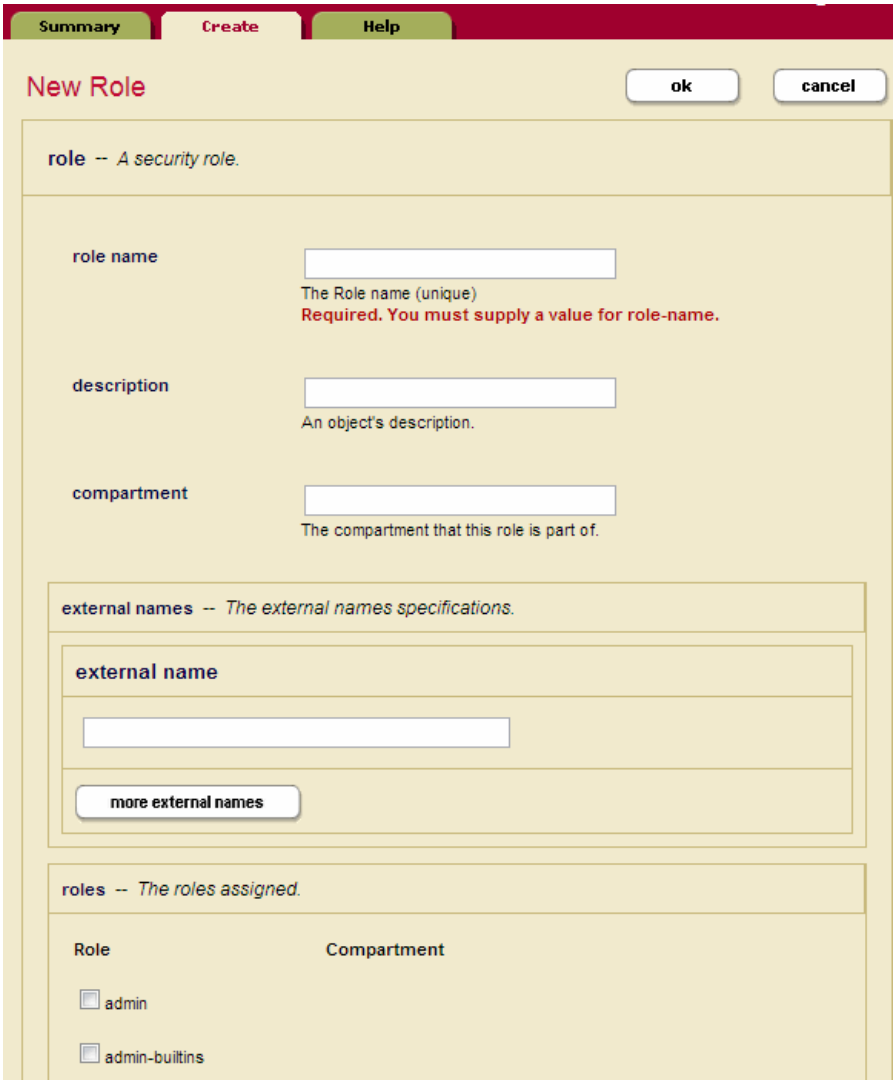
23.3.1 Creating a Role

Perform the following steps to create a role.

1. Click the Security icon in the left tree menu.

2. Click the Roles icon.

3. Click the Create tab. The Role Configuration page appears:



4. Type in a name for 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. 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, enter one or more group names in the External Names section. For details on external authorization, see the External Authentication (LDAP and Kerberos) chapter in the *Understanding and Using Security Guide*.

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 be associated with the role.

10. Under the URI privileges section, select from the available URI privileges to be associated 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.

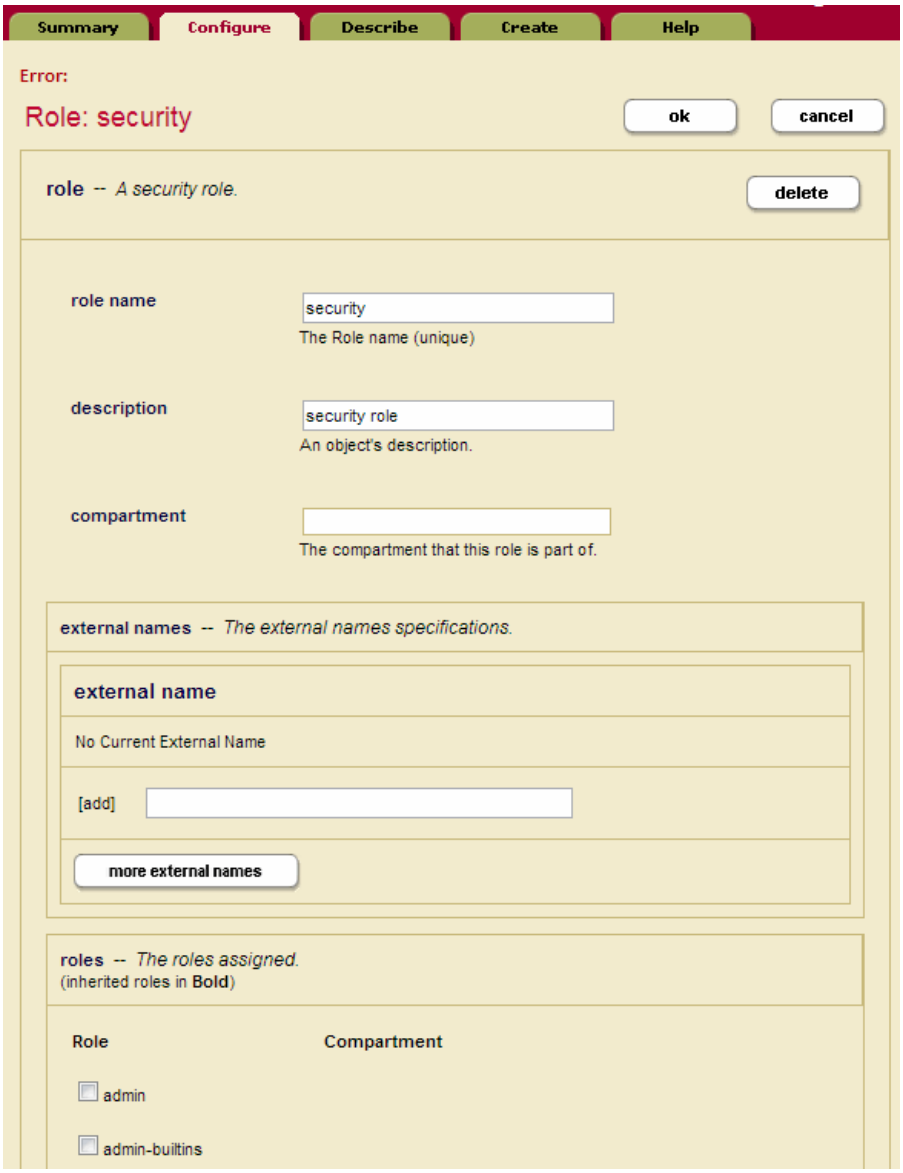
23.3.2 Viewing a Role

Perform the following steps to create a role.

1. Click the Security icon in the left tree menu.

2. Click the Roles icon.

3. Click the name of the role you want to view, either on the tree menu or on the summary page. The Role Configuration page appears.



View the configuration for the role.

23.3.3 Modifying a Role Configuration

Perform the following steps to modify a role configuration:

1. For the role to which you want to modify, view the role configuration as described in "Viewing a Role" on page 303.

2. 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 *Understanding and Using Security Guide*.

3. 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.

Perform the following steps to delete a role.

1. Click the Security icon in the left tree menu.

2. Click the Roles icon.

3. Click the name of the role you want to delete, either on the tree menu or on the summary page.

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

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.

Use the following procedures to create, manage and maintain execute privileges:

- Creating an Execute Privilege
- Viewing an Execute Privilege
- Modifying an Execute Privilege
- Deleting an Execute Privilege

23.4.1 Creating an Execute Privilege

Perform the following steps to create an execute privilege:

1. Click the Security icon in the left tree menu.
2. Click on the Execute Privileges icon.
3. Click the Create tab. The Execute Privilege Configuration page appears:



4. 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 execute privilege that gives a role the authorization to create a user.

5. 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.

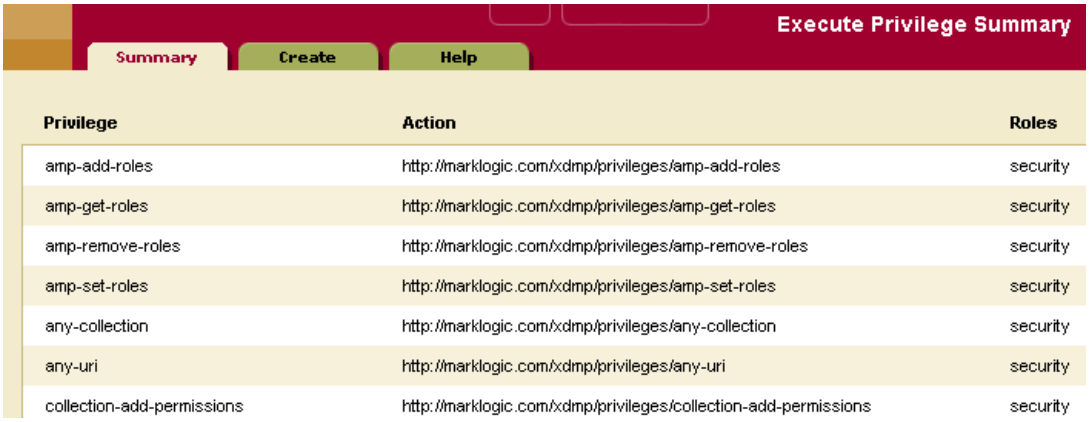
The execute privilege is now added to the security database. You can now use the `xdmp:security-assert()` function in your code to associate this privilege with a protected operation.

23.4.2 Viewing an Execute Privilege

Perform the following steps to view an execute privilege:

1. Click the Security icon in the left tree menu.

2. Click the Execute Privileges icon. The Execute Privileges Summary page appears:



3. Click on the name of the execute privilege that you want to view.

4. View the configuration for the execute privilege.

23.4.3 Modifying an Execute Privilege

Perform the following steps to modify an execute privilege:

1. For the privilege to which you want to modify, view the configuration as described in "Viewing an Execute Privilege" on page 306.

2. 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 *Understanding and Using Security Guide*.

3. Click OK to save the changes.

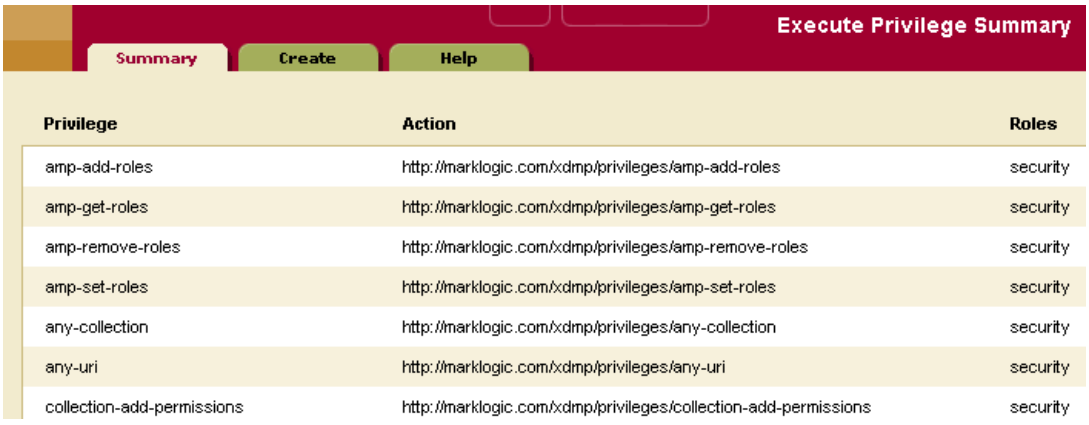
The new changes are in effect for all transactions beginning after the user changes are committed.

23.4.4 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. Follow these steps to delete an execute privilege:

1. Click the Security icon in the left tree menu.

2. Click the Execute Privileges icon. The Execute Privileges Summary page appears:



3. Click the name of the execute privilege that you want to delete.

4. On the Execute Privileges page for the given privilege, click the Delete button.

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 following procedures to create, manage and maintain URI privileges:

- Creating a URI Privilege
- Viewing a URI Privilege
- Modifying a URI Privilege
- Deleting a URI Privilege

23.5.1 Creating a URI Privilege

Perform the following steps to create a URI privilege:

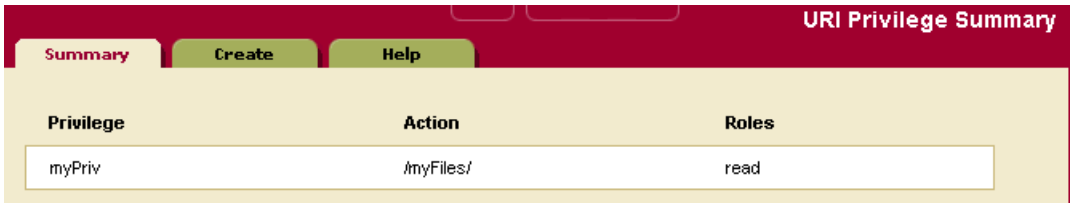
1. Click the Security icon in the left tree menu.
2. Click the URI Privileges icon.
3. Click on the Create tab. The URI Privilege Configuration page appears:
4. Enter the name of the URI privilege. Use a name that is descriptive of the base URI to be protected. For example, to restrict the creation of documents under a base URI reserved for the accounting group, you might use the name “accounting_files”.
5. In the action field, enter the base URI to be protected. 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. Under 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

Perform the following steps to view a URI privilege:

1. Click the Security icon in the left tree menu.
2. Click the URI Privileges icon. The URI Privileges Summary Page appears:



3. Click the name of the URI privilege you want to view.
4. View the URI privilege.

23.5.3 Modifying a URI Privilege

Perform the following steps to modify an execute privilege:

1. For the privilege to which you want to modify, view the configuration as described in "Viewing a URI Privilege" on page 309.

2. 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 *Understanding and Using Security Guide*.

3. Click OK to save the changes.

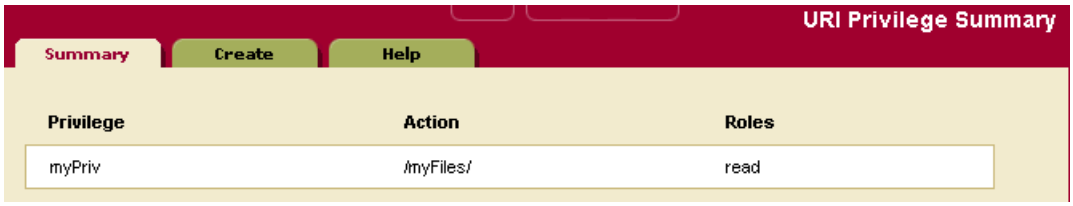
The new changes are in effect for all transactions beginning after the user changes are committed.

23.5.4 Deleting a URI Privilege

You can delete a URI privilege from the security database. Perform the following steps to delete a URI privilege:

1. Click the Security icon in the left tree menu.

2. Click the URI Privileges icon. The URI Privileges Summary Page appears:



3. On the URI Privileges page for the given privilege, click the Delete button.

4. 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 the `document-count()` function 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.

For more details about amps, see *Understanding and Using Security Guide*. For examples of amps, look at one of the amps created during installation. To view an amp, follow the instructions in the section "Viewing an Amp" on page 313.

Use the following procedures to create, manage and maintain amps:

- Creating an Amp
- Viewing an Amp
- Modifying an Amp
- Deleting an Amp

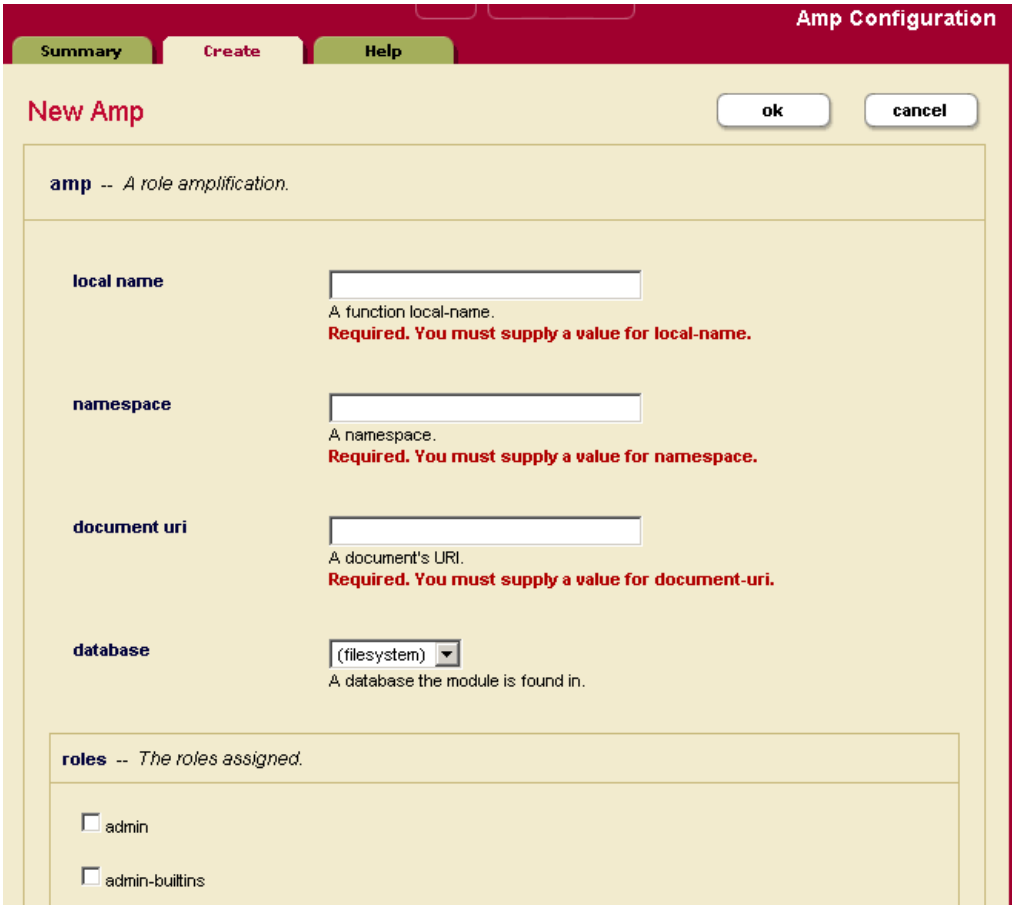
23.6.1 Creating an Amp

To create an amp, Perform the following steps:

1. Click the Security icon in the left tree menu.

2. Click the Amps icon.

3. Click on the Create tab. The Amp Configuration page appears:



4. Enter the database in which 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).

5. Enter the local name of the function (without parenthesis) 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. Under the roles section, select the additional roles that will be given to the user while the user is executing the function.

9. Click OK.

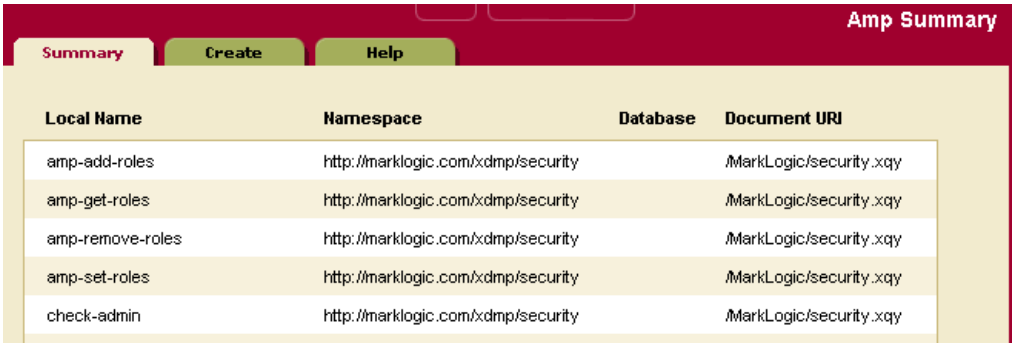
The amp is now added to the security database.

23.6.2 Viewing an Amp

Perform the following steps to view an amp:

1. Click the Security icon in the left tree menu.

2. Click the Amps icon. The Amps Summary page appears:



3. Click on the name of the amp you want to view.

4. View the amp.

23.6.3 Modifying an Amp

Perform the following steps to modify an amp:

1. For the amp to which you want to modify, view the configuration as described in "Viewing an Amp" on page 313.

2. 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 *Understanding and Using Security Guide*.

3. Click OK to save the changes.

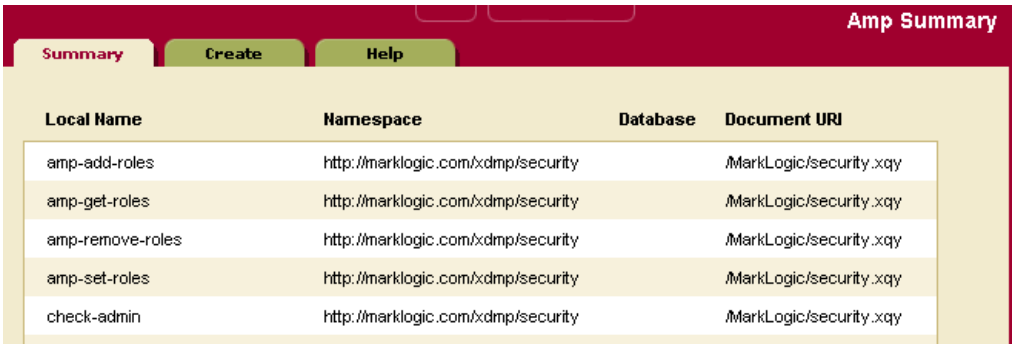
The new changes are in effect for all transactions beginning after the user changes are committed.

23.6.4 Deleting an Amp

You can delete an amp from the security database. Perform the following steps to delete an amp:

1. Click the Security icon in the left tree menu.

2. Click the Amps icon. The Amps Summary page appears:



3. Click on the name of the amp you want to delete.

4. On the Amp page for the given amp, click the Delete button.

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. However, an associated collection object is not created and stored in the security database unless it is protected. A collection created through the Admin Interface is a protected collection and is stored in the security database.

Read, Insert, Update, and Execute capabilities apply for permissions on a collection. A user needs to have read permissions for both the collection and the documents to be able to see the documents in the collection by collection. A user needs to have Update permissions for both the collection and the document to be able to add documents to or remove documents from a collection.

Use the following procedures to create, manage and maintain collections:

- Creating a Protected Collection
- Viewing a Protected Collection
- Removing a Permission from a Protected Collection
- Deleting a Protected Collection

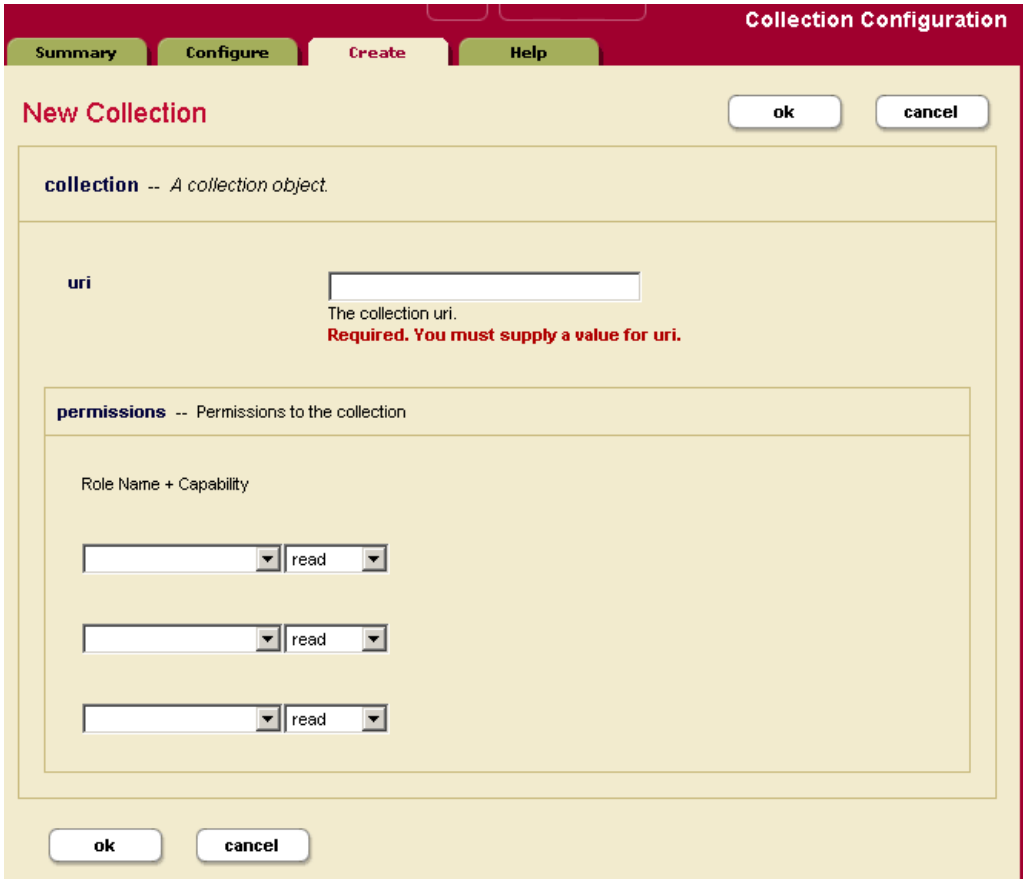
23.7.1 Creating a Protected Collection

Perform the following steps to create a protected collection:

1. Click the Security icon in the left tree menu.

2. Click the Collections icon.

3. Click the Create tab, The Collection Configuration page appears:



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 Read, Insert, Update, or Execute capability for the role. If you want to add more than 3 permissions to the role, you can do so from the next screen after you click OK.

6. Click OK.

The protected collection is added to the database.

23.7.2 Viewing a Protected Collection

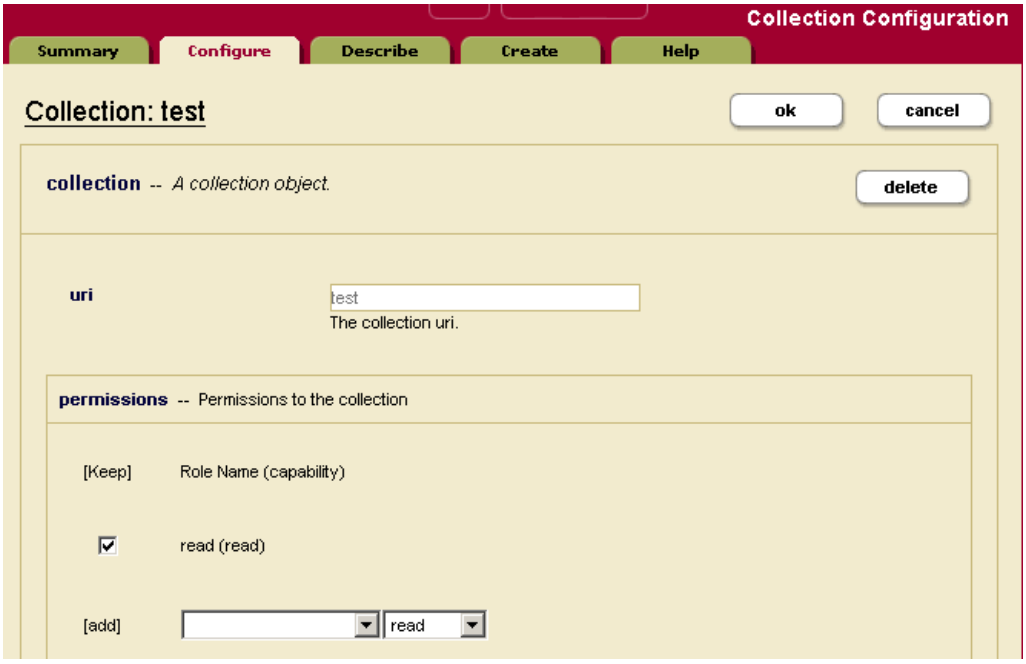
Perform the following steps to view a protected collection:

1. Click the Security icon in the left tree menu.
2. Click the Collections icon. The Collection Summary page appears.
3. Click the name of the collection you want to view, either on the tree menu or on the summary page. The Collection Configuration page appears.
4. View the collection.

23.7.3 Removing a Permission from a Protected Collection

Perform the following steps to remove a permission from a protected collection:

1. Click the Security icon in the left tree menu.
2. Click the Collections icon. The Collection Summary page appears.
3. Click the name of the collection from which you want to remove a permission, either on the tree menu or on the summary page. The Collection Configuration page appears.



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.

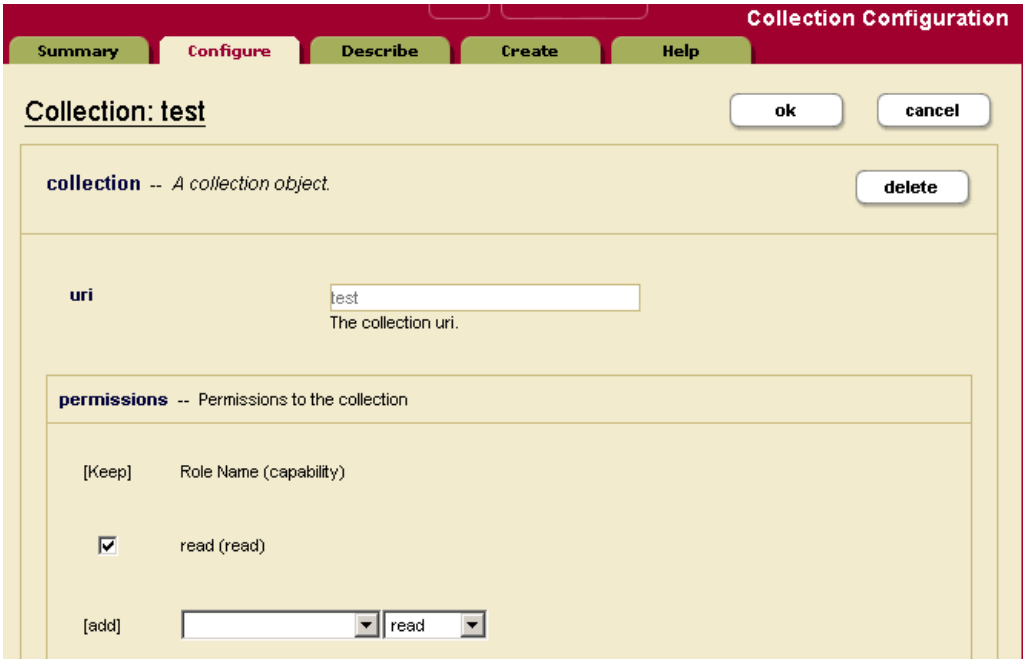
23.7.4 Deleting a Protected Collection

Perform the following steps to remove delete a protected collection:

1. Click the Security icon in the left tree menu.

2. Click the Collections icon.

3. Click the name of the collection you want to delete, either on the tree menu or on the summary page. The Collection Configuration page appears.



4. Click on the Delete button 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

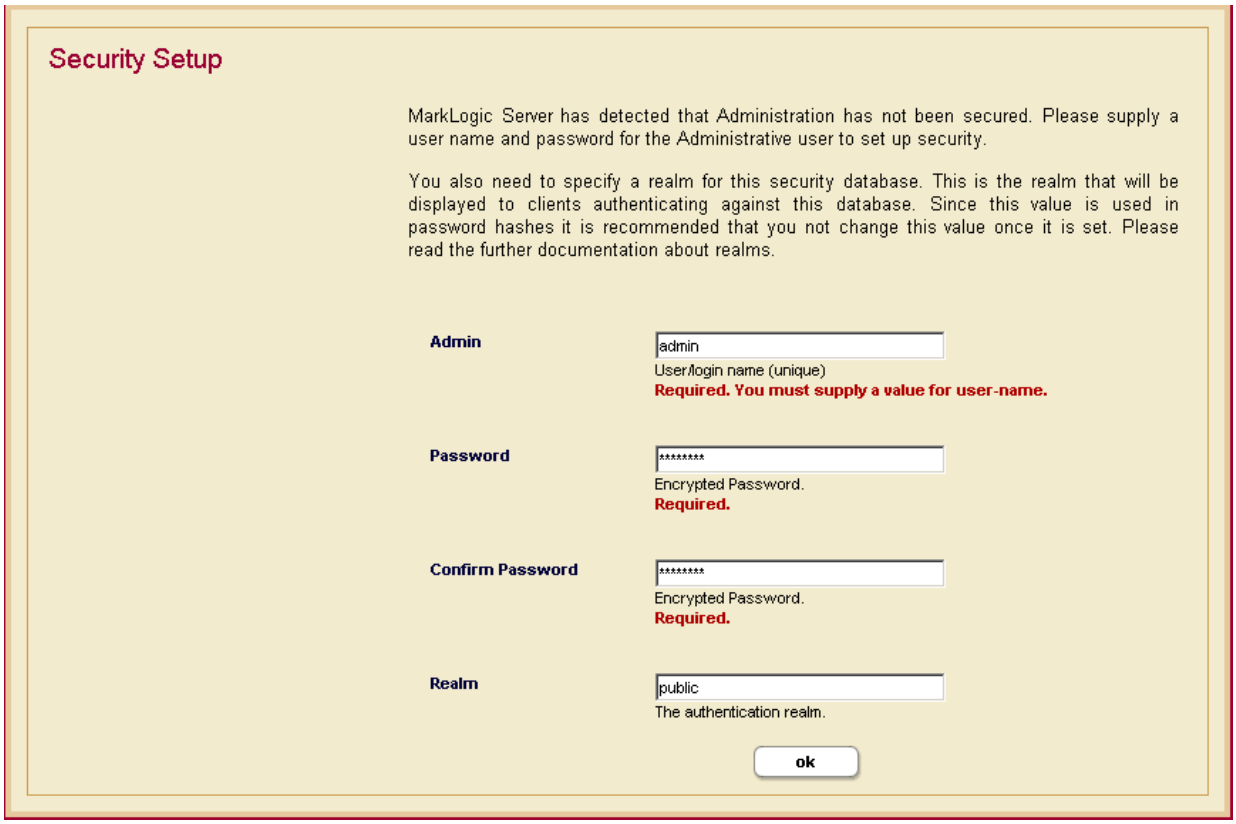
A Certificate Template contains the identification information associated with an SSL certificate. See Configuring SSL on App Servers for details.

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, perform the following steps (note the warning above, however):

1. Click Security in the left tree menu.

2. Click the Configure tab. The Security Configuration page appears.



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).

24.0 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 chapter describes those parameters and includes the following sections:

- Text Indexes
- Phrasing and Element-Word-Query Boundary Control
- Query Behavior with Reindex Settings Enabled and Disabled

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 “Range Indexes and Lexicons” on page 340. You can also create named fields which can explicitly include or exclude specified elements. For details on fields, see “Fields Database Settings” on page 152.

This section describes the text indexes in MarkLogic Server and includes the following subsections:

- Understanding the Text Index Settings
- Viewing Text Index Configuration
- Configuring Text Indexes

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 `xml:lang` 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.

Contains SAP BusinessObjects Text Analysis XI from SAP AG. Copyright © 1996-2011. All rights reserved.

Index	Default Setting	Description
`stemmed searches`	Basic (index is built, each word stems to a single stem)	Enables searches to return relevance ranked results by matching word stems. A word *stem* is the word that has the same meaning as the specified word, and other words can also have that same stem; therefore, stemmed searches will return more matching results than the exact words specified in the query. A stemmed search for a word finds the exact same terms as well as terms that derive from the same meaning and part of speech as the search term. For example, a stemmed search for `run` returns results containing `run`, `running`, `runs`, and `ran`. For details on stemming, see the chapter Understanding and Using Stemmed Searches in the *Search Developer's Guide*. There are three types of stemming: basic (one stem per word), advanced (one or more stems per word), and decompounding (advanced plus smaller component words of large compound words). Without either this index or the word searches index, MarkLogic Server is unable to perform relevance ranking and will refuse to execute any `cts:word-query()`-related built-in function. 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). 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.
`word searches` (unstemmed)	Off (index is not built)	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 `cts:word-query()`-related function. For many applications, keeping this word search index off and the stemmed search index on is sufficient to return the desired results for queries. 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.
`word positions`	Off (index is not built)	Speeds up the performance of proximity queries that use the `cts:near-query` function and of multi-word phrase searches. 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 `fast phrase searches`, as they have some overlapping functionality.

Index	Default Setting	Description
`fast phrase searches`	On (index is built)	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. 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.
`fast case sensitive searches`	On (index is built)	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. 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.
`fast reverse searches`	Off (index is not built)	Speeds up reverse query searches by indexing stored queries. Turn this option on to speed up searches that use `cts:reverse-query`.
`fast diacritic sensitive searches`	On (index is built)	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.
`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 `cts:near-query` 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.

Index	Default Setting	Description
`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 `cts:element-value-query` 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 `cts:element-attribute-value-query` function and speeds up `cts:element-query` searches that us 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 `cts:field-value-query` function. Without this index or the corresponding index on the field definition, queries that use `cts:field-value-query` 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 `cts:field-value-query` 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, `abc*`). Turn this index on to speed up wildcard searches that match a trailing wildcard. The `trailing wildcard search` index uses roughly the same space as the `three character searches` 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.

Index	Default Setting	Description
`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 `cts:near-query` 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.
`three character searches`	Off (index is not built)	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 `three character search` index with a codepoint collation word lexicon. For details on wildcard characters, see Understanding and Using Wildcard Searches in the *Application Developer's Guide*. When character indexing is turned on, performance is also improved for `fn:contains()`, `fn:matches()`, `fn:starts-with()` and `fn:ends-with()` for most query expressions. 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 two character searches and/or one character searches.
`three character word positions`	Off (index is not built)	Speeds up the performance of proximity queries that use three-character word searches, such as queries that use the `cts:near-query` function and multi-word phrase searches that contain one or more wildcard terms. Turn this index on if you are using wildcard searches and proximity queries together in the same search.

Index	Default Setting	Description
`two character searches`	Off (index is not built)	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 *Application Developer's Guide*. When character indexing is turned on in the database, the system also delivers higher performance for `fn:contains()`, `fn:matches()`, `fn:starts-with()` and `fn:ends-with()` for most query expressions. 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 `three character searches` and a word lexicon.
`one character searches`	Off (index is not built)	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 *Application Developer's Guide*. When character indexing is turned on in the database, the system also delivers higher performance for `fn:contains()`, `fn:matches()`, `fn:starts-with()` and `fn:ends-with()` for most query expressions. 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 `three character searches` and a word lexicon.
`fast element character searches`	Off (index is not built)	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 *Application Developer's Guide*.

Index	Default Setting	Description
`word lexicons`	Off (index is not built)	Maintains a lexicon of all of the words in a database, with uniqueness determined by a specified collation. For details on lexicons, see "Range Indexes and Lexicons" on page 340 and the chapter on lexicons in the *Application Developer's Guide*. For details on collations, see the Language Support in MarkLogic Server chapter in the *Search Developer's Guide*. 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 `three character search` index, improves wildcard index resolution and speeds up wildcard searches. If you have `three character search` and a word lexicon enabled for a database, then there is no need for either the `one character` or `two character search` 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 chapter on wildcard searches in the *Application Developer's Guide*.
`uri lexicon`	Off (index is not built)	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 `cts:uris` or `cts:uri-match` APIs.
`collection lexicon`	Off (index is not built)	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 `cts:collections` or `cts:collection-match` APIs.

24.1.2 Viewing Text Index Configuration

To view text index configuration for a particular database, complete the following procedure:

1. Click on the Databases icon on the left tree menu.

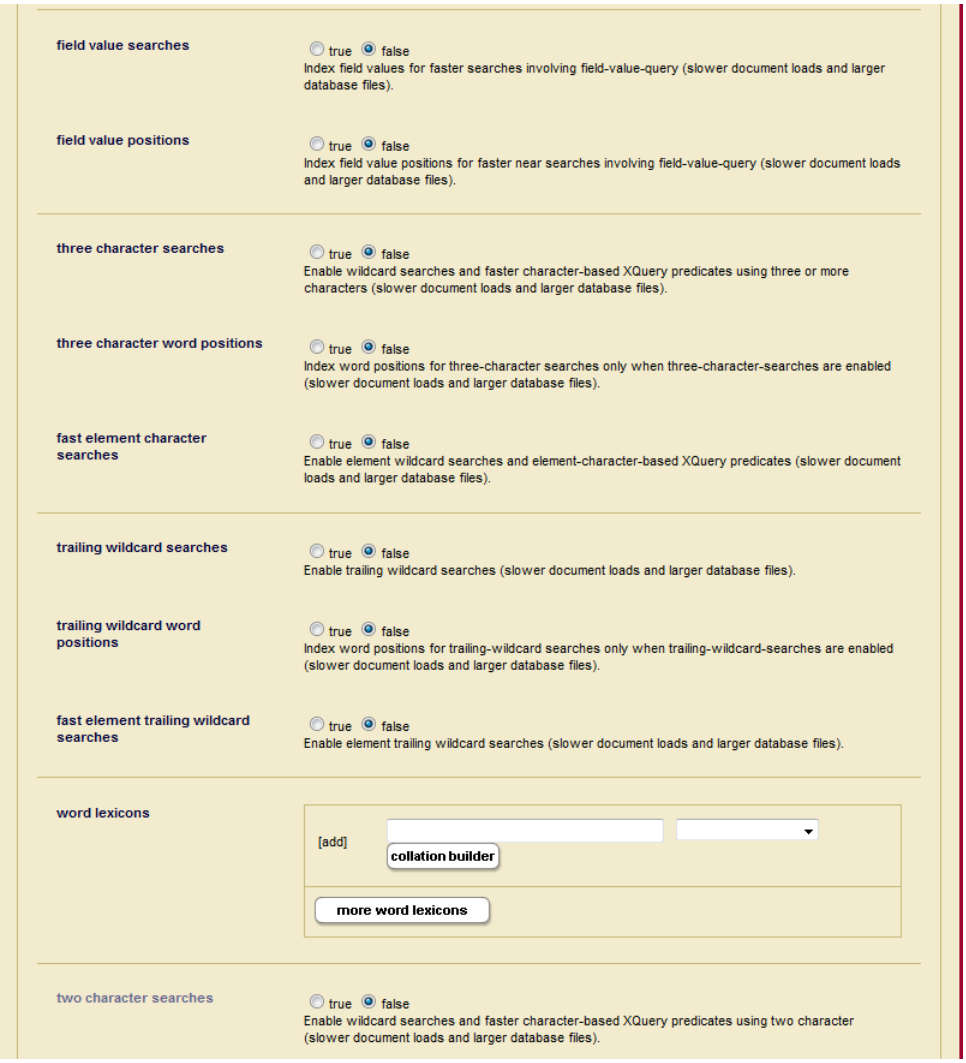
2. Locate the database for which you want to view text index configuration settings, either in the tree menu or in the Database Summary table.

3. Click the name of the database for which you want to view the settings.

4. Scroll down until the text index settings are visible. The following screen shots show the default configuration of text indexing for a database:







24.1.3 Configuring Text Indexes

To configure text indexes for a particular database, complete the following procedure:

1. Click on the Databases icon on the left tree menu.

2. Locate the database for which you want to view text index configuration 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. Scroll down until the text indexing controls are visible.

5. Configure the text indexes for this database by selecting the appropriate radio buttons for each index type.

   Click on the `true` radio button for a particular text index type if you want that index to be maintained. Click on 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.

6. Leave the rest of the parameters unchanged.

7. Scroll to the top or bottom of the right frame 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. This section has the following parts:

- Phrasing Control
- Element Word Query Throughs
- Procedures

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 the following 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
`Phrase-through`	Elements that should not create phrase boundaries (as in the example above, `italic` should be specified as a phrase-through element).
`Phrase-around`	Elements whose content should be completely ignored in the context of the current phrase (as in the example above, `footnote` should be specified as a phrase-around 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 the following 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 `<b>`, it does not satisfy the `cts:element-word-query`.

If you add an element-word-query-through for the element `<b>`, however, then the `cts:element-word-query` on `<a>` searching for the word `hello` returns a match. The element-word-query-through on `<b>` causes the text node children of `<b>` 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 is 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

Use the following procedures to configure phrase controls for a particular database:

- Viewing Phrasing and Element-Word-Query Settings
- Configuring Phrasing and Element-Word-Query Settings
- Deleting a Phrasing or Element-Word-Query Setting

24.2.3.1 Viewing Phrasing and Element-Word-Query Settings

To view element-word-query-through, phrase-through, and phrase-around settings for a particular database, complete the following procedure in the Admin Interface:

1. Click on the Databases icon on the left tree menu.

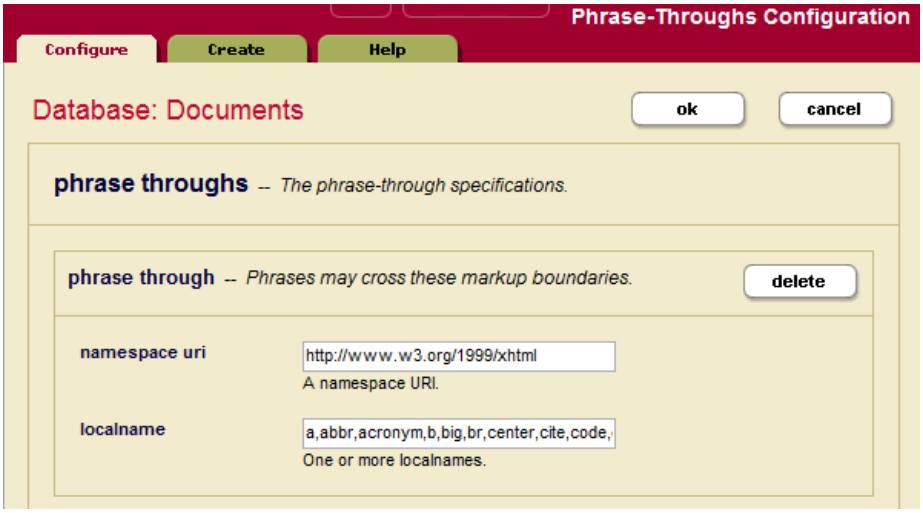
2. Locate the database for which you want to view element-word-query-through, phrase-through, or phrase-around settings, either in the tree menu or in the Database Summary table.

3. Click the name of the database for which you want to view the settings.

4. Click the Element-Word-Query-Throughs, Phrase-Throughs, or Phrase-Arounds icon, depending on which one you want to view.

5. 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>`, `<b>`, `<big>`, `<br>` and `<center>` elements of the XHTML namespace:



24.2.3.2 Configuring Phrasing and Element-Word-Query Settings

To configure element-word-query-through, phrase-through, and phrase-around settings for a particular database, perform the following procedure in the Admin Interface:

1. Click the Databases icon in the left tree menu.

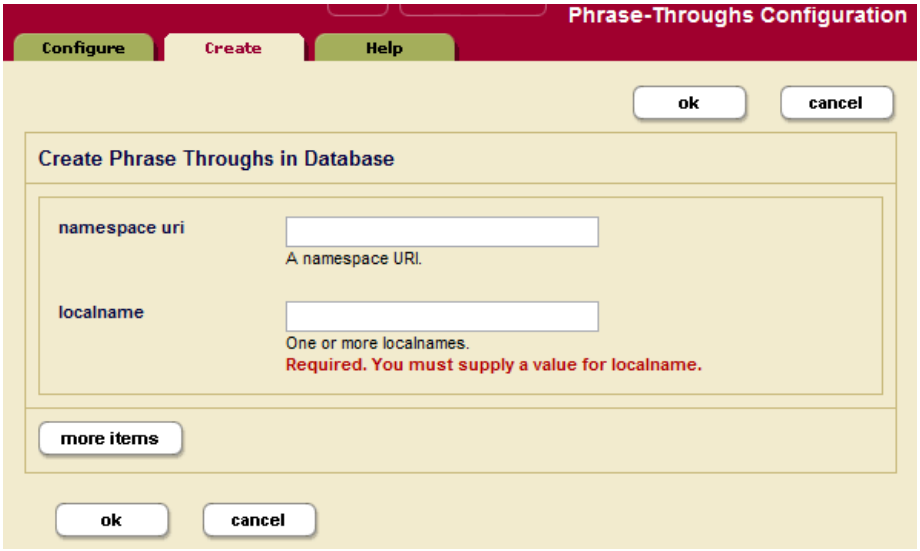
2. Locate the database for which you want to configure element-word-query-through, phrase-through, or phrase-around settings, either in the tree menu or in the Database Summary table.

3. Click the name of the database for which you want to configure the settings.

4. Click the Element-Word-Query-Throughs, Phrase-Throughs, or Phrase-Arounds icon, depending on which one you want to configure.

   Note: 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 at the top right. The Phrase-Throughs Configuration page displays:



6. Enter the namespace 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 the More Items button 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.

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.

24.2.3.3 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, perform the following procedure in the Admin Interface:

1. Click the Databases icon in the left tree menu.

2. Locate the database for which you want to delete element-word-query-through, phrase-through, or phrase-around settings, either in the tree menu or in the Database Summary table.

3. Click the name of the database for which you want to delete the settings.

4. Click the Element-Word-Query-Throughs, Phrase-Throughs, or Phrase-Arounds icon, depending on which one you want to delete.

   The appropriate configuration page displays.

5. Scroll down to the element that you want to delete.

6. Click the Drop button next to the element that you want to delete.

   A confirmation message displays.

7. 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, and includes the following parts:

- Understanding the Reindexer Enable Settings
- Query Evaluation According to the Lowest Common Denominator
- Reindexing Does Not Apply to Point-In-Time Versions of Fragments
- Example Scenario

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.

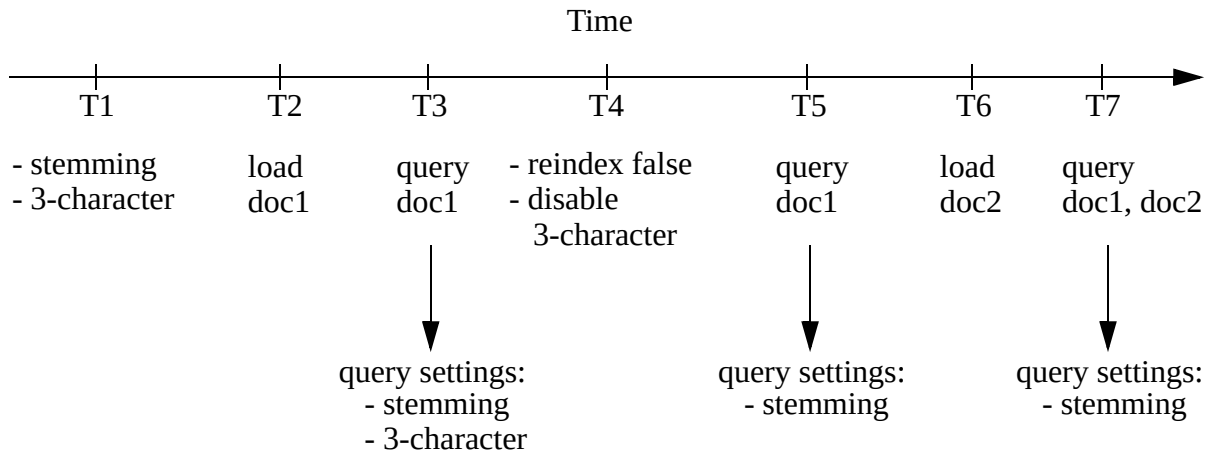
24.3.3 Reindexing Does Not Apply to Point-In-Time Versions of Fragments

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” chapter in the *Application Developer’s Guide*. For details on setting the `merge timestamp` parameter, see “Merges and Point-In-Time Queries” on page 177.

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 effect 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.

25.0 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. This chapter describes these range indexes and lexicons. The following sections are included:

- Understanding Range Indexes
- Using Range Indexes for Value Lexicons
- Understanding Word Lexicons
- Understanding Path Range Indexes
- Viewing Element Range Index Settings
- Defining Element Range Indexes
- Viewing Attribute Range Index Settings
- Defining Attribute Range Indexes
- Viewing Path Range Index Settings
- Defining Namespace Prefixes Used in Path Range Indexes and Fields
- Defining Path Range Indexes
- Viewing Element Word Lexicon Settings
- Defining Element Word Lexicons
- Viewing Attribute Word Lexicon Settings
- Defining Attribute Word Lexicons
- Defining Value Lexicons
- Deleting Range Indexes or Lexicons
- Defining Field Range Indexes

Additionally, you can create range indexes on fields, as described in “Creating a Range Index on a Field” on page 168.

This chapter 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 chapter describes the three types of range indexes shown in the table below. There are also field range indexes, as described in “Creating a Range Index on a Field” on page 168.

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 “Fields Database Settings” on page 152.

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 the Using Range Queries in cts:query Expressions chapter 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 “Understanding Word Lexicons” on page 345).

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 XMLattributes, 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* 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 "Configuring Fields" on page 159.

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 chapter 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 an error to the `ErrorLog.txt` file. 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" on page 129.

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 chapter of 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. 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. 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 exrepssions 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.

- Examples of Index Path Expressions
- Limitations on Index Path Expressions
- Testing the Validity of an Index Path Expression
- Using Namespace Prefixes in Index Path Expressions
- Grammar for Index Path Expressions

25.4.1 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/b/concat(c,d)` Function calls not allowed.
`/a/*/c`	

Valid	Invalid
`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`

Namespace prefixes are permitted in all valid path expressions.

25.4.2 Limitations on Index Path Expressions

Not all XPath features are supported by path range indexes. The limitations are described in this section. XPath expressions are summarized in XPath Quick Reference in the *XQuery and XSLT Reference Guide*. For detailed information about XPath, see the W3C XPath 2.0 language reference (http://www.w3.org/TR/xpath20/).

> **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.

A path expression used to define a path range index must conform to the following rules:

- The only operators you can use in predicate expressions are comparison operators. (`=`, `!=`, `<`, `<=`, `>=`, `>`, `eq`, `ne`, `lt`, `le`, `ge`, `gt`).
- The right operand of a comparison in a predicate can only be a string literal, numeric literal, or a sequence of string or numeric literals.
- You can only use forward axes in path steps. That is, you can use axes such as `self::`, `child::`, `descendant::`, but you cannot use reverse axes such as `parent::`, `ancestor::`, or `preceding::`. For details, see http://www.w3.org/TR/xpath/#predicates.

- You cannot span a fragment root. Paths should be scoped within fragment roots.
- You cannot use unnamed nodes in path steps. For example, when addressing JSON, you cannot include a path step such as `node()` or `array-node()`. You can use named nodes, such as `node('a')`.

Additional restrictions may apply. 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 "Testing the Validity of an Index Path Expression" on page 348.

Note that for JSON documents numbers, booleans, and nulls are indexed separately rather than all being treated as text. For details on constructing XPath expressions on JSON documents, see Traversing JSON Documents Using XPath in the *Application Developer's Guide*.

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 "Defining Path Range Indexes" on page 357.

25.4.5 Grammar for Index Path Expressions

Most users should rely on the examples in "Examples of Index Path Expressions" on page 346 and `cts:valid-index-path` to develop valid path range index expressions.

For advanced users, this section contains the grammar for the subset of XPath usable in path range index definitions, in BNF form. The grammar is derived from the W3C XML Path Language specification; for details, see http://www.w3.org/TR/xpath/.

```
IndexPathExpr ::= (PathExpr)* (("/" | "//") LeafExpr Predicates)
LeafExpr ::= "(" UnionExpr ")" | LeafStep
PathExpr ::= ("/" RelativePathExpr?) | ("//" RelativePathExpr) |
RelativePathExpr
RelativePathExpr ::= UnionExpr | "(" UnionExpr ")"
UnionExpr ::= GeneralStepExpr ("|" GeneralStepExpr)*
GeneralStepExpr ::= ("/" | "//")? StepExpr (("./" | ".//")? StepExpr)*
StepExpr ::= ForwardStep Predicates
ForwardStep ::= (ForwardAxis AbbreviatedForwardStep) |
AbbreviatedForwardStep
AbbreviatedForwardStep ::= "." | ("@" NameTest) | NameTest
LeafStep ::= ("@"QName) | QName
NameTest ::= QName | Wildcard
Wildcard ::= "*" | "<" NCName ":" "*" ">" | "<" "*" ":" NCName ">"
QName::= PrefixedName | UnprefixedName
PrefixedName ::= Prefix ":" LocalPart
UnprefixedName ::= LocalPart
Prefix ::= NCName
LocalPart ::= NCName
NCName ::= Name - (Char* ":" Char*) /* An XML Name, minus the ":" */
Name::= NameStartChar (NameChar)*
Predicates ::= Predicate*
Predicate ::= PredicateExpr | "[" Digit+ "]"
Digit ::= [0-9]
PredicateExpr ::= "[" PredicateExpr "and" PredicateExpr "]" | "["
PredicateExpr "or" PredicateExpr  "]" | "[" ComparisonExpr "]"
ComparisonExpr ::= RelativePathExpr GeneralComp SequenceExpr |
RelativePathExpr ValueComp Literal | PathExpr
GeneralComp ::= "=" | "!=" | "<" | "<=" | ">" | ">="
ValueComp ::= "eq" | "ne" | "lt" | "le" | "gt" | "ge"
SequenceExpr ::= Literal+
Literal::= NumericLiteral | StringLiteral
```

If you find it easier to explore the grammar graphically, the above BNF is suitable for use with many tools that generate "railroad diagrams" from BNF, such as `http://bottlecaps.de/rr/ui`.

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, perform the following steps:

1. Click the Databases icon on the left tree menu.

2. Locate the database whose range index you want to view, 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 range index.

4. Click the Element Range Indexes icon.

The Element Index Configuration page displays.

25.6 Defining Element Range Indexes

To define a range index for an XML element or JSON property, perform the following steps:

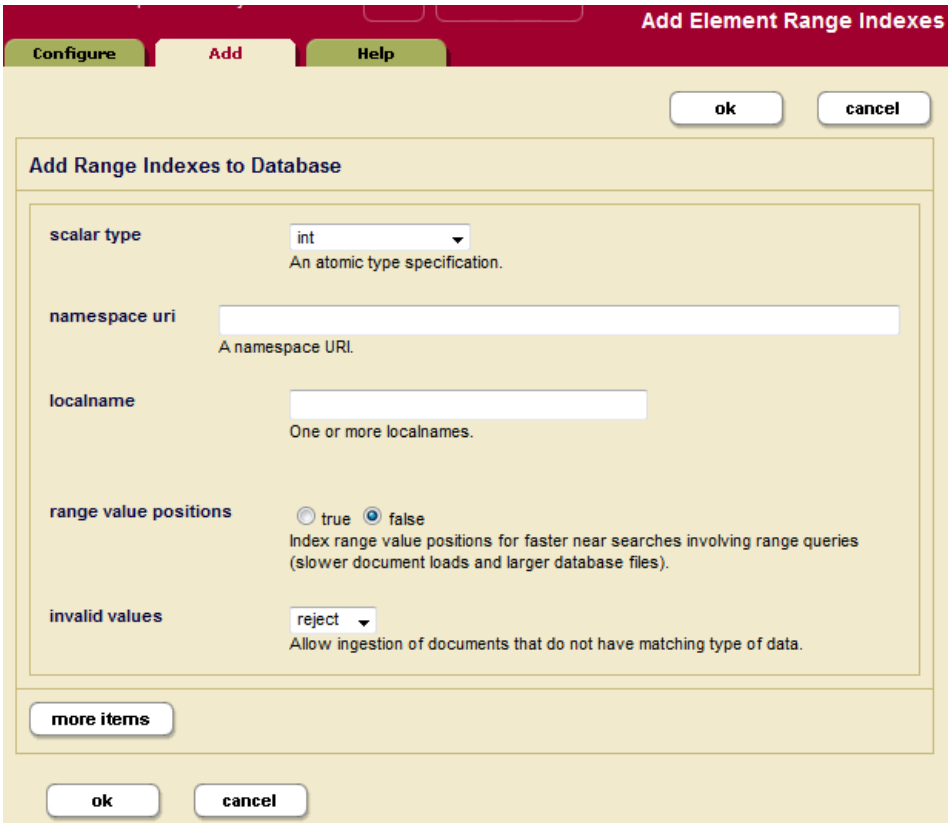
1. Click the Databases icon on the left frame.

2. Locate the database for which you want to create a range index, either in the tree menu or in the Database Summary table.

3. Click the name of the database for which you want to create a range index.

4. Click the Element Range Indexes icon in the tree menu, under the selected database.

5. Click the Add tab. The The Add Range Indexes configuration page displays:



6. Select the type of the XML element or JSON property for which you want to build a range index.

7. If the index is of type `xs:string`, a collation box appears with a default collation. If you want the index to use a different collation than the default, enter the collation URI. You can 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 the Language Support in MarkLogic Server chapter in the *Search Developer's Guide*.

8. 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.

9. Enter the element or JSON property name in the localname field.

   The local name 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.

10. Choose whether to index range value positions for this index. Setting range value positions to `true` will speed the performance of searches that use `cts:near-query` and `cts:element-query` with this index, but will use more disk space than leaving the positions off (range value positions `false`).

11. 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. It can be configured to either `ignore` or `reject`. By default server rejects insertion of such documents. However, if you configure invalid values to `ignore`, documents containing invalid element or JSON property values can be inserted, but the invalid values will not be indexed. Range queries and lexicon functions that mainly operate off of range index will ignore existence of such documents in the database.

12. To add more indexes, click the More Items button and repeat step 6 – step 11 for each index as needed.

13. 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, perform the following steps:

1. Click the Databases icon on the left frame.

2. Locate the database for which you want to view a range index, 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 range index.

4. Click the Attribute Range Indexes icon in the tree menu, under the selected database.

The Attribute Range Index Configuration page displays.

25.8 Defining Attribute Range Indexes

To define a range index for an attribute of a particular element, perform the following steps:

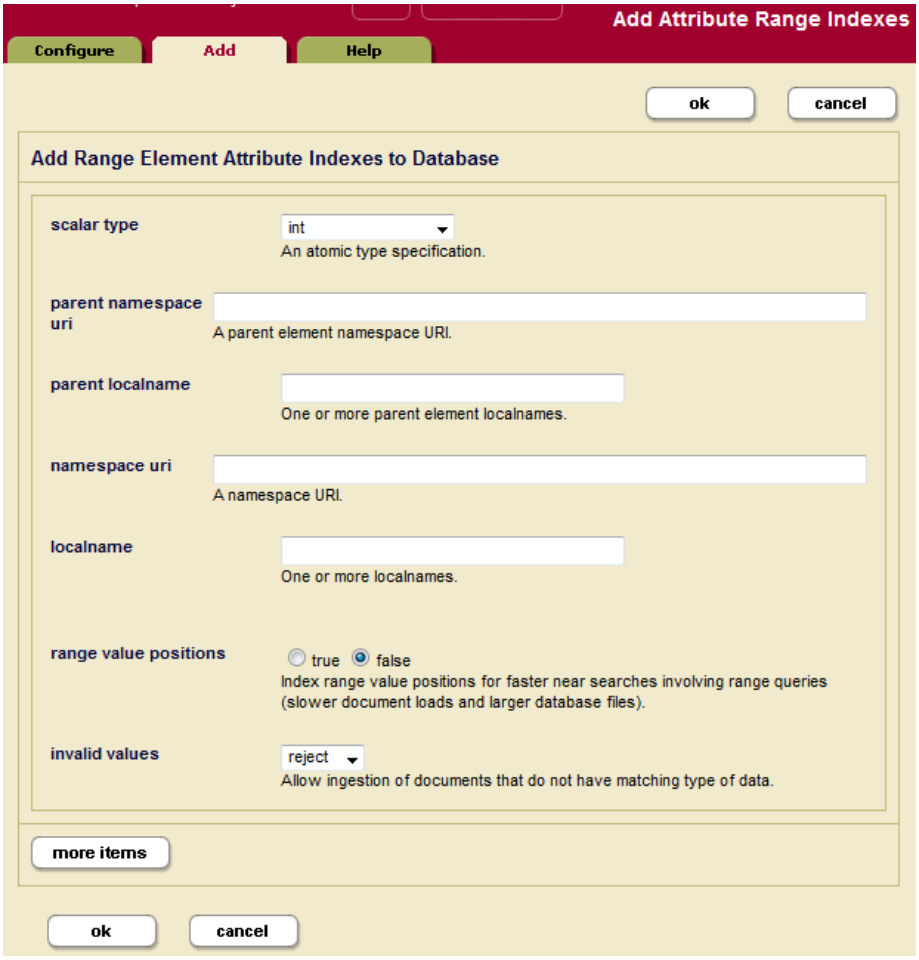
1. Click the Databases icon on the left frame.

2. Locate the database for which you want to create an index, either in the tree menu or in the Database Summary table.

3. Click the name of the database for which you want to create an index.

4. Under the selected database, click the Attribute Range Indexes icon in the tree menu for an attribute range index.

5. Click the Add tab. The Add Attribute Range Indexes page displays:



6. Select the type of the XML attribute for which you want to build an attribute range index.

7. If the index is of type `xs:string`, a collation box appears with a default collation. If you want the index to use a different collation than the default, enter the collation URI. You can 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 the Language Support in MarkLogic Server chapter in the *Search Developer's Guide*.

8. 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.

9. 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.

10. 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.

11. 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.

12. Choose whether to index range value positions for this index. Setting the value to `true` will speed the performance of searches that use `cts:near-query` and `cts:element-query` with this index, but will use more disk space than leaving the positions off (range value positions `false`).

13. 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. It can be configured to either `ignore` or `reject`. By default server rejects insertion of such documents. However, if you configure invalid values to `ignore`, documents containing invalid attributes can be inserted, but the invalid attribute values will not be indexed. Range queries and lexicon functions that mainly operate off of range index will ignore existence of such documents in the database.

14. To add more indexes, click the More Items button and repeat step 6 – step 13 for each attribute index as needed.

15. 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, perform the following steps:

1. Click the Databases icon on the left frame.

2. Locate the database for which you want to view a range index, 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 range index.

4. Click the Path Range Indexes icon in the tree menu, under the selected database.

The Path Range Index Configuration page displays.

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, perform the following steps:

1. Click the Databases icon on the left frame.

2. Locate the database for which you want to create a namespace prefix binding, either in the tree menu or in the Database Summary table.

3. Click the name of the database for which you want to create a namespace binding.

4. Click the Path Namespaces icon in the tree menu, under the selected database.

5. Click the Add tab. The Path Namespaces Configuration page displays:



6. 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.

7. In the Namespace URI field, enter the namespace URI of the XML element or attribute in the XPath expression.

8. Click OK.

25.11 Defining Path Range Indexes

To define a range index expressed by an XPath expression, perform the following 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 “Defining Namespace Prefixes Used in Path Range Indexes and Fields” on page 356.

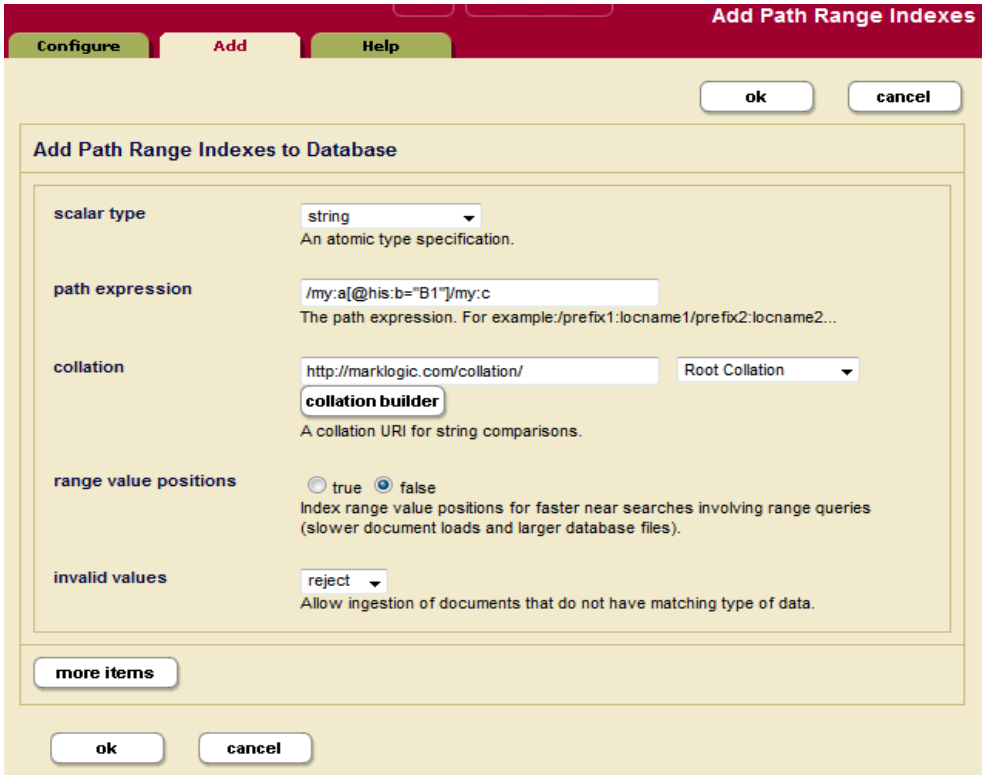
2. Click the Databases icon on the left frame.

3. Locate the database for which you want to create a range index, either in the tree menu or in the Database Summary table.

4. Click the name of the database for which you want to create a range index.

5. Click the Path Range Indexes icon in the tree menu, under the selected database.

6. Click the Add tab. The Path Range Index Configuration page displays:



7. Select the type of the XML element, XML attribute, or JSON property for which you want to build a range index.

8. If the index is of type `xs:string`, a collation box appears with a default collation. If you want the index to use a different collation than the default, enter the collation URI. You can 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 the Language Support in MarkLogic Server chapter in the *Search Developer's Guide*.

9. 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 "Understanding Path Range Indexes" on page 346.

   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.

   Note: You cannot have a path span across a fragment root. Paths should be scoped within fragment roots

10. Choose whether to index range value positions for this index. Setting the value to `true` will speed the performance of searches that use `cts:near-query`, `cts:element-query`, and

`cts:json-property-scope-query` with this index, but will use more disk space than leaving the positions off (range value positions `false`).

11. 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. It can be configured to either `ignore` or `reject`. By default server rejects insertion of such documents. However, if you configure invalid values to `ignore`, documents containing invalid such values can be inserted, but the invalid values will not be indexed. Range queries and lexicon functions that mainly operate off of range index will ignore the existence of such documents in the database.

12. To add more indexes, click the More Items button and repeat step 7 – step 11 for each index as needed.

13. 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.

Note: 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 Word Lexicon Settings

To view the lexicon that will be applied to documents as they are loaded or reindexed, perform the following steps:

1. Click the Databases icon on the left tree menu.

2. Locate the database whose range index or lexicon you want to view, 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 lexicon.

4. Click the Element Word Lexicons icon.

The Element Word Lexicon Configuration page displays.

25.13 Defining Element Word Lexicons

To define a lexicon for an XML element or JSON property, perform the following steps:

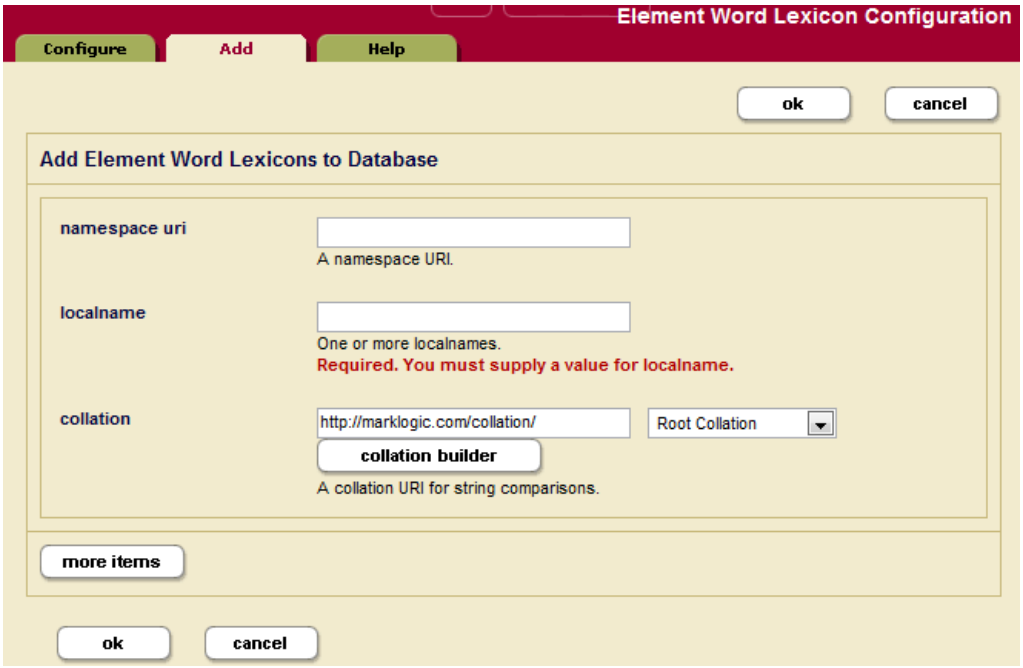
1. Click the Databases icon on the left frame.

2. Locate the database for which you want to create lexicon, either in the tree menu or in the Database Summary table.

3. Click the name of the database for which you want to create a lexicon.

4. Click the Element Word Lexicons icon in the tree menu, under the selected database.

5. Click the Add tab. The Element Word Lexicon Configuration page displays:



6. If you are defining a lexicon on an XML element, enter the namespace URI of the XML element.

   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 XML element or JSON property name in 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.

8. 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. You can 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 the Language Support in MarkLogic Server chapter in the *Search Developer's Guide*.

9. To add more word lexicons, click the More Items button and repeat step 6 – step 8 for each lexicon as needed.

10. 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, perform the following steps:

1. Click the Databases icon on the left frame.

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 displays.

25.15 Defining Attribute Word Lexicons

To define a lexicon for an attribute of a particular element, perform the following steps:

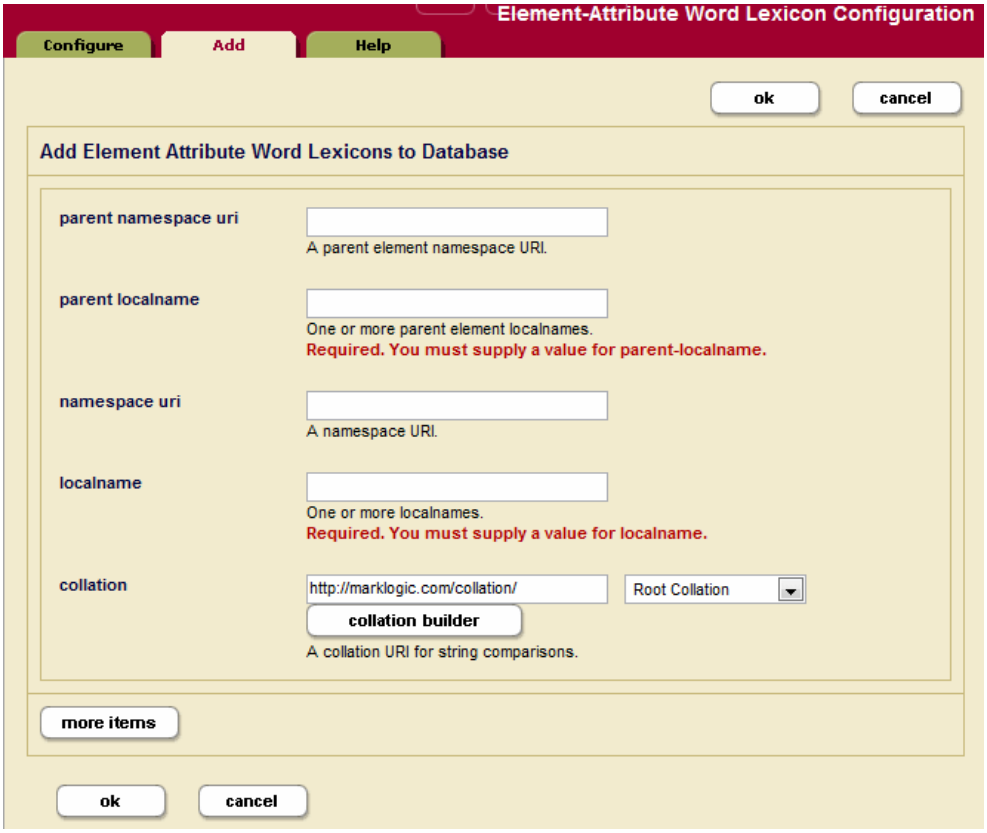
1. Click the Databases icon on the left frame.

2. Locate the database for which you want to create 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 create a lexicon.

4. Under the selected database, click the Attribute Word Lexicon icon.

5. Click the Add tab. The Element-Attribute Word Lexicon Configuration page displays:



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. 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. You can 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 the Language Support in MarkLogic Server chapter in the *Search Developer's Guide*.

11. To add more element-attribute word lexicons, click the More Items button and repeat step 6 – step 10 for each attribute index as needed.

12. 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, perform the following steps:

1. Click the Databases icon on the left frame.

2. Locate the database for which you want to delete a range index or lexicon, either in the tree menu or in the Database Summary table.

3. Click the name of the database for which you want to delete a range index or lexicon.

4. Determine whether you need to delete an element range index, an attribute range index, an element word lexicon, or an attribute word lexicon.

5. Click the Element Range Index icon, Attribute Range Index icon, Path Range Index icon, Element Word Lexicon icon, or the Attribute Word Lexicon icon. The configuration page for the appropriate index appears.

6. Locate the index you want to delete and click Delete.

7. A confirmation message displays. 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 “Fields Database Settings” on page 152.

26.0 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.

Use the following procedures for managing fragmentation rules:

- Choosing a Fragmentation Strategy
- Defining Fragment Roots
- Defining Fragment Parents
- Viewing Fragment Rules
- Deleting Fragment 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 threhold” database configuration setting.

After you decide how to fragment your data, you can use either of the following methods:

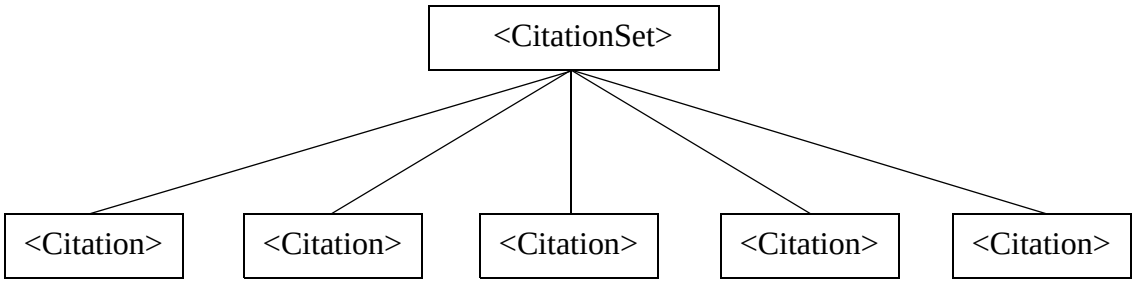
- Fragment Roots
- Fragment Parents

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.

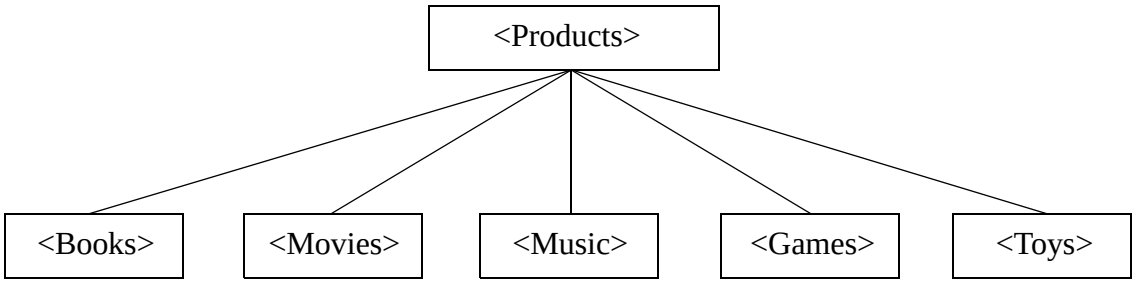
The following diagram shows an XML document rooted at `<CitationSet>` that contains many instances of a `<Citation>` node. Each `<Citation>` node contains further XML and averages between 15K and 20K in size. Based on this information, `<Citation>` is a sensible element to use as a fragment root:



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:

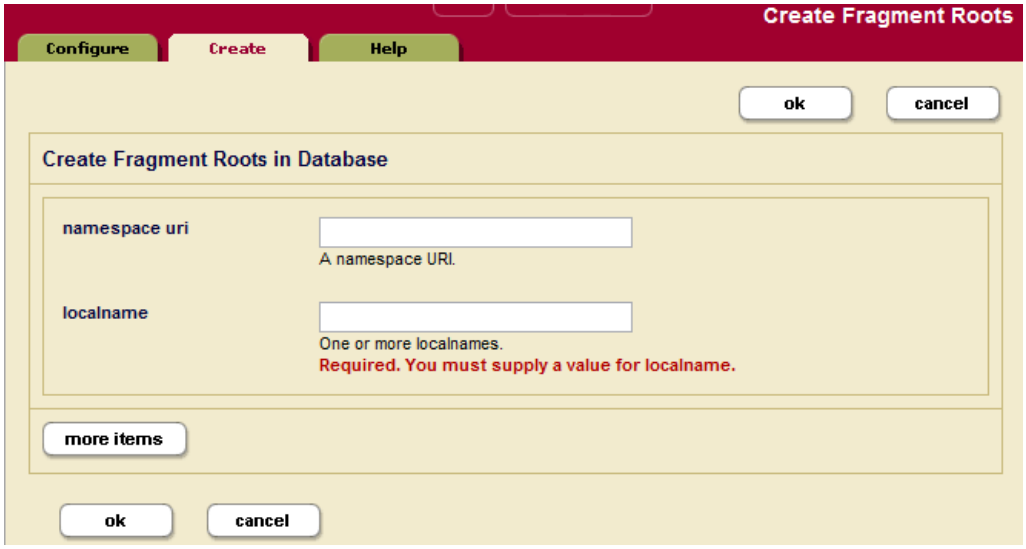


In this case, you can use the `<Products>` element as a fragment parent, and the `<Books>`, `<Movies>`, `<Music>`, `<Games>` and `<Toys>` children automatically become fragments.

26.2 Defining Fragment Roots

To define a rule for a fragment root, complete the following procedure:

1. Click the Databases icon on the left tree menu.
2. Determine the database for which you are specifying a new fragment rule.
3. Click the icon for this database, either in the tree menu or the Database Summary page.
4. Click the Fragment Roots icon.
5. Click the Create tab. The Fragment Roots Configuration page displays:



6. Enter the namespace URI of the XML element that you are using as a rule for the fragment root.

   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 (*) in the namespace URI field.

7. Enter the element name in 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.

8. To add more fragment roots, click the More Items button and repeat step 6 – step 7 for each fragment root as needed.

9. 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, perform the following steps:

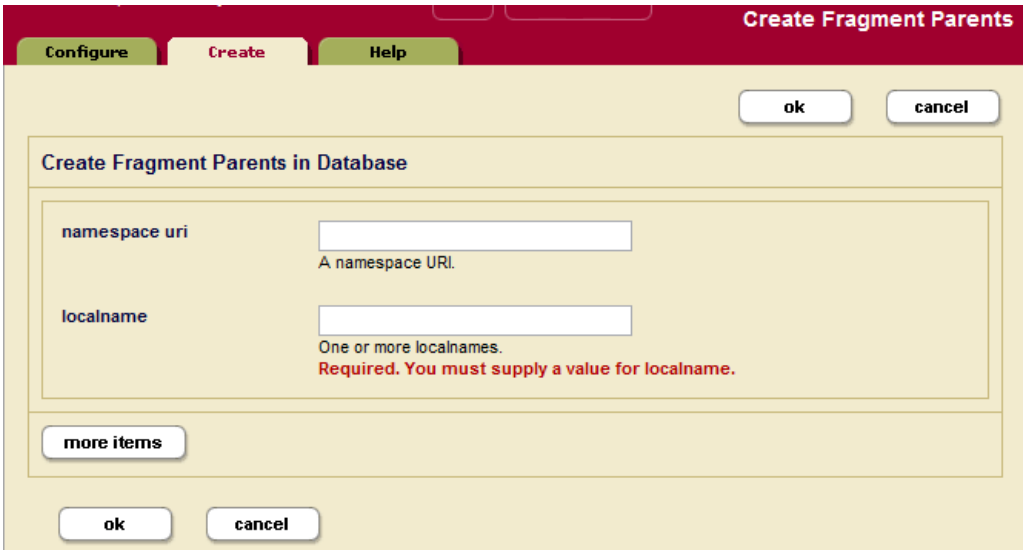
1. Click the Databases icon on the left tree menu.

2. Determine the database for which you are specifying a new fragment parent.

3. Click the icon for this database, either in the tree menu or the Database Summary page.

4. Click the Fragment Parents icon.

5. Click the Create tab. The Fragment Parents Configuration page displays:



6. Enter the namespace URI of the XML element that you are using as a rule for the fragment parent.

   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 (*) in the namespace URI field.

7. Enter the element name in 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.

8. To add more fragment parents, click the More Items button and repeat step 6 – step 7 for each fragment parent as needed.

9. 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, perform the following steps:

1. Click the Databases icon on the left tree menu.

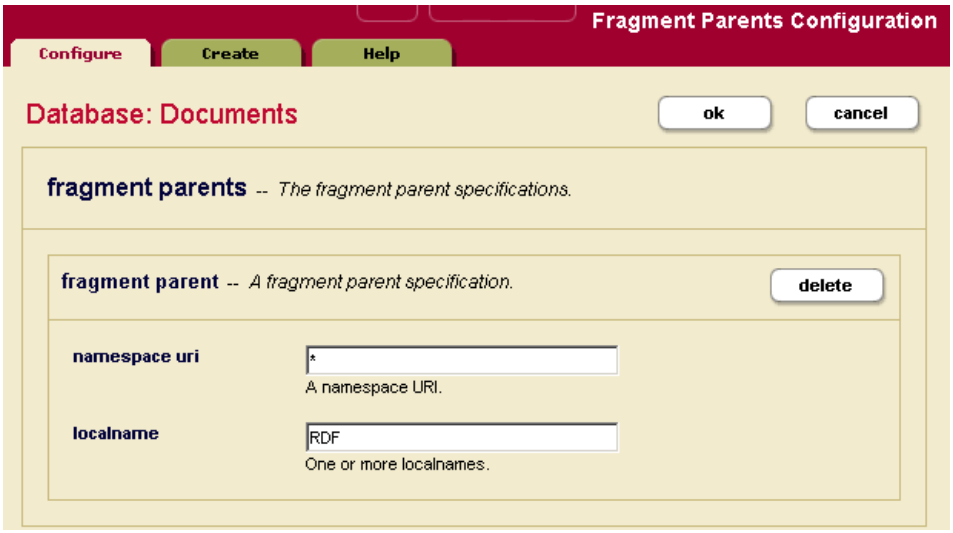
2. Locate the database whose fragment rules you want to view, either in the tree menu or the Database Summary page.

3. Click the icon for this database.

4. Determine whether to view the rules for the fragment root or fragment parent.

5. Click either the Fragment Roots icon 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, perform the following steps:

1. Click the Databases icon on the left tree menu.
2. Locate the database that contains the fragment rules you want to delete, either in the tree menu or the Database Summary page.
3. Click the icon for this database.
4. Determine whether you need to delete a rule for a fragment root or fragment parent.
5. Click either the Fragment Roots icon or Fragment Parents icon, under the specified database.
6. Locate the fragment rule you want to delete and click Delete.
7. A confirmation message displays. 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.

27.0 Namespaces

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 "Namespaces" chapter in *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.

Use the following procedures for managing namespaces in the Admin Interface:

- Defining Namespaces for a Group
- Defining Namespaces for an HTTP, ODBC, or XDBC Server
- Viewing Namespace Settings for a Group
- Viewing Namespace Settings for an HTTP, ODBC, or XDBC Server
- Deleting Namespaces for a Group
- Deleting Namespaces for an HTTP, ODBC, or XDBC Server

This chapter 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*.

27.1 Defining Namespaces for a Group

To define namespaces using the Admin Interface for a group, perform the following steps:

1. Click the Groups icon on the left tree menu.
2. Click the group in which you want to define the namespace, either in the tree menu or the Groups Summary page.
3. Click the Namespaces icon on the left tree menu, under the group name.
4. Click the Add tab. The Namespaces Configuration page displays:



5. Enter a prefix for your namespace.
6. Enter a URI for your namespace.

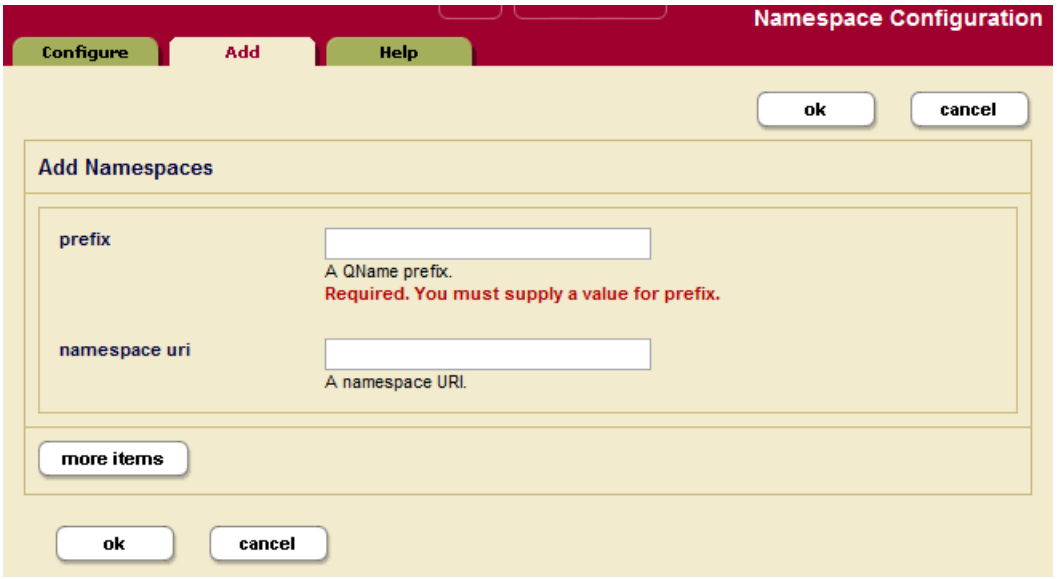
   If you are defining a prefix for the universal unnamed namespace, leave the URI blank.
7. To add more namespace definitions, click the More Items button and repeat step 5 – step 6 for each namespace as needed.
8. Scroll to the top or bottom and click OK.

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, perform the following steps:

1. Click the Groups icon on the left tree menu.
2. Click the group which contains the HTTP, ODBC, or XDBC server for which you want to define the namespace, either in the tree menu or the Groups Summary page.
3. Click the App Servers icon as appropriate.
4. Click on the name of the App server for which you want to define the namespace.
5. Click on the Namespaces icon on the left tree menu, under the specified App server.
6. Click the Add tab at the top right. The Namespaces Configuration page displays:



7. Enter a prefix for your namespace.
8. Enter a URI for your namespace.

   If you are defining a prefix for the universal unnamed namespace, leave the URI blank.
9. To add more namespace definitions, click the More Items button 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.

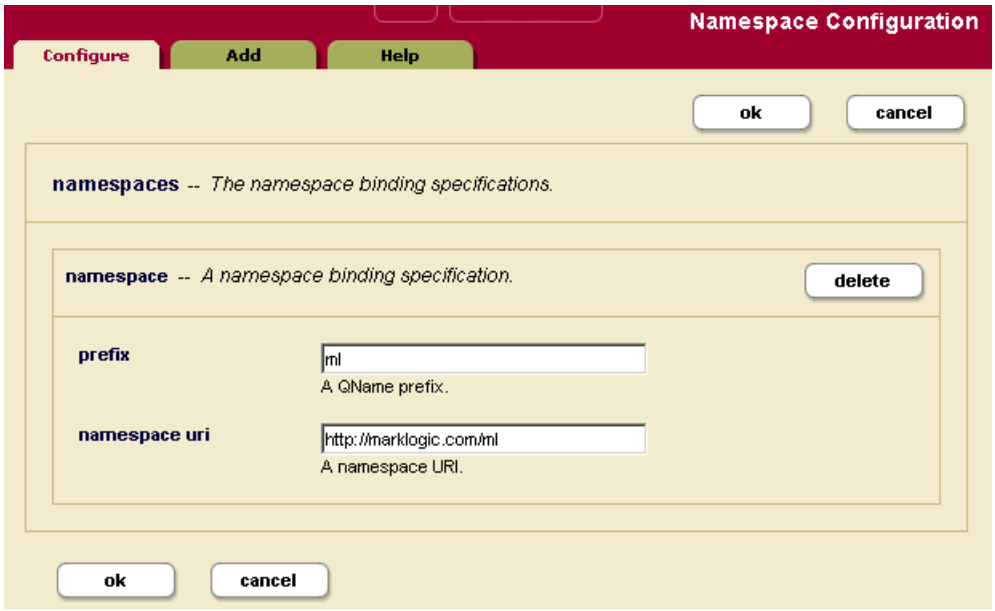
27.3 Viewing Namespace Settings for a Group

To view namespaces you have defined in the Admin Interface, perform the following steps:

1. Click the Groups icon on the left tree menu.

2. Click the group which contains the namespace you want to view, either in the tree menu or the Groups Summary page.

3. Click the Namespaces icon on the left tree menu, under the specified group. The Namespace Configuration page appears.



27.4 Viewing Namespace Settings for an HTTP, ODBC, or XDBC Server

To view namespaces you have defined in the Admin Interface, perform the following steps:

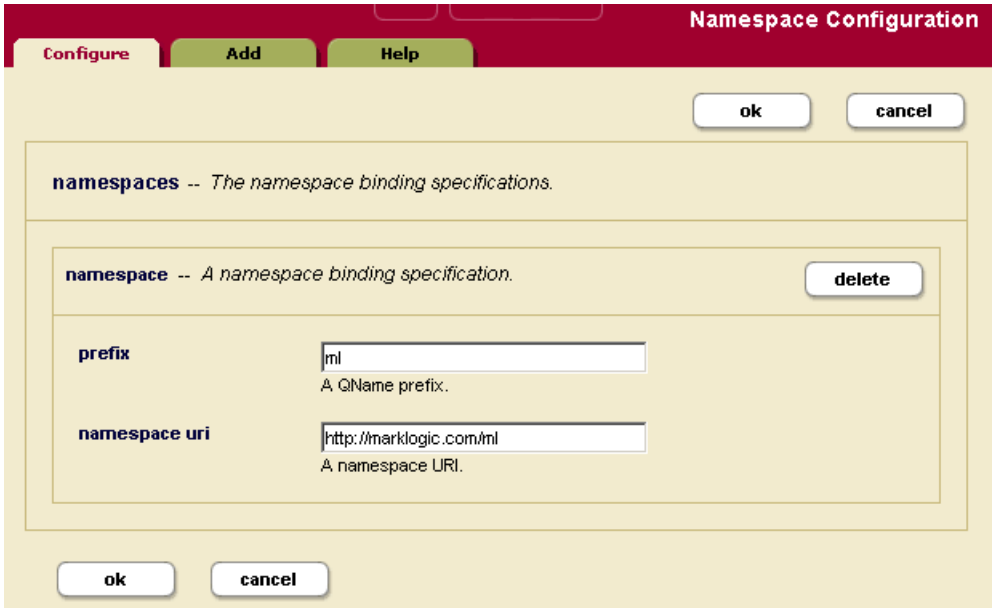
1. Click the Groups icon on the left menu tree.

2. Click the group which contains the HTTP, ODBC, or XDBC server for which you want to view the namespace, either in the tree menu or the Groups Summary page.

3. Click the App Servers icon as appropriate.

4. Click on the name of the App Server for which you want to view the namespace.

5. Click the Namespaces icon on the left tree menu, under the specified App Server. The Namespace Configuration page appears.



27.5 Deleting Namespaces for a Group

To delete namespaces that you defined in the Admin Interface, perform the following steps:

1. Click the Groups icon on the left tree menu.

2. Click the group from which you want to delete the namespace, either in the tree menu or the Group Summary page.

3. Click the Namespaces icon on the left tree menu, under the specified group.

4. Locate the namespace to be deleted and click Delete.

5. A confirmation message displays. 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, perform the following steps:

1. Click the Groups icon on the left tree menu.

2. Click on the group which contains the App Server from which you want to delete the namespace, either in the tree menu or the Group Summary page.

3. Click on the App Servers icon.

4. Click on the name of the App Server from which you want to delete the namespace, either in the tree menu or the App Server Summary page.

5. Click the Namespaces icon on the left tree menu, under the specified App Server. The namespace configuration screen appears.

6. Locate the namespace to be deleted and click Delete.

7. A confirmation message displays. Confirm the delete and click OK.

The namespace is deleted from the App Server.

28.0 Understanding and Defining Schemas

This chapter describes schemas and lists procedures for defining them. The following topics are included:

- Understanding Schemas
- Procedures For Defining Schemas

For more information on the Schema database, loading schemas into MarkLogic Server, and using schemas in your applications, see the “Loading Schemas” chapter of 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 the following 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 the following 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

Use the following procedures for defining schemas:

- Adding a Schema Definition for a Group
- Adding a Schema Definition for an HTTP, ODBC, or XDBC Server
- Viewing Schema Definitions for a Group
- Viewing Schema Definitions for an HTTP, ODBC, or XDBC Server
- Deleting a Schema Definition for a Group
- Deleting a Schema Definition for an HTTP, ODBC, or XDBC Server

28.2.1 Adding a Schema Definition for a Group

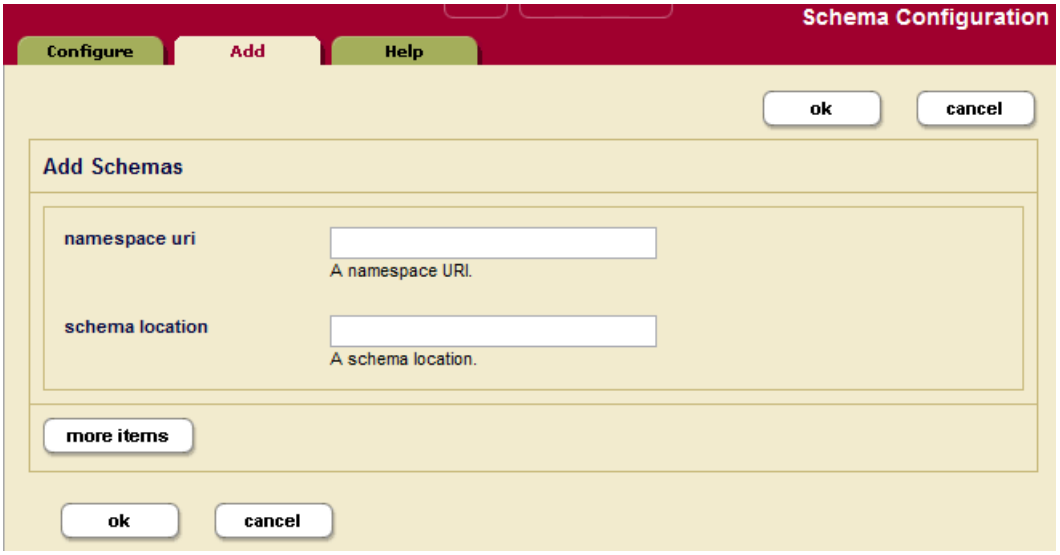
To make a schema available to all HTTP, ODBC, or XDBC servers in a group, complete the following procedure:

1. Click the Groups icon on the left tree menu.

2. Click the group in which you want to define the schema.

3. Click the Schemas icon on the left tree menu, under the specified group.

4. Click the Add tab. The Schema Configuration page displays:



5. Enter a namespace URI and corresponding schema location.

   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`
Sun Solaris	`/opt/MARKlogic/Config`

6. To add more schema definitions, click the More Items button and repeat step 5 for other schemas as needed.

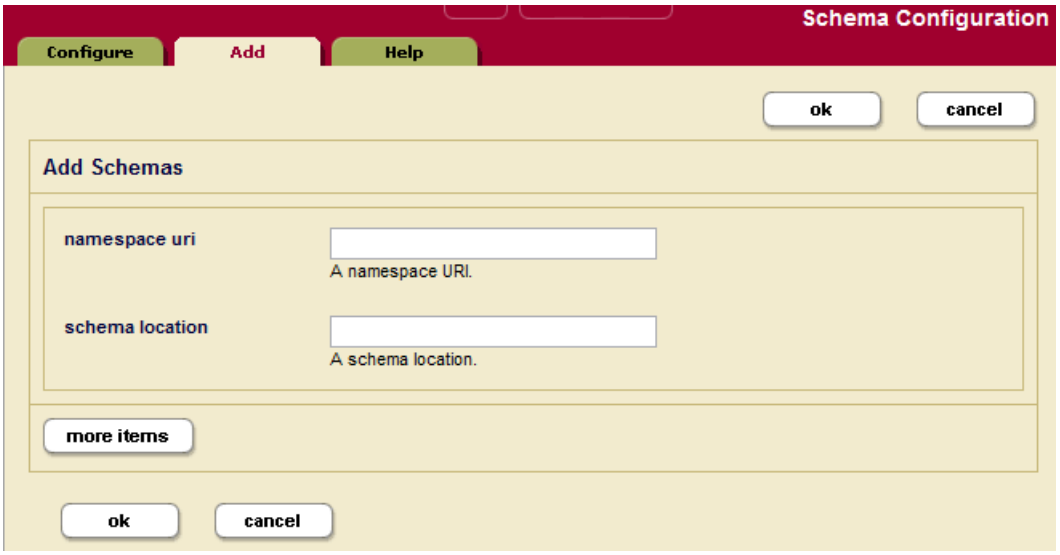
7. Scroll to the top or bottom and click OK.

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, perform the following steps:

1. Click the Groups icon on the left tree menu.
2. Click the name of the group which contains the HTTP, ODBC, or XDBC server to which you want to add a schema.
3. Click the App Servers icon.
4. Click the name of the App Server to which you want to add a schema.
5. Click the Schemas icon on the left tree menu, under the specified App Server.
6. Click the Add tab. The Schema Configuration page displays:



7. Enter a namespace URI and corresponding schema location.

   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`
Sun Solaris	`/opt/MARKlogic/Config`

8. To add more schema definitions, click the More Items button 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

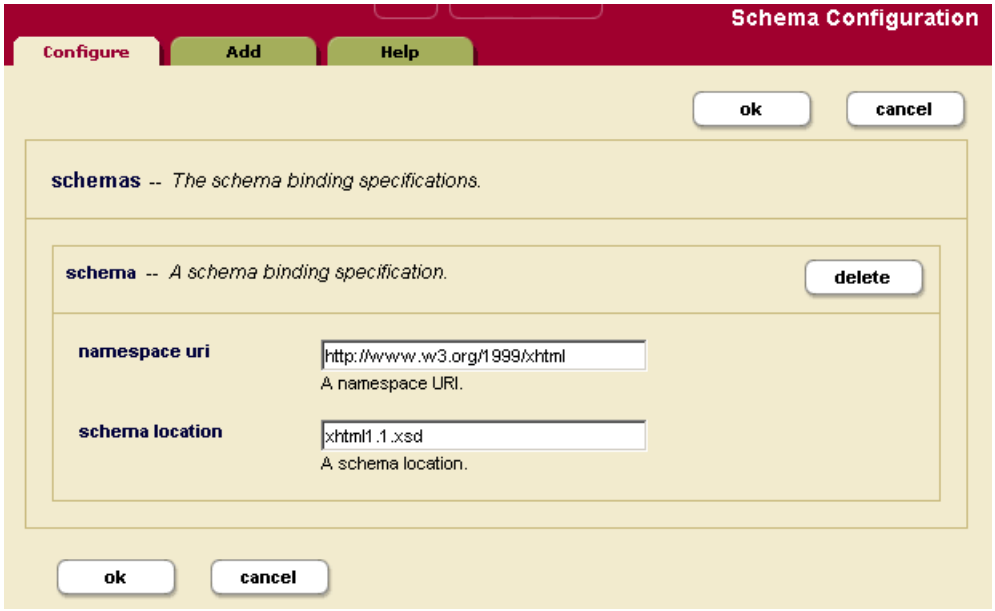
To view a schema definition for a group, complete the following procedure:

1. Click the Groups icon on the left tree menu.

2. Click the group that contains the schema you want to view.

3. Click the Schemas icon on the left tree menu, under the specified group.

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.

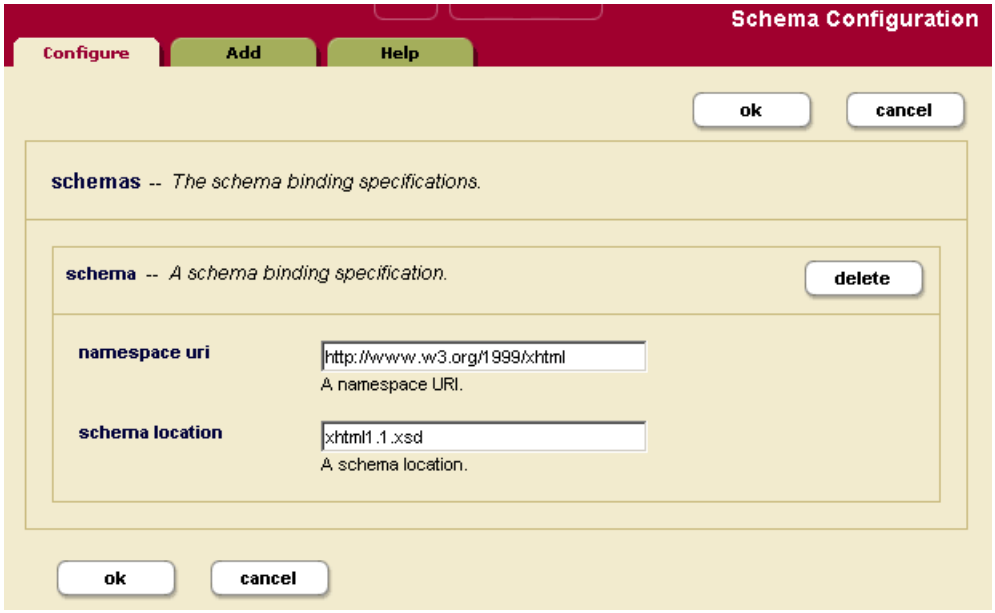

Schema Configuration
Configure
Add
Help
ok
cancel
schemas -- The schema binding specifications.
schema -- A schema binding specification.
delete
namespace uri
http://www.w3.org/1999/xhtml
A namespace URI.
schema location
xhtml1.1.xsd
A schema location.
ok
cancel

28.2.4 Viewing Schema Definitions for an HTTP, ODBC, or XDBC Server

To view a schema definition for an HTTP or XDBC Server, perform the following steps:

1. Click the Groups icon on the left tree menu.
2. Click on the name of the group which contains the HTTP , ODBC, or XDBC server with the schema you want to view.
3. Click the App Servers icon.
4. Click the name of the App Server with the schema you want to view.
5. Click the Schemas icon on the left tree menu, under the specified App Server.

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, perform the following steps:

1. Click the Groups icon on the left tree menu.
2. Click the group from which you want to delete the schema.
3. Click the Schemas icon on the left tree menu, under the specified group.
4. Locate the schema definition to be deleted from the system and click Delete.
5. A confirmation message displays. Confirm the delete and click OK.

The schema is dropped 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, perform the following steps:

1. Click the Groups icon on the left tree menu.
2. Click the name of the group which contains the HTTP, ODBC, or XDBC server with the schema you want to delete.
3. Click the App Servers icon.
4. Click the name of the App Server with the schema you want to delete.
5. Click the Schemas icon on the left tree menu, under the specified App Server.
6. Click the Schemas icon on the left tree menu, under the specified App Server.
7. Locate the schema definition to be deleted from the system and click Delete.
8. A confirmation message displays. Confirm the delete and click OK.

The schema is dropped from the App Server.

29.0 Log Files

This chapter describes the log files and includes the following sections:

- Understanding the Log Levels
- Configuring Log Files
- Viewing the System Log
- Viewing the File Log
- Access Log Files

For information on the audit log files, see “Auditing Events” on page 106.

29.1 Understanding the Log Levels

MarkLogic Server sends log messages to both the operating system log and the MarkLogic Server file log. Depending on how you configure your logging functions, both 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. For example, if you are debugging a system problem, you may want to set the level to Debug to get more information. Keep in mind that log levels Debug and above degrade system performance significantly, so these log levels should not normally be used.

29.2 Configuring Log Files

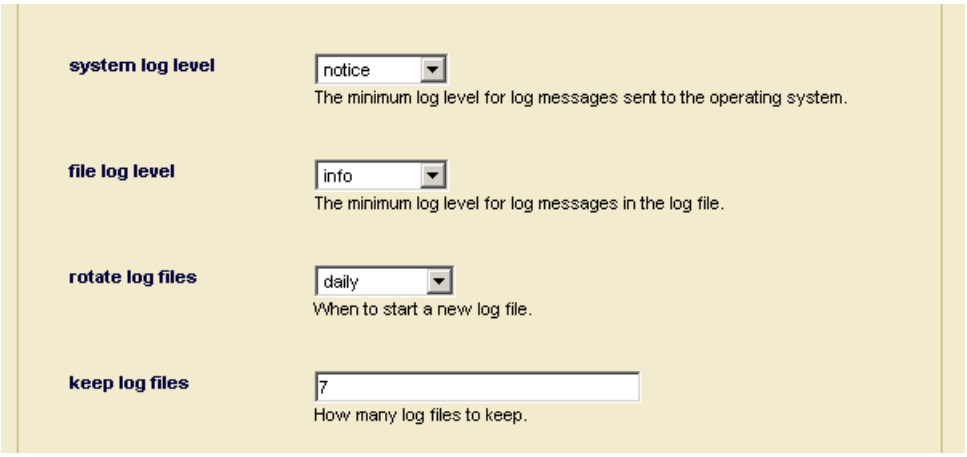
To configure how log information is generated, perform the following steps:

1. Click the Groups icon on the left tree menu.

2. Click the group for which you want to configure the log file settings.

3. Scroll down to the log settings, towards the bottom of the page.

   The following example shows the default log settings:



4. Go to System Log Level and change the level if needed.

5. Go to File Log Level and change the logging level of the MarkLogic Server private log file (`ErrorLog.txt`) if needed.

6. Go to Rotate Log Files and 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.

7. Go to Keep Log Files and 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 File setting, when a new log file is started, the oldest log file archive is automatically deleted.

8. Scroll to the top or bottom and click OK.

29.3 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.

29.4 Viewing the File Log

The private file log is maintained as a simple text file. 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 file is stored in 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 File
Microsoft Windows	`C:\Program Files\MarkLogic\Data\Logs\ErrorLog.txt`
Red Hat Linux	`/var/opt/MarkLogic/Logs/ErrorLog.txt`
Sun Solaris	`/var/opt/MARKlogic/Logs/ErrorLog.txt`
Mac OS X	`~/Library/Application Support/MarkLogic/Data/Logs/ErrorLog.txt`

This file contains 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.

> **Note:** There must be sufficient disk space on the file system in which the log files reside. If there is no space left on the log file device, MarkLogic Server will abort. Additionally, if there is no disk space available for the log files, MarkLogic Server will fail to start.

29.5 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 “Configuring Log Files” on page 387.

30.0 Scheduling Tasks

This chapter describes how to schedule tasks that execute XQuery main modules at a predefined date/time or interval. The following topics are included:

- Understanding Scheduled Tasks
- Scheduling a Module for Invocation
- Selecting a Task Type

This chapter describes how to use the Admin Interface to manage scheduled tasks. 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, do the following:

1. Click the Groups icon in the left frame.

2. Click on the group in which you want to schedule a task (for example, Default).

3. Click the Scheduled Tasks icon on the left tree menu.

4. Click on 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 (files system) or URI root (database) that contains the module. For example, if the module is located in the file system under `MarkLogic/Docs`, specify `Docs`.

7. In the Task Type field, select one of the task types described in “Selecting a Task Type” on page 393.

8. In the Database field, select the database on which to invoke the module.

9. In the Task Modules field, select either the file system or 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. For example, in the configuration shown below, you would place the `Scheduler_test.xqy` file in the `MarkLogic/Docs` directory.

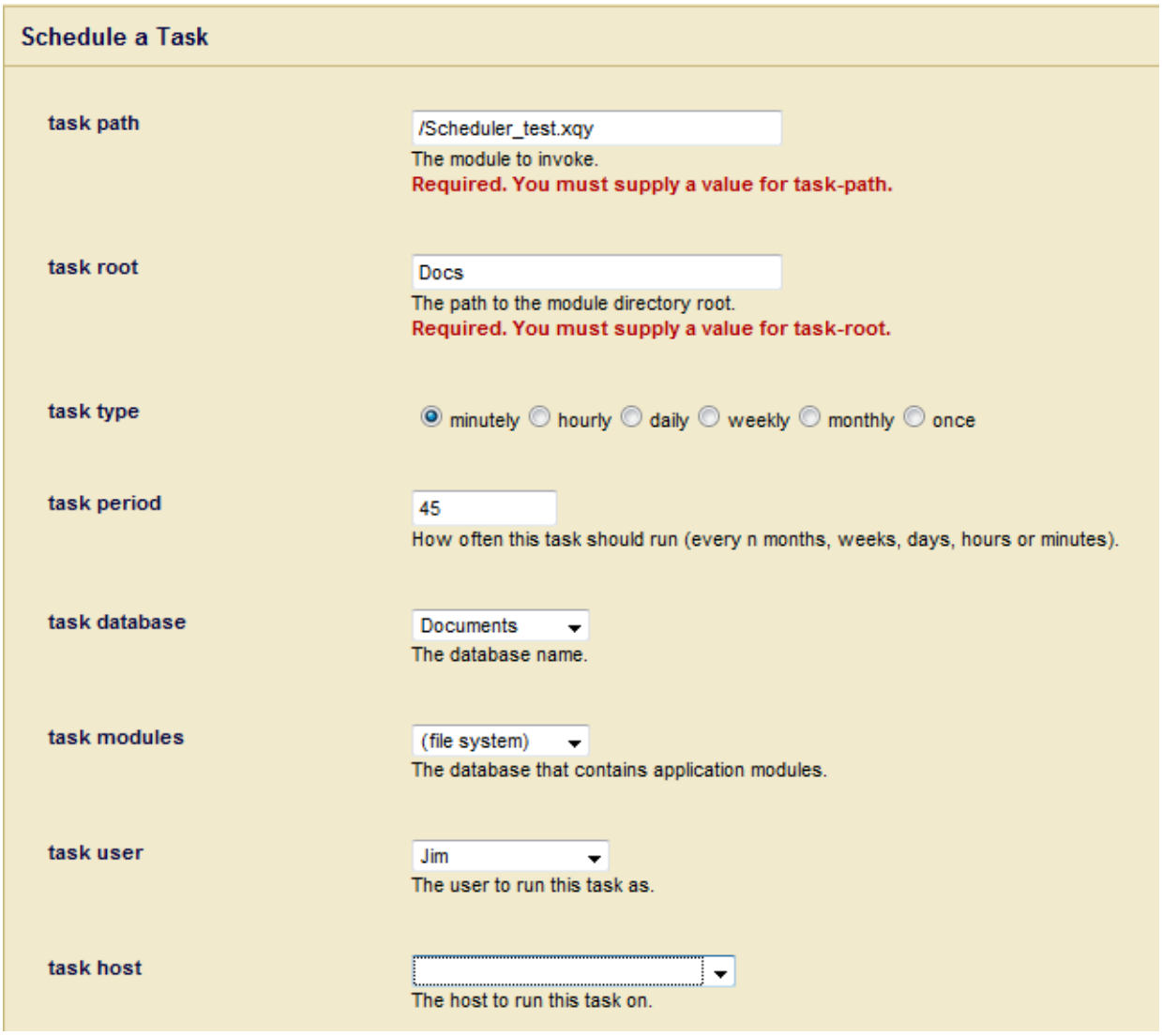
   If Task Modules is set to a database, then load the module into that database under the URI root specified by Task Root. For example, if the configuration shown below specified the `Documents` database in the Task Modules field, you could use the `xdmp:document-load` function to load the module with the following URI option:

   ```
   <uri>Docs/Scheduler_test.xqy</uri>
   ```

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 "Appendix B: Pre-defined Execute Privileges" on page 418 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.

The interval scheduling options that operate on elapsed time are:

- Scheduling Per Minute
- Scheduling Per Hour

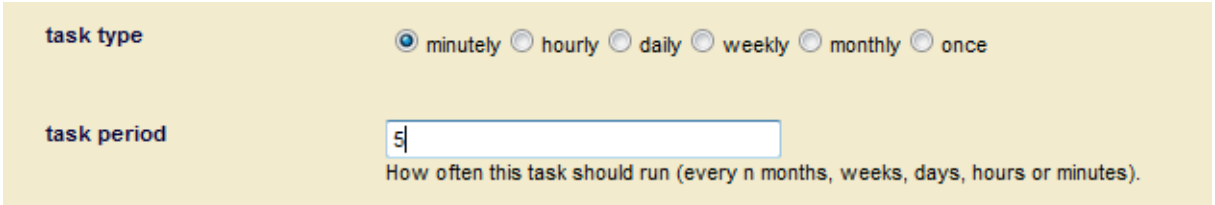
The date/time scheduling options that operate on calendar time are:

- Scheduling Per Day and Time
- Scheduling Per Week, Day, and Time
- Scheduling Per Month, Day, and Time
- Scheduling One Invocation on a Calendar Date and Time

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

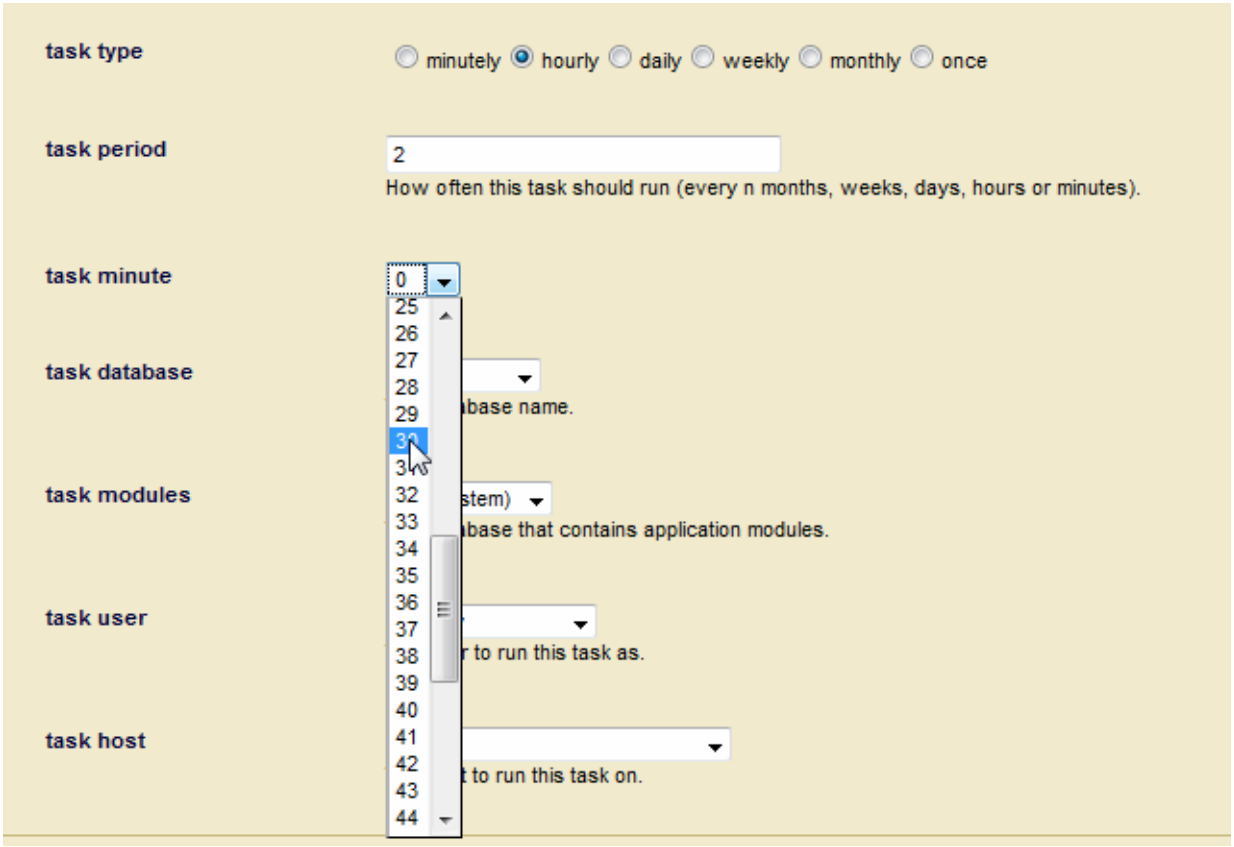
If you select minutely task type, specify how many minutes are to elapse between each invocation of the module. For example, to invoke the module every 5 minutes (or as soon as possible thereafter, if the server is overloaded), enter:



30.3.2 Scheduling Per Hour

If you select hourly task type, specify how many hours are to elapse between each invocation of the module. The Task Minute setting 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.

For example, to invoke the module every 2 hours at 30 minutes past the hour (or as soon as possible thereafter, if the server is overloaded), enter:

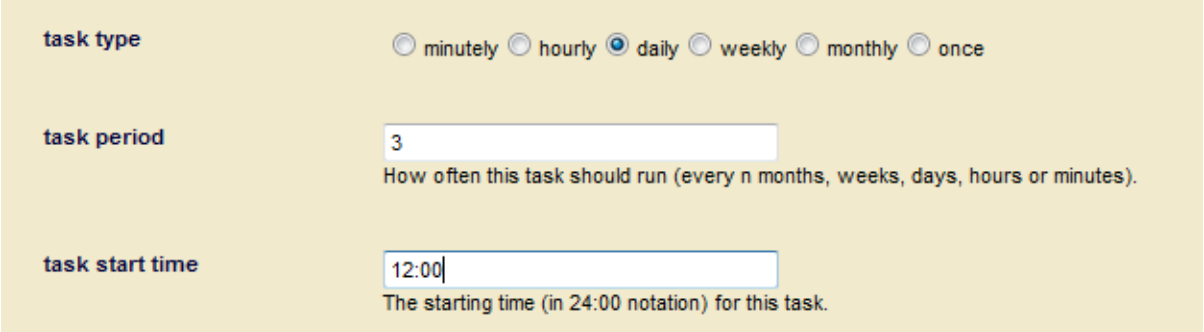


If the current time is 2:15pm, the task will run at 2:30, 4:30pm, 6:30pm, 8:30pm, and so on.

30.3.3 Scheduling Per Day and Time

If you select daily task type, 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.

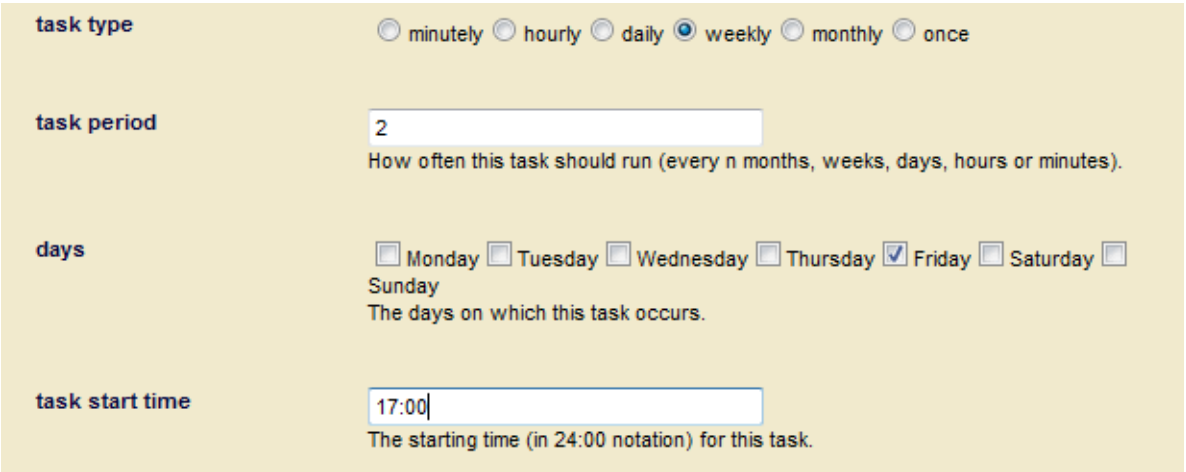
For example, to invoke the module every three days at 12:00pm, enter:



30.3.4 Scheduling Per Week, Day, and Time

If you select weekly task type, 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.

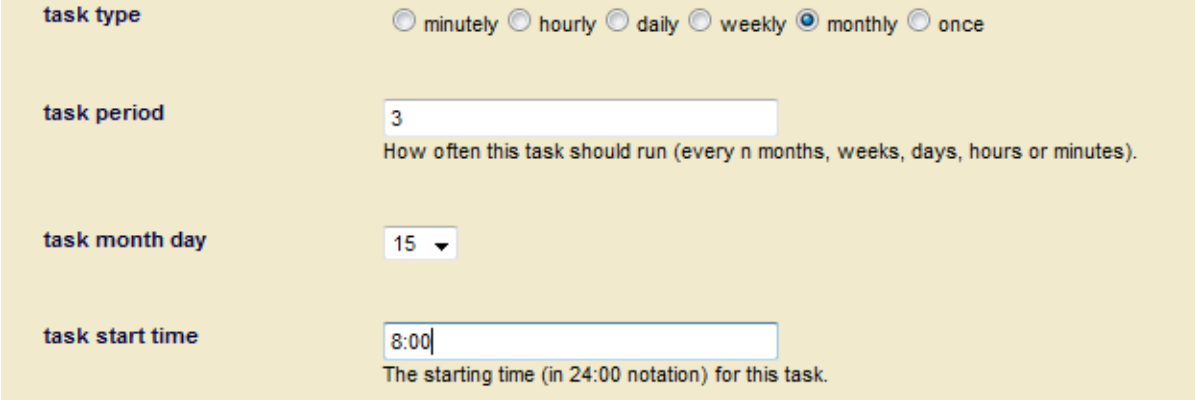
For example, to invoke the module every other week, on Friday, at 5:00pm, enter:



30.3.5 Scheduling Per Month, Day, and Time

If you select monthly task type, 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.

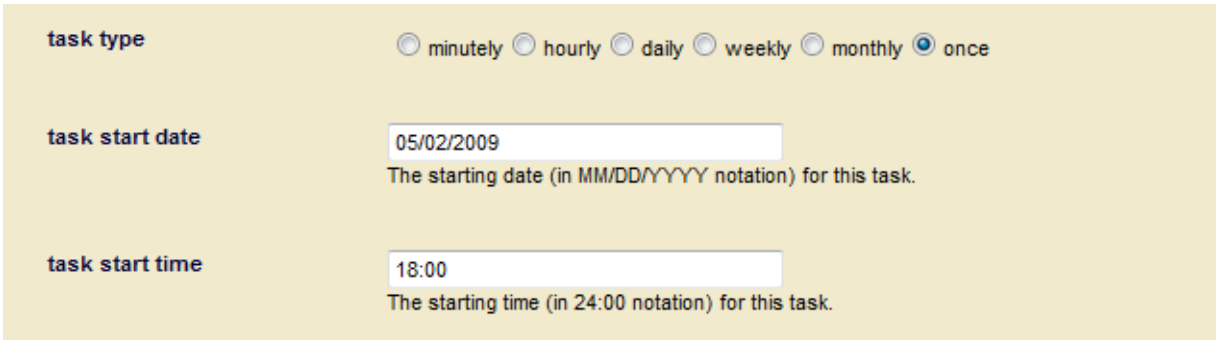
For example, to invoke the module every three months, on the 15th day of the month, at 8:00am, enter:



30.3.6 Scheduling One Invocation on a Calendar Date and Time

If you select once task type, specify the calendar day (month/day/year) and time (in 24:00 notation) of the invocation.

For example, to invoke the module on May 2, 2009 at 6:00pm, enter:



31.0 Using the Configuration Manager

The MarkLogic Server Configuration Manager provides a read-only interface to the Admin Interface and a tool for saving and restoring configuration settings. This chapter includes the following sections:

- Configuration Manager Overview
- Security Considerations
- Accessing the Configuration Manager
- Viewing Configurations
- Searching for a Configuration Setting
- Editing Configuration Settings
- Exporting and Importing Configurations
- Applying Imported Configuration Settings

31.1 Configuration Manager Overview

The Configuration Manager allows you to view the configuration settings for MarkLogic Server resources. A *resource* is a MarkLogic Server object, such as a database, forest, App Server, group or host.

Use the Configuration Manager to:

- Allow non-admin users read-only access to configuration settings for databases, forests, groups, hosts, and App Servers.
- Easily search for resources and configuration settings.
- Safely review settings in read-only mode, then jump directly to the resource in the Admin Interface to modify the settings. (Administrative privileges are required to modify settings).
- Save resource configurations as XML inside a zip folder.
- Import previously saved resource configurations. Importing a configuration allows you to compare versions and update configuration settings.
- View data available through the Management REST API.

For details about the Management REST API, see Using the Management API in the *Monitoring MarkLogic Guide*.

> **Note:** The Packaging REST API has changed for MarkLogic 7. Applications written using the MarkLogic 6 Packaging REST API (v1) must be rewritten to work with the MarkLogic 7 Packaging REST API (v2).

31.2 Security Considerations

To access the Configuration Manager page, users must have the role `manage-user`. The role does not grant any privileges to modify configuration settings. Users with the `manage-user` role may:

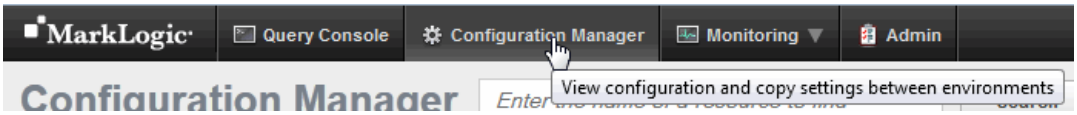
- Access the Configuration Manager pages to view resources and settings.
- Export configurations to an ZIP file (with the exception of modules databases).
- Import previously saved configurations into the Configuration Manager to view or compare them.

The `manage-user` role does not grant privileges to edit configuration settings, apply configuration changes from an imported configuration, or to export modules databases. To export the content of modules databases, you must have the `manage-admin` role. However, you won't be able to see modules that you don't have permissions to access. To install (Apply imported) packages, you must to have the `admin` privilege.

The `manage-internal` role is for MarkLogic Server internal use only. Do not assign this role to any users. Assigning this role to users would grant them privileges on the system that you typically do not want them to have.

31.3 Accessing the Configuration Manager

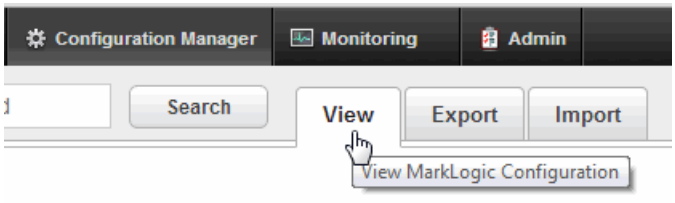
To access the Configuration Manager, navigate to the URL http://yourhostname:8002/nav/ or click on the Configuration Manager tab from any application service page.



Note: If the application does not appear, you may not have sufficient privileges. To make full use of the Configuration Manager, you must have the `manage-admin` security role. See "Security Considerations" on page 398.

31.4 Viewing Configurations

To view the configuration settings for your MarkLogic Server resources, click on the View tab.



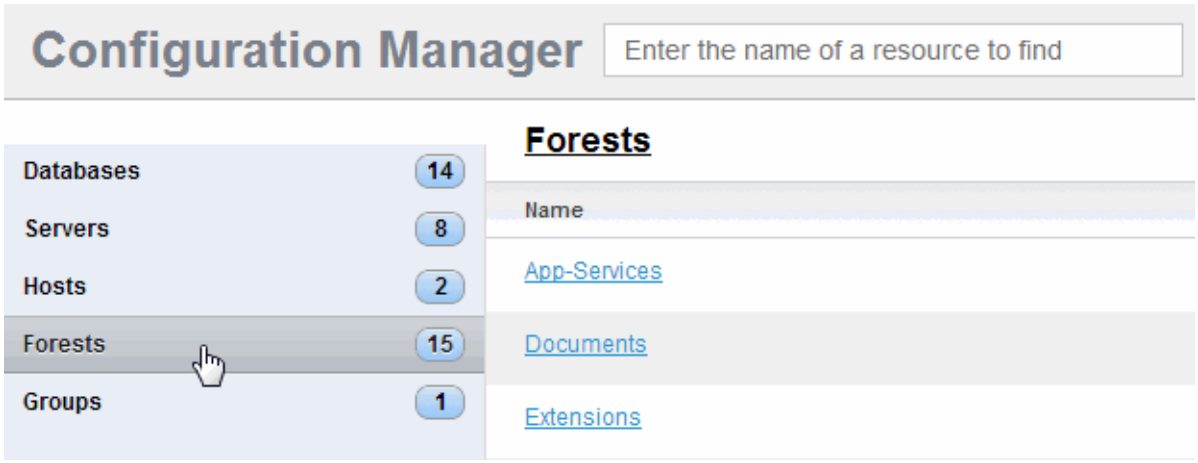
The Configuration Manager provides two methods for locating the configuration settings for a particular resource:

- Browsing Resource Configurations
- Searching for a Resource

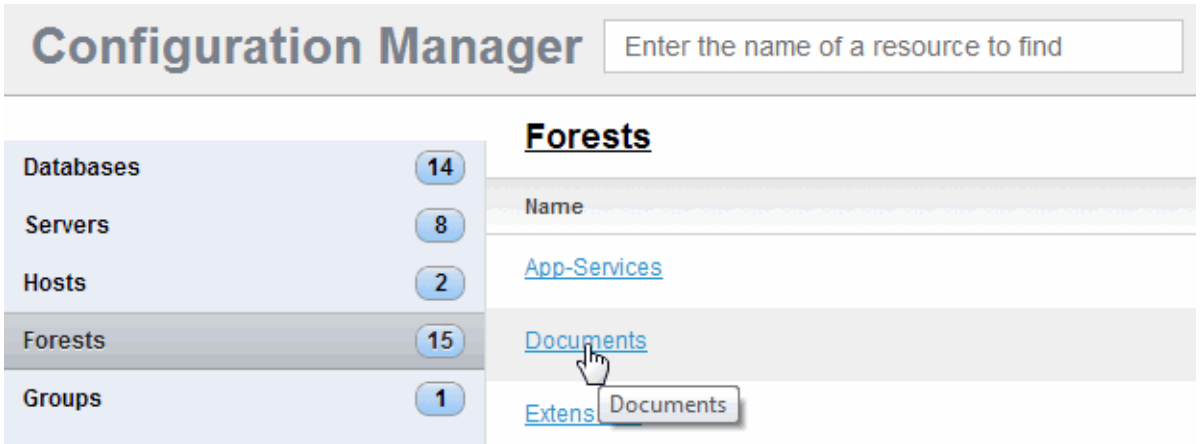
31.4.1 Browsing Resource Configurations

To find a resource and view its configuration by browsing:

1. Click on the resource category in the resource list on the left side of the Configuration Manager. The list of resources in the selected category appears. The number to the right of the resource category name indicates how many of that resource are present:

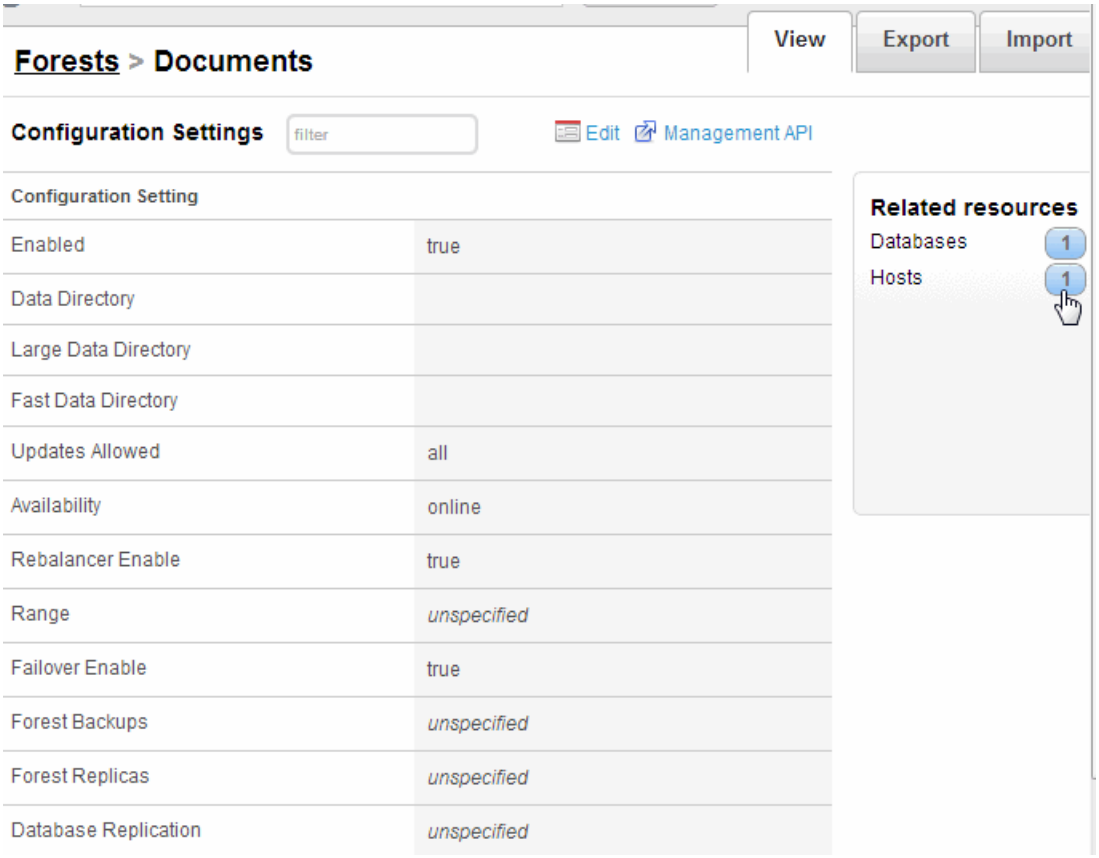


2. Click on the name of a resource to display the configuration:

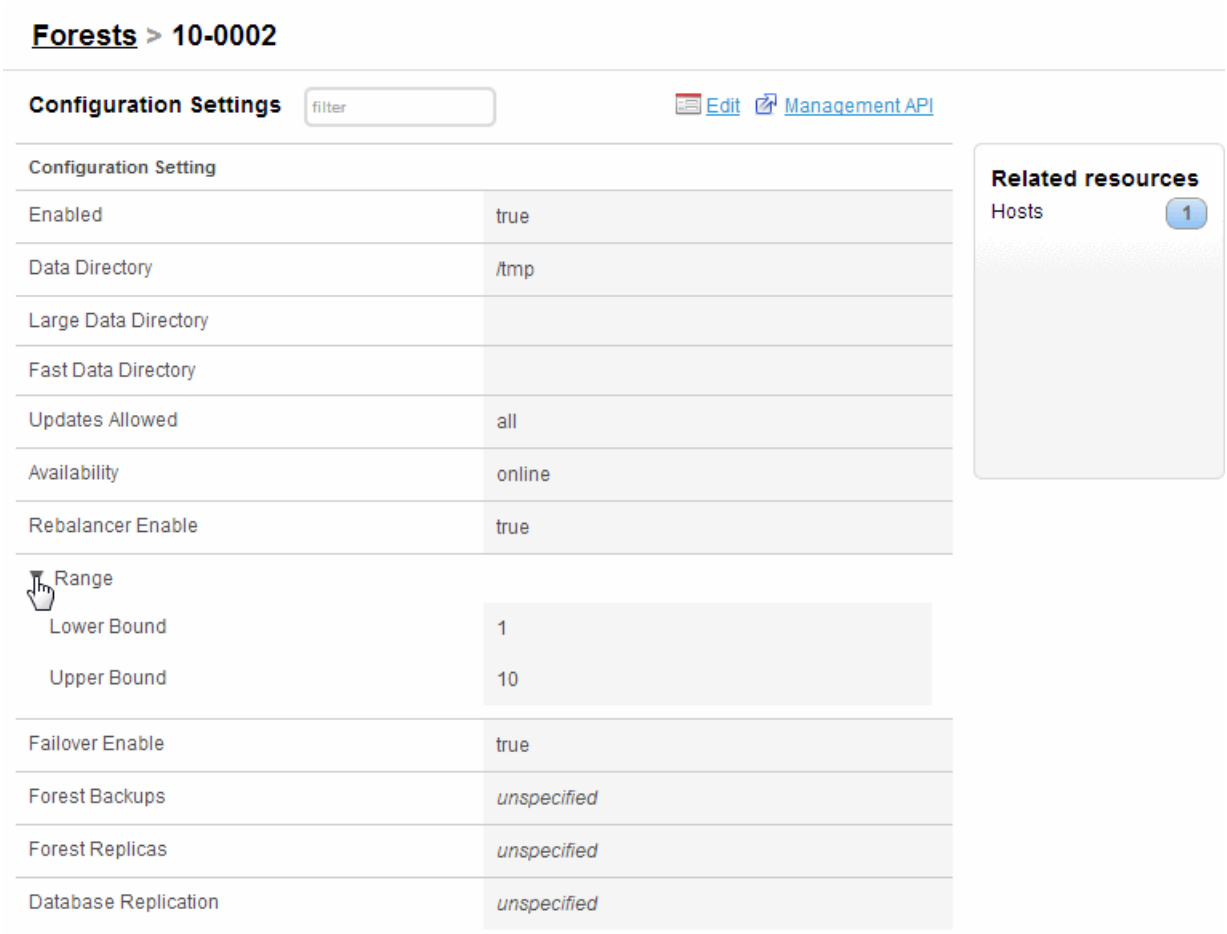


3. The Configuration Settings page displays the settings for the selected resource. In the right-hand frame, the resources that are related to the resource are also listed. You can click on a related resource to view its configuration.

   Note: For forests with partitions, a Partitions link will be included under related resources.



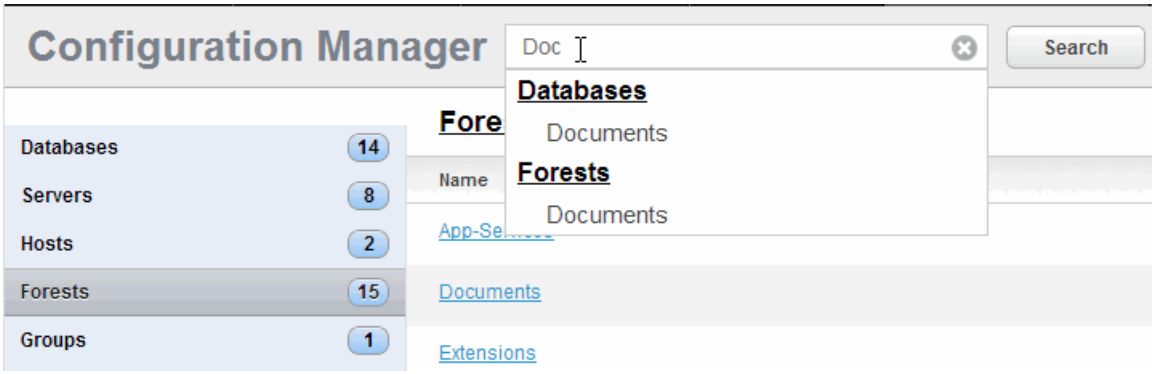
4. Some of the configuration settings are collapsed into containers. These are identified by an arrow preceding the container name. You can click on the arrow to expand the container to display the details. For example, to view the forest range settings, you click on the arrow next to the Range container, as shown below:



31.4.2 Searching for a Resource

To search for a resource by name, use the search box at the top of the page. As you type, the Configuration Manager suggests matching resources.

1. Click in the search box and begin typing the name of a resource. A dropdown of suggested search matches appears:



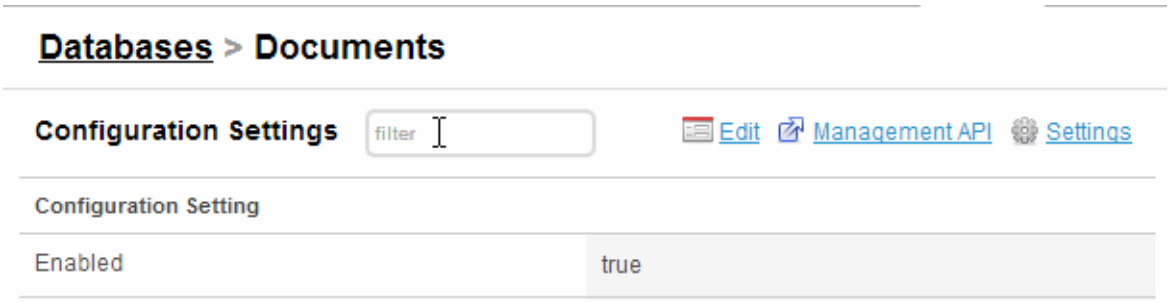
2. Click on a search suggestion in the dropdown list to bring up that configuration, or click the Search button to display a page of matching resources from which to select.

31.5 Searching for a Configuration Setting

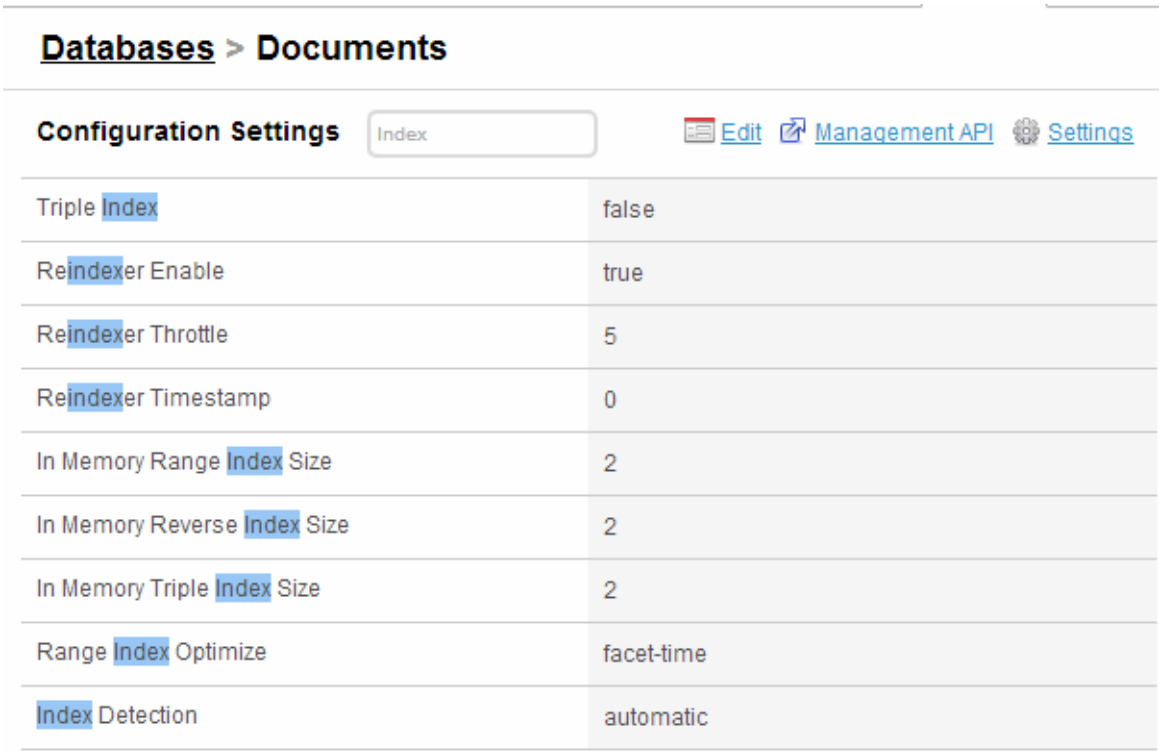
You may search within a resource configuration for a setting name or value:

1. Navigate to the resource you wish to search. See "Viewing Configurations" on page 398.

2. Click in the filter box just above the settings.



3. Begin typing any part of a setting name or value. The configuration settings are filtered as you type. The matching text in each setting is highlighted.



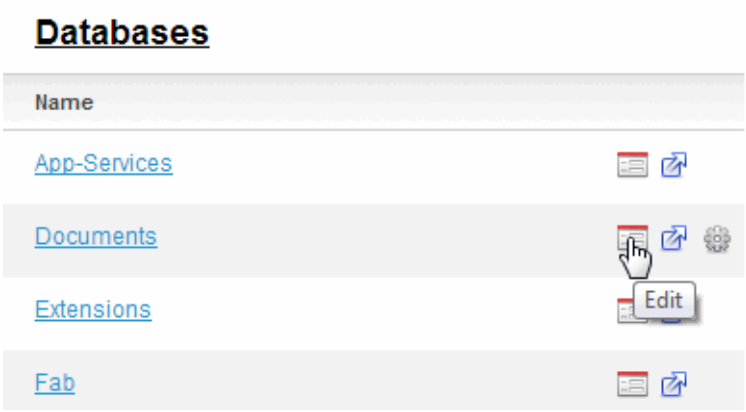
31.6 Editing Configuration Settings

Use the Edit feature to open the Admin UI at a particular resource and modify the settings. You may edit resource settings from a resource category list page or from the settings page for a resource.

Note: To edit configuration settings, you must have administrative privileges.

To edit configuration settings for a resource:

1. From a resource category list, click the edit icon to the right of the resource name. The Admin Interface opens to that resource.



2. MarkLogic Server includes a REST Management API for collecting resource monitoring and management data, as described in the Using the Management API chapter in the *Monitoring MarkLogic Guide*. Using the Configuration Manager, you can easily view the data available for a resource through the REST Management API as HTML, XML, or JSON. Though the information available in the Configuration Manager overlaps with the data available through the Management API, the Management API exposes additional data, such as status information. You can view the resource settings in the Management API by clicking on the Management API icon:

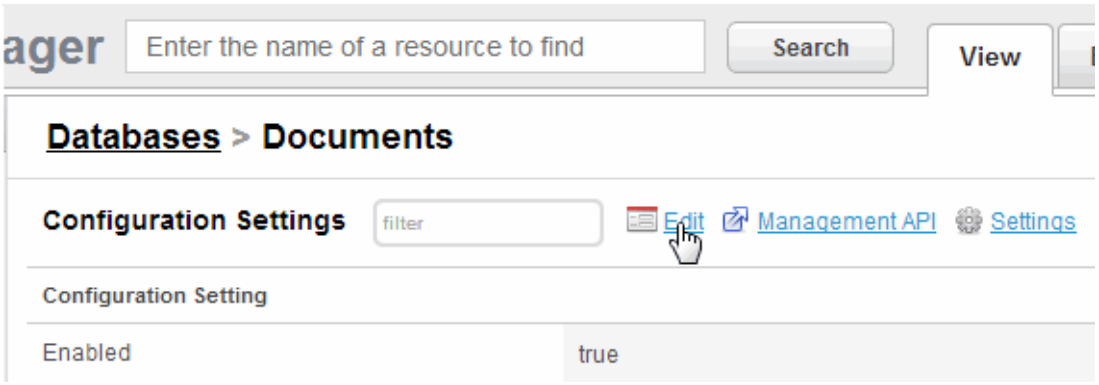


3. If the resource can be managed using Information Studio, an additional Setting icon appears that links to the configuration page for the resource in Information Studio:



For details on Information Studio, see the *Information Studio Developer's Guide*.

4. Links to the Admin Interface, Management API, and Information Studio are also available on the resource Configuration Settings page. For example, the Edit icon at the top of a database Configuration Settings page opens the configuration page for that database in the Admin Interface.



31.7 Exporting and Importing Configurations

The Configuration Manager allows you to save configuration settings in a zip file, and then later import them to compare configurations and apply updates.

The Configuration Manager provides the following capabilities:

- Exporting a Configuration
- Importing a Configuration
- Comparing Imported Configuration with Current Configuration

Note: Security settings, such as SSL and External Security (LDAP and Kerberos) configurations, cannot be imported from an exported configuration file. If your exported configuration includes a server configured with SSL and/or External Security, you must reconfigure these security settings on the server after importing it to the new host.

Note: You can import a configuration saved in MarkLogic 6 into MarkLogic 7. However, you cannot import a configuration saved in MarkLogic 7 into MarkLogic 6.

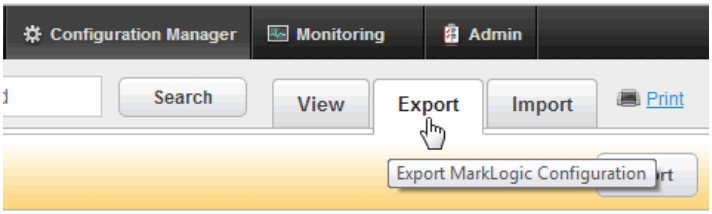
31.7.1 Exporting a Configuration

Exporting resource configurations saves them to a zip file.You may import these configurations back in to compare, review, and apply updates.

> **Note:** You can only export App Server and Database configurations. You cannot export Host, Forest, or Group configurations. In addition, you cannot export an App Server or Database that has more than 200 characters in its name.

The following procedure describes how to export a configuration.

1. Navigate to the Configuration Manager and select the Export tab.



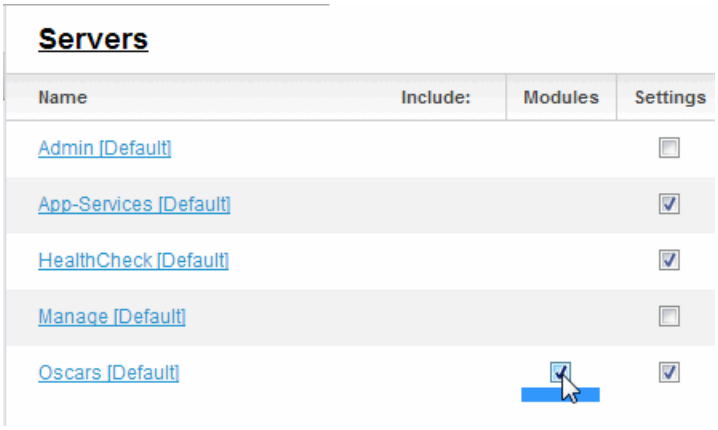
2. Select the App Servers and/or Databases for which you want to export the configurations. You can select all of the Databases and or Servers by checking the category name in the left-hand frame. For example, to select all of the Databases, do the following:



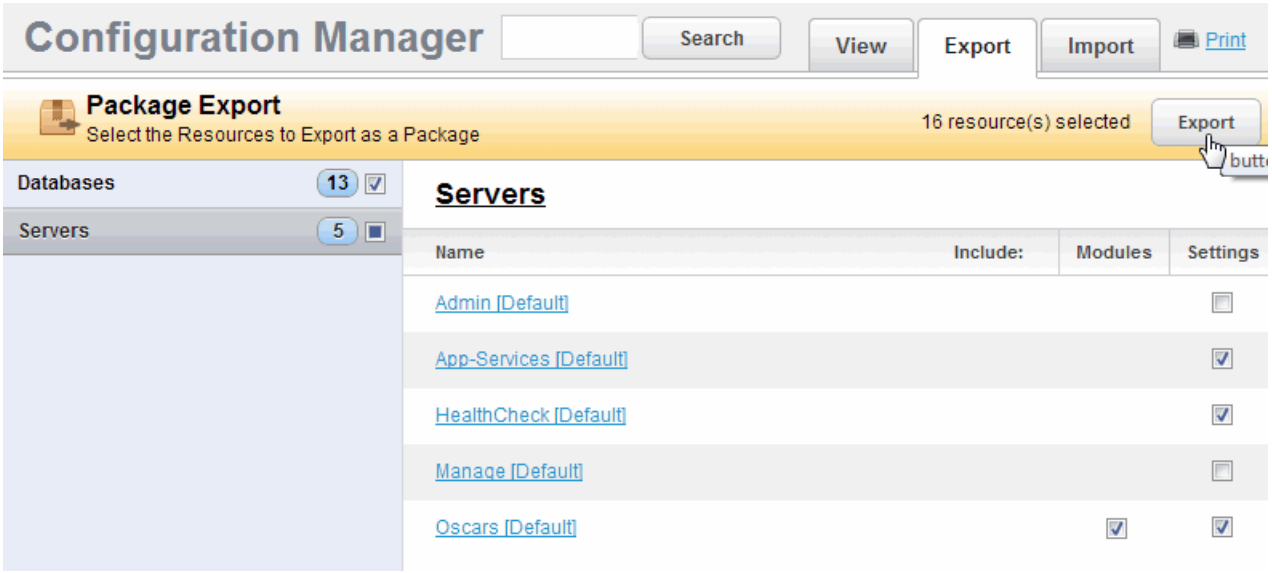
> **Note:** A colored-in check box indicates that some, but not all, of the resources in that category have been selected. You can clear all of the selected resources by clicking on the check box until it's clear.

3. If an App Server makes use of a modules database, an additional check box appears in the Modules column. Check this box to export the modules database along with the App Server configuration.

   Note: You must have the correct roles to export and import a modules database, as described in "Security Considerations" on page 398.



4. After you have selected all of the configurations to be exported, click Export.



A zip file with a name format of `package{id}.zip` will be downloaded to your browser's download directory. A unique id is produced and added to the name of each exported zip file.

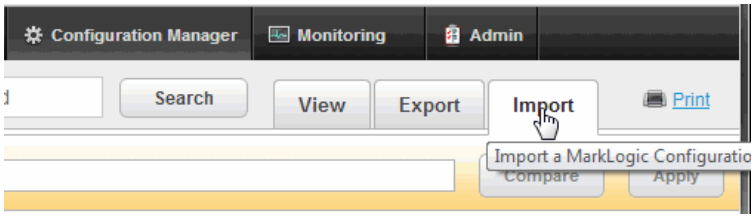
31.7.2 Importing a Configuration

Use the Import feature of the Configuration Manager to upload and apply previously exported configuration packages.

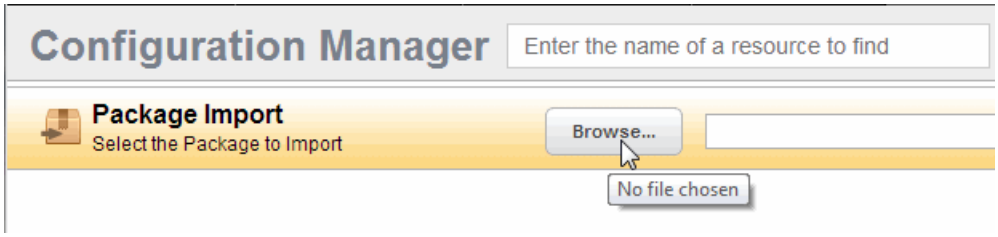
Importing a configuration loads the saved settings into the Configuration Manager. No automatic configuration changes occur. Once you import a configuration (or set of configurations), you may compare the settings with existing configurations and optionally apply configuration changes, as described in "Comparing Imported Configuration with Current Configuration" on page 409.

The following procedure describes how to import a configuration.

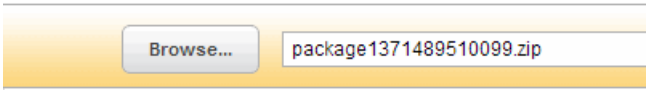
1. Navigate to the Configuration Manager and select the Import tab.



2. Click Browse and navigate to the location of the exported zip file.



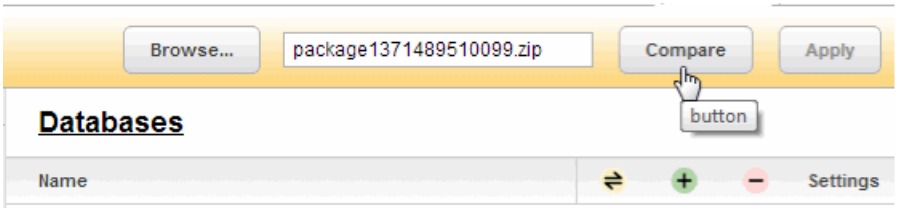
3. A pop-up directory window appears. Double click on the package name and the package appears in the field next to the Browse button.



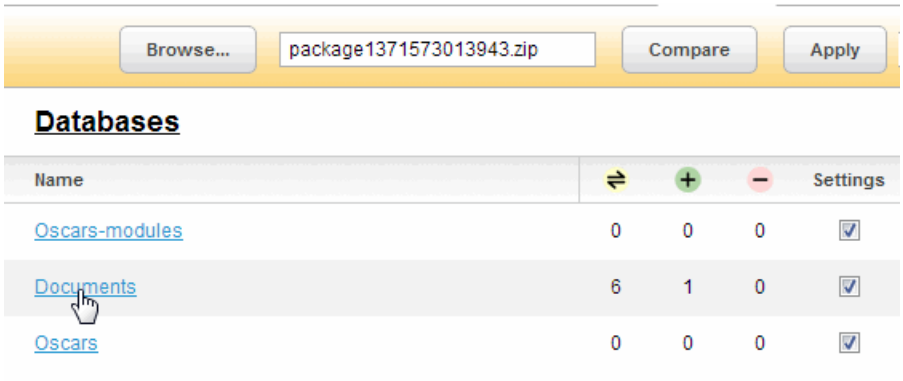
31.7.3 Comparing Imported Configuration with Current Configuration

This section describes how to compare the imported configuration package with your current configuration and determine how the settings are to be applied to your current configuration.

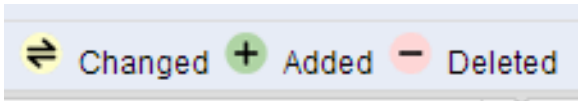
1. After importing a configuration, you can click the Compare button to compare the imported configuration with you current configuration.



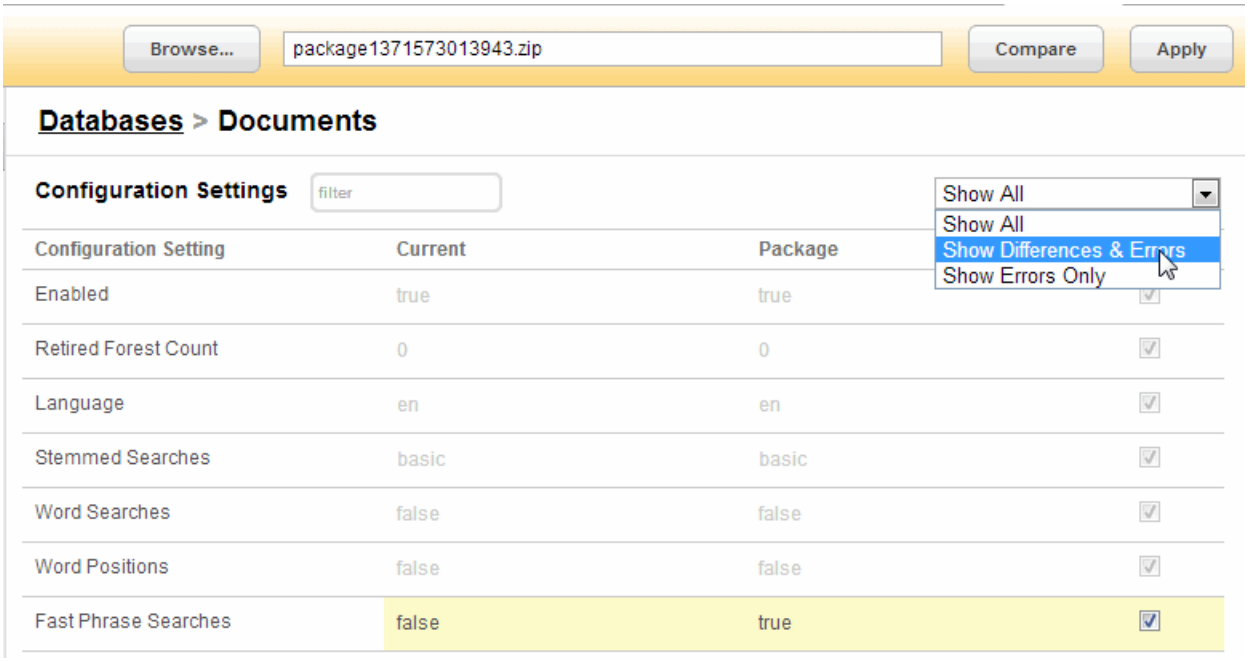
2. You will see a summary of the differences between your current configuration and the imported configuration. To view details of the differences for a resource, click on the resource, as shown for the Documents database below.



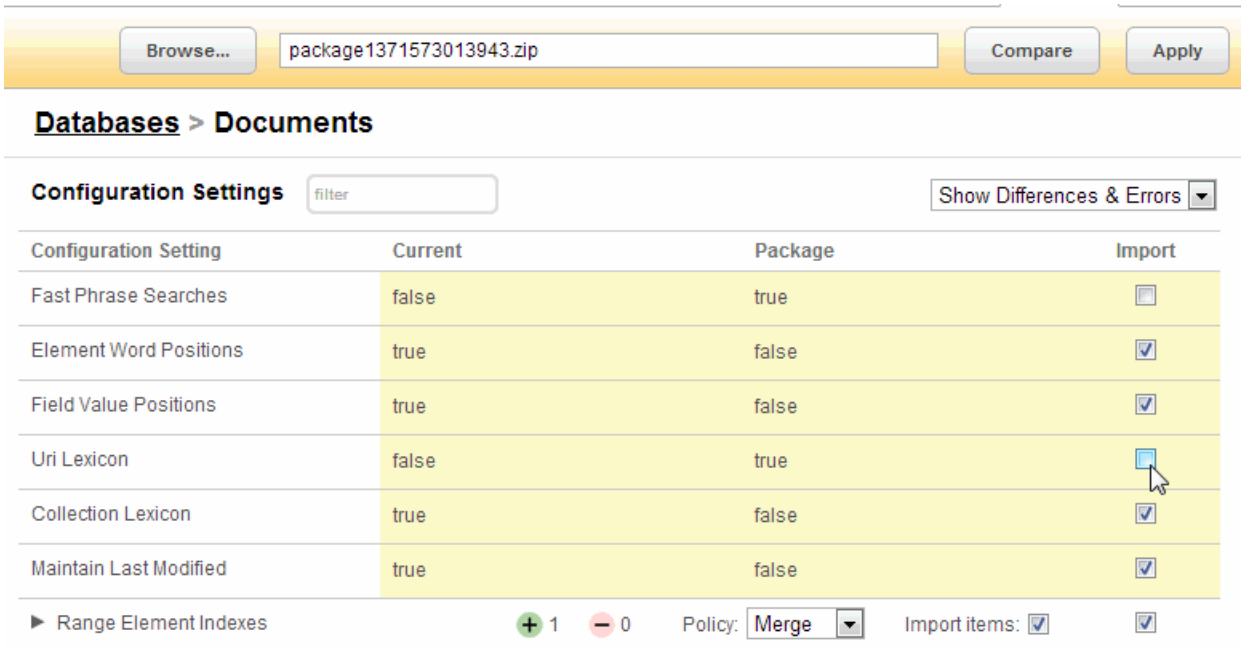
The meaning of the Settings symbols are shown in the legend at the bottom left-hand part of the page.



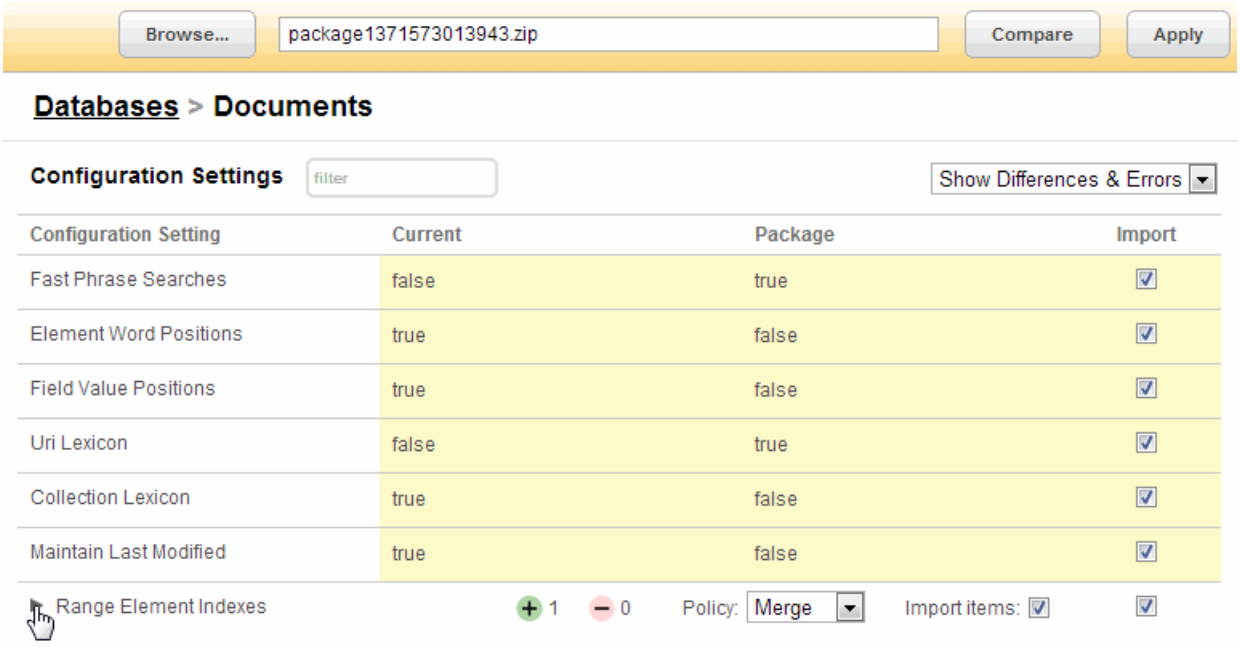
3. The resource Configuration Settings page displays the settings for both the Current and Package configurations and highlights any differences and errors, if present. To view only the differences and errors, select Show Differences & Errors from the pull-down menu. To view only the errors, select Show Errors Only.



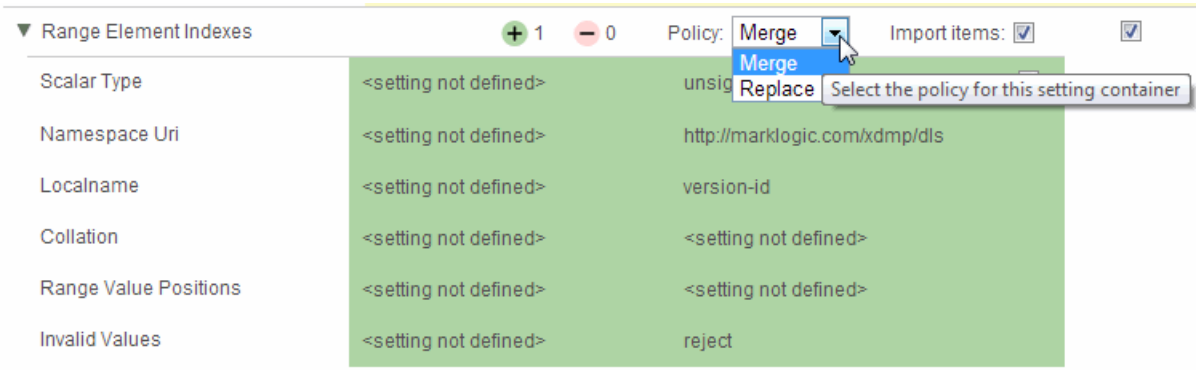
4. By default, all of the packaged settings are imported to your current configuration. If you want to keep a setting in your current configuration that conflicts with the packaged configuration, click on the setting's Import box and uncheck to deselect the setting.



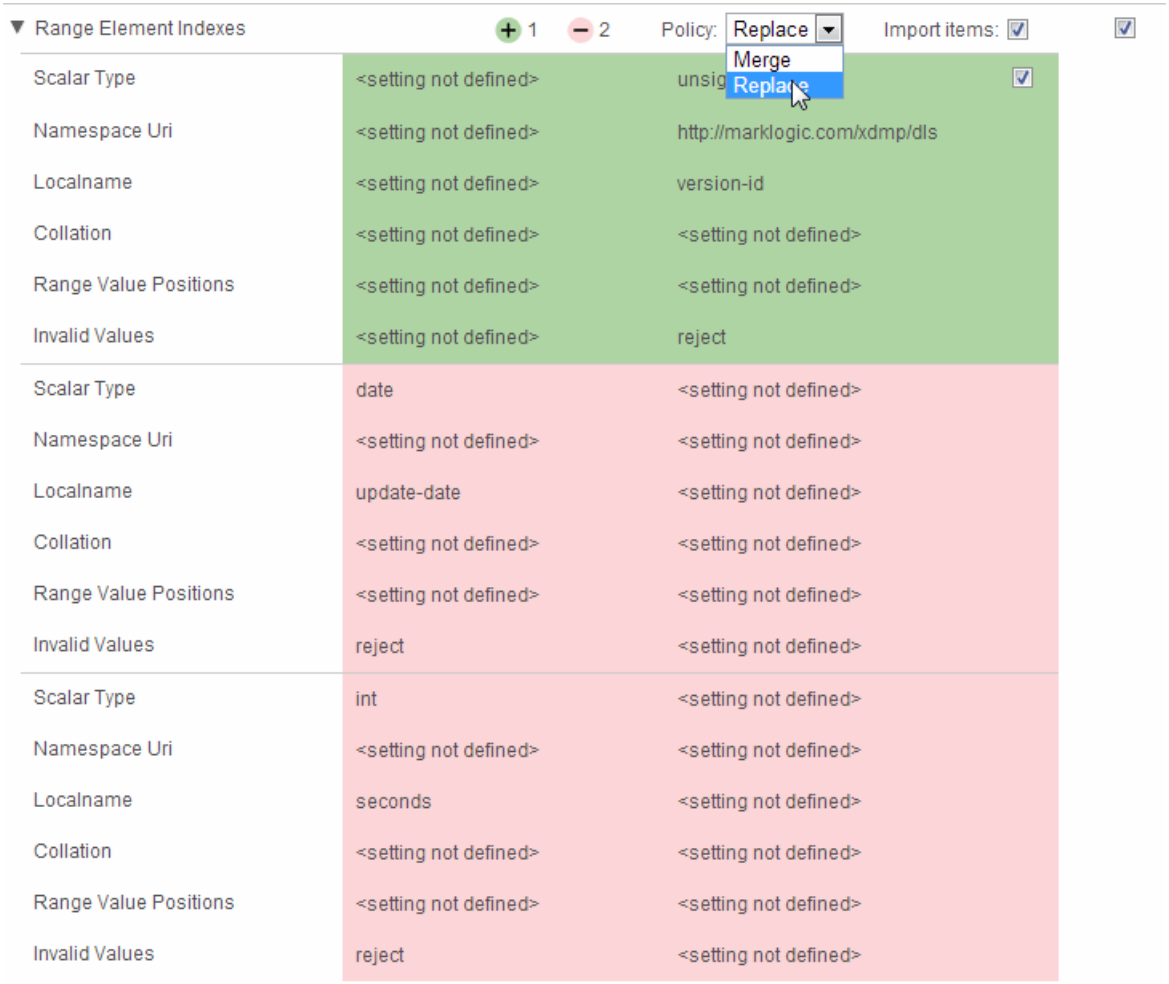
5. Some of the configuration settings are collapsed into containers. These are identified by an arrow preceding the container name. You can click on the arrow to expand the container to display the details. For example, to view the differences between the range element index settings, you click on the arrow next to the Range Element Indexes container, as shown below:



6. You can set how the container settings are to be imported into your current configuration. Select the Merge policy if you want to import only additional settings. This policy ensures that only new settings are imported and no settings are removed from your current configuration.

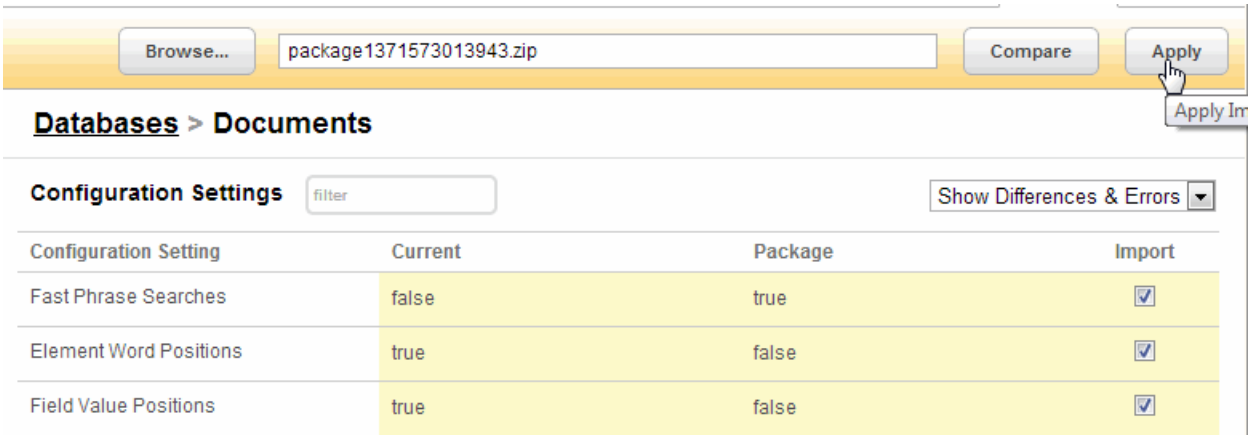


7. If there are settings in your current configuration that are missing from the imported configuration, they are displayed when you select the Replace policy. The Replace policy will remove these settings from your current configuration.

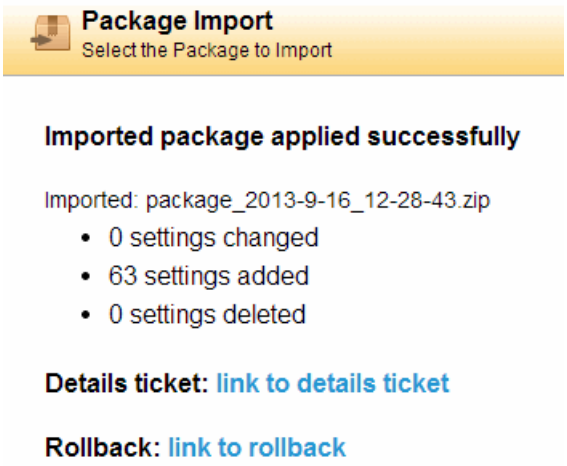


31.8 Applying Imported Configuration Settings

Once you have selected the settings to be used to modify your current configuration, click the Apply button:



If the Apply Import operation is successful, a summary count of the imported settings is displayed. The import operation generated a ticket that provides the details of the imported resources. You can view this ticket by clicking on Link to Details Ticket. Should you wish to roll back the import operation to restore your previous configuration, click on Link to Rollback.



Note: Rolling back an import operation does not remove the imported forests.

32.0 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 UI.

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.

The tables below show the "hot" or "cold" status for adding objects, changing configuration parameters, and dropping objects for the following object types:

- Groups
- HTTP, ODBC, XDBC, and WebDAV Servers
- Databases
- Hosts
- Forests
- Mimetypes
- Security

32.1 Groups

Add Object	Change Configuration Parameters	Delete Object
Hot	The following group parameters are hot: > 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: > list cache size > compressed tree cache size > expanded tree cache size Adding and dropping hosts from groups is cold for that host.	Hot

32.2 HTTP, ODBC, XDBC, and WebDAV Servers

Add Object	Change Configuration Parameters	Delete Object
Hot	The following App Server parameters are hot: > 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 The following App Server parameters are cold for all hosts in the group defining the HTTP, ODBC, XDBC, or WebDAV Server: > port > address > backlog > threads > ssl enabled	Cold

32.3 Databases

Add Object	Change Configuration Parameters	Delete Object
Hot	Parameters changes are hot	Hot

32.4 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.

32.5 Forests

Add Object	Change Configuration Parameters	Delete Object
Hot	Parameter changes are hot. Backup is hot. Restore, clear and drop are hot	Hot

32.6 Mimetypes

Add Object	Change Configuration Parameters	Delete Object
Cold	Parameter changes are cold.	Cold

32.7 Security

Add Object	Change Configuration Parameters	Delete Object
Hot	Parameter changes are hot.	Hot

33.0 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
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 use the Admin Interface	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
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, xdmp:document-load, xdmp:load
app-builder	http://marklogic.com/xdmp/privileges/app-builder	privilege to use the Application Builder UI	
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

Name	Action URI	Description	Protects Function
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
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, xdmp: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, xdmp:xa-complete
count-builtins	http://marklogic.com/xdmp/privileges/counts	privilege to run xdmp:forest-counts	xdmp:forest-counts
create-amp	http://marklogic.com/xdmp/privileges/create-amp	privilege to create an amp	sec:create-amp
create-domain	http://marklogic.com/xdmp/privileges/create-domain	privilege to create an domain	dom:create
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
debug-any-requests	http://marklogic.com/xdmp/privileges/debug-any-requests	privilege to debug all requests from any user	debug built-ins

Name	Action URI	Description	Protects Function
debug-my-requests	http://marklogic.com/xdmp/privileges/debug-my-requests	privilege to debug your own requests	debug buit-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

Name	Action URI	Description	Protects Function
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-compartments	http://marklogic.com/xdmp/privileges/get-compartments	privilege to get a the compartments	sec:get-compartments
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-user-names	http://marklogic.com/xdmp/privileges/get-user-names	privilege to get user names	sec:get-user-names

Name	Action URI	Description	Protects Function
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

Name	Action URI	Description	Protects Function
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.
infostudio	http://marklogic.com/xdmp/privileges/infostudio	privilege to use Information Studio	Information Studio functions
manage	http://marklogic.com/xdmp/privileges/manage	privilege to run the Management API	package:add-database, package:add-appserver, All of the resource addresses in the Management API
odbc:eval	http://marklogic.com/xdmp/privileges/odbc-eval	privilege to execute eval statements from odbc	xdmp:eval

Name	Action URI	Description	Protects Function
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
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

Name	Action URI	Description	Protects Function
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

Name	Action URI	Description	Protects Function
plugin-server-fields	http://marklogic.com/xdmp/privileges/plugin-server-fields	privilege to use the plugin API	Used by the plugin API
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
qconsole	http://marklogic.com/xdmp/privileges/qconsole	privilege to run Query Console	
remove-amp	http://marklogic.com/xdmp/privileges/remove-amp	privilege to remove an amp from the security database	sec:remove-amp
remove-privilege	http://marklogic.com/xdmp/privileges/remove-privilege	privilege to remove a privilege from the security database	sec:remove-privilege
remove-role	http://marklogic.com/xdmp/privileges/remove-role	privilege to remove a role from the security database	sec:remove-role

Name	Action URI	Description	Protects Function
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
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-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-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-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

Name	Action URI	Description	Protects Function
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-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
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-builtins	privilege to access the status built-ins	status built-ins
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
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

Name	Action URI	Description	Protects Function
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-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-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-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)
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

Name	Action URI	Description	Protects Function
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

Name	Action URI	Description	Protects Function
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.	alert-admin
xdmp:alert-internal	http://marklogic.com/xdmp/privileges/xdmp-alert-internal	privilege used by the Alerting API functions.	alert-internal
xdmp:alert-user	http://marklogic.com/xdmp/privileges/xdmp-alert-user	privilege to perform user-level Alerting functions.	alert-user, 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-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

Name	Action URI	Description	Protects Function
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
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-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:delete-cluster-config-file	http://marklogic.com/xdmp/privileges/xdmp-delete-cluster-config-file	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: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

Name	Action URI	Description	Protects Function
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
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-file	http://marklogic.com/xdmp/privileges/xdmp-filesystem-file	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-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-delete	http://marklogic.com/xdmp/privileges/xdmp-forest-delete	privilege to perform admin functions	admin built-ins
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
xdmp:forest-rollback	http://marklogic.com/xdmp/privileges/xdmp-forest-rollback	privilege to perform admin functions	admin built-ins

Name	Action URI	Description	Protects Function
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-name	http://marklogic.com/xdmp/privileges/xdmp-get-session-field-name	privilege to get session field names	xdmp:get-session-field-names
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

Name	Action URI	Description	Protects Function
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
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-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-partitons	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

Name	Action URI	Description	Protects Function
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
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-host-config-file	http://marklogic.com/xdmp/privileges/xdmp-read-host-config-file	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: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 perfrom 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

Name	Action URI	Description	Protects Function
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
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: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: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

Name	Action URI	Description	Protects Function
xdmp:write-cluster-config-file	http://marklogic.com/xdmp/privileges/xdmp-write-cluster-config-file	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
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
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

34.0 Appendix C: Pre-defined Roles

The following roles are pre-defined in every installation of MarkLogic Server. To give a user execute privileges listed for each pre-defined role, you may 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.

The following are the pre-built roles in MarkLogic Server:

- admin
- admin-builtins
- admin-module-internal
- alert-admin
- alert-execution
- alert-internal
- alert-user
- app-builder
- app-builder-internal
- app-user
- application-plugin-registrar
- appservices-internal
- cpf-restart
- custom-dictionary-admin
- custom-dictionary-user
- dls-admin
- dls-internal
- dls-user
- domain-management
- filesystem-access
- flexrep-admin
- flexrep-internal
- flexrep-user
- hadoop-internal
- hadoop-user-all
- hadoop-user-read

- hadoop-user-write
- infostudio-admin-internal
- infostudio-internal
- infostudio-user
- manage-admin
- manage-admin-internal
- manage-internal
- manage-user
- merge
- network-access
- pipeline-execution
- pipeline-management
- pki
- plugin-internal
- qconsole-internal
- qconsole-user
- rest-admin
- rest-admin-internal
- rest-extension-user
- rest-internal
- rest-reader
- rest-writer-internal
- rest-writer
- rest-reader-internal
- search-internal
- security
- trigger-management
- welcome-internal
- xa
- xa-admin
- xinclude

34.1 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 follow proper administrative procedures.

34.2 admin-builtins

The `admin-builtins` role has the execute privileges to call the admin built-in functions. The execute privileges given to the admin-builtins role are:

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

Name	Action URI
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
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.

34.3 admin-module-internal

The `admin-module-internal` role is used internally by the Admin Library Module and should not be assigned to any user. For details, see Scripting Administrative Tasks in MarkLogic Server in the *Scripting Administrative Tasks Guide*.

34.4 alert-admin

The `alert-admin` role is used for administrators of an alerting application. For details, see the Creating Alerting Applications chapter of the *Search Developer's Guide*.

34.5 alert-execution

The `alert-execution` 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 chapter of the *Search Developer's Guide*.

34.6 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 chapter of the *Search Developer's Guide*.

34.7 alert-user

The `alert-user` role is used by users of an alerting application. For details, see the Creating Alerting Applications chapter of the *Search Developer's Guide*.

34.8 app-builder

The `app-builder` role provides the privileges needed to run Application Builder. Application Builder performs many administrative tasks on MarkLogic Server (for example, creating databases and App Servers), and this role provides the privileges to perform those tasks. While the privileges are minimized to the needed functions and to amped functions, it still allows users with the role to create these resources on MarkLogic Server, and therefore, only trusted users (users who are assumed to be non-hostile, appropriately trained, and follow proper administrative procedures) should be granted the `app-builder` role. Assign the `app-builder` role to users who are allowed to generate applications with Application Builder.

For details, see the *Application Builder Developer's Guide*.

34.9 app-builder-internal

The `app-builder-internal` role is used by Application Builder to amp certain functions that Application Builder performs. You should not explicitly grant the `app-builder-internal` role to any user; it is only for internal use by Application Builder.

For details, see the *Application Builder Developer's Guide*.

34.10 app-user

The `app-user` role is a minimally privileged role that is needed to run any application that Application Builder generates. You must grant this role to all users who are allowed to run the generated application.

For details, see the *Application Builder Developer's Guide*.

34.11 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

Name	Action URI
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

34.12 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.

34.13 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.

34.14 custom-dictionary-admin

The `custom-dictionary-admin` role is to perform adminstative functions (for writing dictionaries in the configuration) in the custom dictionary API.

34.15 custom-dictionary-user

The `custom-dictionary-user` role is to perform user functions (for reading dictionaries in the configuration) in the custom dictionary API.

34.16 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 the Library Services Applications chapter in the *Application Developer's Guide*.

34.17 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 the Library Services Applications chapter in the *Application Developer's Guide*.

34.18 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 the Library Services Applications chapter in the *Application Developer's Guide*.

34.19 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
domain-management	Read
domain-management	Update

34.20 filesystem-access

The `filesystem-access` role has the privileges to access the file system. The execute privileges given to the `filesystem-access` role are:

Name	Action URI
xdmp:document-get	http://marklogic.com/xdmp/privileges/xdmp-document-get
xdmp:document-load	http://marklogic.com/xdmp/privileges/xdmp-document-load
xdmp:get	http://marklogic.com/xdmp/privileges/xdmp-get
xdmp:load	http://marklogic.com/xdmp/privileges/xdmp-load
xdmp:save	http://marklogic.com/xdmp/privileges/xdmp-save

There are no default permissions associated with the filesystem-access role.

34.21 flexrep-admin

The `flexrep-admin` role is required to configure replication.

34.22 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.

34.23 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.

34.24 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.

34.25 hadoop-user-all

The `hadoop-user-all` role combines the privileges of `hadoop-user-read` and `hadoop-user-write`.

34.26 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
hadoop-user-read	http://marklogic.com/xdmp/privileges/hadoop-user-read
xdbc:eval	http://marklogic.com/xdmp/privileges/xdbc-eval
xdbc:eval-in	http://marklogic.com/xdmp/privileges/xdbc-eval-in
xdmp:value	http://marklogic.com/xdmp/privileges/xdmp-value
xdmp:with-namespaces	http://marklogic.com/xdmp/privileges/xdmp-with-namespace

34.27 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
any-uri	http://marklogic.com/xdmp/privileges/any-uri
hadoop-user-write	http://marklogic.com/xdmp/privileges/hadoop-user-write
unprotected-collections	http://marklogic.com/xdmp/privileges/unprotected-collections
xdbc:eval	http://marklogic.com/xdmp/privileges/xdbc-eval
xdbc:insert-in	http://marklogic.com/xdmp/privileges/xdbc-insert-in
xdmp:with-namespaces	http://marklogic.com/xdmp/privileges/xdmp-with-namespace

34.28 infostudio-admin-internal

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.

For more details, see the Controlling Access to Information Studio chapter in the *Information Studio Developer's Guide*.

34.29 infostudio-internal

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.

34.30 infostudio-user

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`

34.31 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 execute privileges given to the `manage-admin` role are:

Name	Action URI
manage	http://marklogic.com/xdmp/privileges/manage
manage-admin	http://marklogic.com/xdmp/privileges/manage-admin
ts:database-create-sub-database	http://marklogic.com/xdmp/privileges/database-create-sub-database
ts:database-create-super-database	http://marklogic.com/xdmp/privileges/database-create-super-database
ts:database-delete-sub-database	http://marklogic.com/xdmp/privileges/database-delete-sub-database
ts:database-delete-sub-database	http://marklogic.com/xdmp/privileges/database-delete-super-database
ts:database-partitions	http://marklogic.com/xdmp/privileges/database-partitions
ts:forest-combine	http://marklogic.com/xdmp/privileges/forest-combine
ts:forest-migrate	http://marklogic.com/xdmp/privileges/forest-migrate
ts:partition-create	http://marklogic.com/xdmp/privileges/partition-create
ts:partition-delete	http://marklogic.com/xdmp/privileges/partition-delete
ts:partition-forests	http://marklogic.com/xdmp/privileges/partition-forests
ts:partition-migrate	http://marklogic.com/xdmp/privileges/partition-migrate
ts:partition-resize	http://marklogic.com/xdmp/privileges/partition-resize
ts:partition-set-availability	http://marklogic.com/xdmp/privileges/partition-set-availability
ts:partition-set-updates-allowed	http://marklogic.com/xdmp/privileges/partition-set-updates-allowed
ts:partition-transfer	http://marklogic.com/xdmp/privileges/partition-transfer

There are no default permissions associated with the `manage-admin` role.

34.32 manage-admin-internal

The `manage-admin-internal` role is used to amp certain functions used by the Configuration Manager and the Management API. You should not explicitly grant the `manage-admin-internal` role to any user; it is only for internal use.

34.33 manage-internal

The `manage-internal` role is used to amp certain functions used by the Configuration Manager. You should not explicitly grant the `manage-internal` role to any user; it is only for internal use.

34.34 manage-user

The `manage-user` role has the privileges related to accessing the Configuration Manager. The execute privileges given to the `merge` role are:

Name	Action URI
manage	http://marklogic.com/xdmp/privileges/manage

There are no default permissions associated with the `manage-user` role.

34.35 merge

The `merge` role has the privileges related to forest merging. The execute privileges given to the `merge` role are:

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.

34.36 network-access

The network-access role has the privileges to run the `xdmp:http-*` functions (`xdmp:http-get`, `xdmp:http-post`, and so on). The execute privileges given to the `network-access` role are:

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

34.37 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* guide.

34.38 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 the following default permissions:

Role	Capability
pipeline-management	Read
pipeline-management	Update

34.39 pki

The `pki` role has the privileges to use the PKI Library functions. For details, see "Configuring SSL on App Servers" on page 78.

34.40 plugin-internal

The `plugin-user` role is used to amp certain functions assocated with plugins. You should not explicitly grant the `plugin-internal` role to any user; it is only for internal use by the plugin API.

34.41 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.

34.42 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 the following execute privileges:

- `qconsole` (`http://marklogic.com/xdmp/privileges/qconsole`)

34.43 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.

34.44 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.

34.45 rest-extension-user

The `rest-extension-user` role enables access to resource service extension methods. .

34.46 rest-internal

The `rest-internal` role is used internally by the REST Library. You should not explicitly grant it to any user or role.

34.47 rest-reader

The `rest-reader` role enables read operations through the MarkLogic REST API, such as retrieving documents and metadata.

34.48 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.

34.49 rest-writer

The `rest-writer` role enables write operations through the MarkLogic REST API, such as creating documents, metadata, or configuration information.

34.50 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.

34.51 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.

34.52 security

The `security` role has the privileges needed to perform security functions. The execute privileges given to the `security` role are:

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

Name	Action URI
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

Name	Action URI
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
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

Default permissions for the `security` role are:

Role	Capability
security	Read
security	Insert
security	Update

34.53 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 the following default permissions:

Role	Capability
trigger-management	Read
trigger-management	Update

34.54 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 the *XCC Developer's Guide*. The `xa` role has the following execute privileges:

Name	Action URI
complete-my-transaction	http://marklogic.com/xdmp/privileges/complete-my-transactions
forget-my-xa-transactions	http://marklogic.com/xdmp/privileges/forget-my-xa-transactions
prepare-my-xa-transactions	http://marklogic.com/xdmp/privileges/prepare-my-xa-transactions
status-builtins	http://marklogic.com/xdmp/privileges/status-builtins
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

34.55 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 the following execute privileges:

Name	Action URI
complete-any-transactions	http://marklogic.com/xdmp/privileges/complete-any-transactions
complete-my-transaction	http://marklogic.com/xdmp/privileges/complete-my-transactions
forget-any-xa-transactions	http://marklogic.com/xdmp/privileges/forget-any-xa-transactions
forget-my-xa-transactions	http://marklogic.com/xdmp/privileges/forget-my-xa-transactions
prepare-any-xa-transactions	http://marklogic.com/xdmp/privileges/prepare-any-xa-transactions
prepare-my-xa-transactions	http://marklogic.com/xdmp/privileges/prepare-my-xa-transactions
status-builtins	http://marklogic.com/xdmp/privileges/status-builtins
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

34.56 welcome-internal

The `welcome-internal` role is a role that use to be used internally by the MarkLogic Server Welcome Page (now removed). You should not explicitly grant it to any user or role.

34.57 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*.

35.0 Technical Support

MarkLogic 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 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 general questions, join the general discussion mailing list, open to all MarkLogic developers.

36.0 Copyright

MarkLogic Server 8.0 and supporting products.

NOTICE

Copyright © 2018 MarkLogic Corporation.

This technology is protected by one or more U.S. Patents 7,127,469, 7,171,404, 7,756,858, 7,962,474, 8,935,267, 8,892,599 and 9,092,507.

All MarkLogic software products are protected by United States and international copyright, patent and other intellectual property laws, and incorporate certain third party libraries and components which are subject to the attributions, terms, conditions and disclaimers found at http://docs.marklogic.com/guide/copyright/legal.

MarkLogic and the MarkLogic logo are trademarks or registered trademarks of MarkLogic Corporation in the United States and other countries. All other trademarks are property of their respective owners.

For all copyright notices, including third-party copyright notices, see the Combined Product Notices for your version of MarkLogic.