MarkLogic Server

Monitoring MarkLogic Guide

MarkLogic 6
September, 2012

Last Revised: 6.0-1, September, 2012

Copyright © 2012 MarkLogic Corporation. All rights reserved.

1.0 Monitoring MarkLogic Server 4
1.1 Overview 4
1.2 Selecting a Monitoring Tool 5
1.3 Monitoring Architecture, a High-level View 6
1.4 Monitoring Tools and Security 6
1.5 Guidelines for Configuring your Monitoring Tools 7
1.5.1 Establish a Performance Baseline 7
1.5.2 Balance Completeness Against Performance 7
1.6 Monitoring Metrics of Interest to MarkLogic Server 8
1.6.1 Does MarkLogic Have Adequate Resources? 8
1.6.2 What is the State of the System Overall? 9
1.6.3 What is Happening on the MarkLogic Server Cluster Now? 10
1.6.4 Are There Signs of a Serious Problem? 12
1.7 Default Monitor Metrics 13

2.0 Using the MarkLogic Server Monitoring Dashboard 16
2.1 Terms Used in this Chapter 16
2.2 Displaying the Monitoring Dashboard 17
2.3 Monitoring Specific Resources 17
2.4 Monitoring Dashboard Sessions 18
2.5 Setting the Sample Interval 18
2.6 Viewing Monitoring Sample Details 19
2.7 Monitoring Query Execution 19
2.8 Monitoring Rates and Loads 21
2.8.1 Overview 22
2.8.2 XDQP Communication 23
2.8.3 Backup/Restore 26
2.9 Monitoring Disk Space 27
2.10 Exporting Monitoring Data 30

3.0 Configuring Nagios to Monitor MarkLogic Server 31
3.1 Terms Used in this Chapter 31
3.2 Overview of the Nagios Plugin Package 32
3.3 Nagios Plugin Requirements 34
3.3.1 Nagios Host Supported Platforms 34
3.3.2 Nagios Host Library Requirements 34
3.4 Installing the Nagios Plugin 36
3.5 Configuring Nagios for use with MarkLogic Server 37
3.5.1 The generate_marklogic_config.pl Script 38
3.5.2 The Monitoring Services File 40
3.5.2.1 Globally Excluding Resources 41
3.5.2.2 Service Definitions 41
3.5.2.3 The check_command Parameter 43
3.5.2.4 Defining and Setting Thresholds and Ranges 47
3.6 Understanding the Generated Object Definition File 50

3.7 Using Nagios 53
3.7.1 Nagios Navigation Panels 53
3.7.2 Host Groups 55
3.7.3 Service Groups 56
3.7.4 Service Status Details for a Resource 57

4.0 Configuring the MarkLogic Smart Plug-in for HP Operations Manager 58
4.1 Overview 58
4.2 Terms Used in this Chapter 60
4.3 Smart Plug-in for HP Operations Manager Requirements 61
4.4 Installing MarkLogic Smart Plugin 61
4.5 Configuring the MarkLogic Smart Plug-in 63
4.5.1 Edit the Configuration File 64
4.5.2 Enabling SSL between the Proxy Server and MarkLogic Monitor Host .64
4.5.3 Add MarkLogic Proxy nodes to the Node Group 67
4.5.4 Deploy MarkLogic Smart Plug-in Instrumentation 69
4.5.5 Launch the MarkLogic SPI Setup Tools 71
4.5.6 Deploy the MarkLogic Policies to the Plugin Proxy 74
4.6 Uninstalling the MarkLogic Smart Plugin 76
4.7 Configuration Files 88
4.7.1 mlspi_config.xml 88
4.7.2 mlspi_urls.xml 90
4.8 MarkLogic Default Policies 92
4.8.1 MarkLogic Cluster 92
4.8.2 MarkLogic Database 93
4.8.3 MarkLogic Application Server 94
4.8.4 MarkLogic Host 95
4.9 Modifying Policies and Creating New Policies 96
4.9.1 Policy Rules 96
4.9.2 Changing a Policy 97
4.9.3 Creating a New Policy 100
4.10 Reconfiguring HP Operations Manager After Changes to MarkLogic Server ...110
4.11 Performance Data Source 111

5.0 Using the Management API 112
5.1 Terms used in this Chapter 112
5.2 Overview of the Management API 113
5.3 Security 114
5.4 Resource Addresses 114
5.5 Obtaining the Options Node for a Resource Address 125
5.6 Specifying the Management API Version 126
5.7 Specifying Parameters in a Resource Address 126
5.7.1 Formatting the Monitor Results 127
5.7.2 Using the property Parameter on Forest Counts 129
5.8 Interpreting the Output 130

6.0 Extending Management API with Plugins 131
6.1 Types of Management API Plugins 131
6.2 Sample Management API Plugins 132
6.3 Management API Plugin Framework 133
6.4 Creating Custom Resource Plugins 135
6.4.1 Custom Resource Capabilities and Function Signatures 135
6.4.2 Example Custom Resource Plugin 137
6.5 Creating Custom View Plugins 140
6.5.1 Custom View Capabilities and Function Signatures 140
6.5.2 Example Custom View Plugin 141
6.6 Creating Custom Format Plugins 142
6.6.1 Custom Format Capabilities and Function Signatures 142
6.6.2 Example Custom Format Plugin 143

7.0 Technical Support 145

Combined Product Notices 146
COPYRIGHT 146
TRADEMARK NOTICE 158

1.0 Monitoring MarkLogic Server

MarkLogic Server provides a rich set of monitoring features that include a pre-configured monitoring dashboard, plugins that allow you to monitor MarkLogic Server with Nagios and HP Operations Manager, and a Management API that allows you to integrate MarkLogic Server with existing monitoring applications or create your own custom monitoring applications.

This chapter includes the following sections:

- Overview
- Selecting a Monitoring Tool
- Monitoring Architecture, a High-level View
- Monitoring Tools and Security
- Guidelines for Configuring your Monitoring Tools
- Monitoring Metrics of Interest to MarkLogic Server
- Default Monitor Metrics

1.1 Overview

In general, you will use a monitoring tool for the following:

- To keep track of the day-to-day operations of your MarkLogic Server environment.
- For initial capacity planning and fine-tuning your MarkLogic Server environment. For details on how to configure your MarkLogic Server cluster, see the *Scalability, Availability, and Failover Guide*.
- To troubleshoot application performance problems. For details on how to troubleshoot and resolve performance issues, see the *Query Performance and Tuning Guide*.
- To troubleshoot application errors and failures.

The monitoring metrics and thresholds of interest will vary depending on your specific hardware/software environment and configuration of your MarkLogic Server cluster. This chapter lists some of the metrics of interest when configuring and troubleshooting MarkLogic Server. However, MarkLogic Server is just one part of your overall environment. The health of your cluster depends on the health of the underlying infrastructure, such as network bandwidth, disk I/O, memory, and CPU.

1.2 Selecting a Monitoring Tool

Though this guide focuses on the tools available from MarkLogic that enable you to monitor MarkLogic Server, it is strongly recommended that you select an enterprise-class monitoring tool that monitors your entire computing environment to gather application, operating system, and network metrics alongside MarkLogic Server metrics.

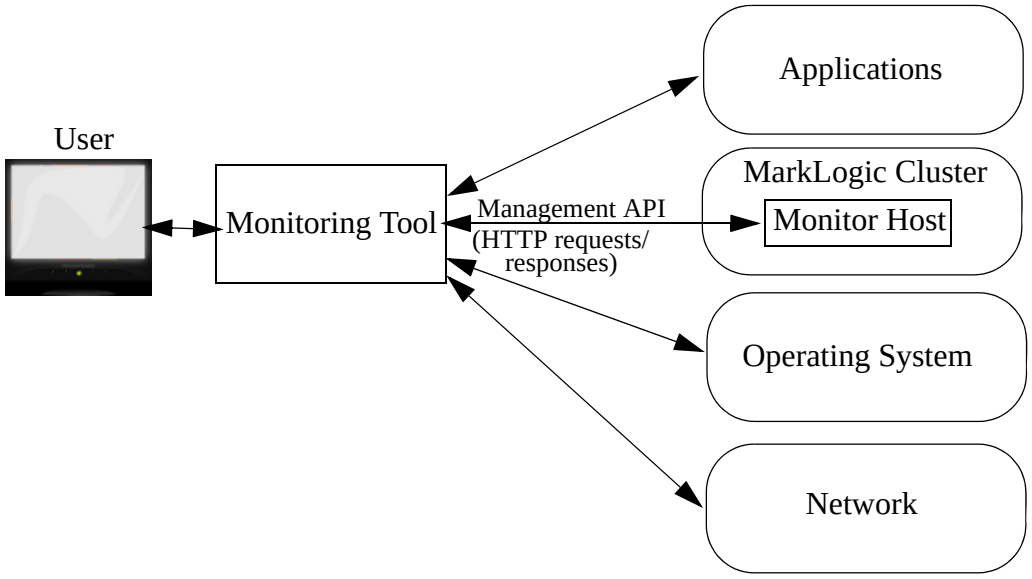
There are many monitoring tools on the market that have key features such as alerting, trending, and log analysis to help you monitor your entire environment. MarkLogic Server includes the following monitoring tools:

- A Monitoring Dashboard that monitors MarkLogic Server. This dashboard is pre-configured to monitor specific MarkLogic Server metrics. For details, see "Using the MarkLogic Server Monitoring Dashboard" on page 16.
- A plugin that enables Nagios to monitor MarkLogic Server. Nagios can be configured to monitor any and all objects in MarkLogic Server and allows you to set alerts on object thresholds. Nagios is freeware and has a large support community and so it is a good default choice if you do not already have a preferred monitoring tool. For details, see "Configuring Nagios to Monitor MarkLogic Server" on page 31.
- A plugin that enables HP Operations Manager to monitor MarkLogic Server. HP Operations Manager can be configured to monitor any and all objects in MarkLogic Server and allows you to set alerts on object thresholds. For details, see "Configuring the MarkLogic Smart Plug-in for HP Operations Manager" on page 58.
- A RESTful Management API that you can use to integrate MarkLogic Server with existing monitoring application or create your own custom monitoring applications. For details, see "Using the Management API" on page 112.

  Note: There is a fundamental difference between how Nagios and HP Operations Manager collect monitoring metrics. Nagios collects each metric with a separate HTTP request and reports a value for all collected metrics, whereas HP Operations Manager collects all of its metrics with a single HTTP request and only reports metrics that have exceeded the specified thresholds.

1.3 Monitoring Architecture, a High-level View

All monitoring tools use a RESTful Management API to communicate with MarkLogic Server. The monitoring tool sends HTTP requests to a monitor host in a MarkLogic cluster. The MarkLogic monitor host gathers the requested information from the cluster and returns it in the form of an HTTP response to the monitoring tool. The Management API is described in “Using the Management API” on page 112.



1.4 Monitoring Tools and Security

To gain access to the monitoring features described in this guide, a user must be assigned the `manage-user` role. Monitoring tools should authenticate as a user with that role. The `manage-user` role is assigned the `http://marklogic.com/xdmp/privileges/manage` execute privilege and provides access to the Management API, `Manage` App Server, and the UI for the Configuration Manager and Monitoring Dashboard. The `manage-user` role also provides read-only access to all of a cluster's configuration and status information, with the exception of the security settings. For details on assigning roles to users, see Users in the *Administrator's Guide*.

If you have enabled SSL on the `Manage` App Server, your URLs must start with HTTPS, rather than HTTP. Additionally, you must have a MarkLogic certificate on your browser, as described in Accessing an SSL-Enabled Server from a Browser or WebDAV Client in the *Administrator's Guide*.

1.5 Guidelines for Configuring your Monitoring Tools

Most monitoring tools, such as Nagios and HP Operations Manager, enable you to set thresholds on specific metrics to alert you when a metric exceeds a pre-specified value.

The topics in this section are:

- Establish a Performance Baseline
- Balance Completeness Against Performance

1.5.1 Establish a Performance Baseline

Many metrics that can help in alerting and troubleshooting are meaningful only if you understand normal patterns of performance. For example, monitoring an App Server for slow queries will require a different threshold on an application that spawns many long-running queries to the task server than on an HTTP App Server where queries are normally in the 100 ms range. Most enterprise-class monitoring tools support data storage to support this type of trend analysis. Developing a starting baseline and tuning it if your application profile changes will yield better results for developing your monitoring strategy.

1.5.2 Balance Completeness Against Performance

The templates provided with the Nagios and HP Operations Manager integrations consist of a mix of MarkLogic metrics that are useful in many situations for problem-solving, performance tuning, and capacity planning.

Collecting and storing monitoring metrics has a performance cost, so you need to balance completeness of desired performance metrics against their cost. The cost of collecting monitoring metrics can differ. In general, the more resources you monitor, the greater the cost. For example, if you have a lot of hosts, server status is going to be more expensive. If you have a lot of forests, database status is going to be more expensive. In most cases, you will use a subset of the available monitoring metrics. And there may be circumstances in which you temporarily monitor certain metrics and, once the issue have been targeted and resolved, you no longer monitor those metrics.

One balancing technique is to measure system performance on a staging environment under heavy load, then enable your monitoring tool and calculate the overhead. You can reduce overhead by reducing collection frequency, reducing the number of metrics collected, or writing a Management API plugin to produce a custom view that pinpoints the specific metrics of interest. Each response from the underlying Management API includes an elapsed time value to help you calculate the relative cost of each response. For details on the Management API, see “Using the Management API” on page 112. For details on how to write a Management API plugin, see “Extending Management API with Plugins” on page 131.

1.6 Monitoring Metrics of Interest to MarkLogic Server

Environments and workloads vary. Each environment will have a unique set of requirements based on variables including cluster configuration, hardware, operating system, patterns of queries and updates, feature sets, and other system components. For example, if replication is not configured in your environment, you can remove templates or policies that monitor that feature.

This section provides a set of guiding questions to help you understand and identify the relevant metrics. The topics in this section are:

- Does MarkLogic Have Adequate Resources?
- What is the State of the System Overall?
- What is Happening on the MarkLogic Server Cluster Now?
- Are There Signs of a Serious Problem?

1.6.1 Does MarkLogic Have Adequate Resources?

MarkLogic Server is designed to fully utilize system resources. Many settings, such as cache sizes, are auto-sized by MarkLogic Server at installation.

Some questions to ask are:

- Does MarkLogic Server have enough resources on the host machine? What processes other than MarkLogic Server are running on the host and what host resources do those processes require? When competing with other processes, MarkLogic Server cannot optimize resource utilization and consequently cannot optimize performance.
- Is there enough disk space for forest data and merges? Merges require at least twice as much free disk space as used by the forest data (3X rule). If a merge runs out of disk space, it will fail.
- Is there enough disk space for the log files reside to log system activity? If there is no space left on the log file device, MarkLogic Server will abort. Also, if there is no disk space available to add messages to the log files, MarkLogic Server will fail to start.
- Is there enough memory for the range indexes? Range index improve performance at the cost of memory and increased load/reindex time. Running out of memory for range indexes may result in undesirable memory swapping that severely impacts performance.
- Is swap space configured correctly? At query time, MarkLogic Server makes use of both memory and swap space. If there is not enough of either, the query can fail with SVC-MEMALLOC messages. For details on configuring swap memory, see Tuning Query Performance in MarkLogic Server in the *Query Performance and Tuning Guide*.
- How many hosts are in the cluster? How are the hosts configured as evaluator and data nodes? How are the hosts organized into groups? For details on configuring MarkLogic Server clusters, see Clustering in MarkLogic Server in the *Scalability, Availability, and Failover Guide*.

- What applications use resource-intensive features, such as CPF, replication, and point-in-time recovery? Are the hardware, software, and network resources available and configured to most efficiently support such applications?

1.6.2 What is the State of the System Overall?

Many problems that impact MarkLogic Server originate outside of MarkLogic Server. Consider the health of your overall environment.

Some questions to ask are:

- How efficiently is CPU being used? How much CPU capacity exists at different time slices? What is the execution speed of the current read and write tasks? Can I optimize queries or choose a better time to batch load?
- How efficiently is I/O being used? What amount of data is currently being read from or written to disk? Are there any I/O bottlenecks?
- Is there enough free disk space for each file system?
- Are there any errors or warnings appearing in the logs for the operating system, MarkLogic Server, and applications?
- What is the current state of the network?
- Are there any serious errors in the system log files? Your monitor tool, or an auxiliary tool such as Splunk, should monitor your system logs and report on any detected errors.

1.6.3 What is Happening on the MarkLogic Server Cluster Now?

When you suspect an error or performance problem originates from MarkLogic Server, some questions to ask are:

- Are all of the hosts in the cluster online? Are all of the App Servers enabled? In what a states are the forests?
- What are the patterns of queries and updates? Do they appear to be evenly distributed across the hosts in the cluster?
- Are there any long-running queries? Longer than usual query execution times may indicate a bottleneck, such as a slow host or problems with XDQP communication between hosts. Other possible problems include increased loads following a failover or more than the usual number of total requests.
- Is there an increase in the number of outstanding requests? A consistent increase in the total number of outstanding requests may indicate the need to add more capacity and/or load balance. Decreases in total requests may indicate some "upstream" problem that needs to be addressed.
- What is the I/O rates and loads pattern? In this context, *rates* refers to amount of data applications are currently reading from or writing to MarkLogic Server databases (throughput) and *loads* refers to the execution time of the current read and write requests, which includes the time requests spend in the wait queue when maximum throughput is achieved.

  Under normal circumstances you will see loads go up as rates go up. As the workload (number of queries and updates) increases, a steadily high rates value indicates the maximum database throughput has been achieved. When this occurs, you can expect to see increasing loads that reflects the additional time requests are spending in the wait queue. As the workload decreases, you can expect to see decreasing loads that reflects less requests in the wait queue.

  If, while the workload is steady, rates decrease and loads increase, something is probably taking away I/O bandwidth from the database. This may indicate that MarkLogic Server has started a background task, such as a merge operation or some process outside of MarkLogic Server is taking away I/O bandwidth.

- What is the journal and save write rates and loads pattern? During a merge, you should see the rates for journal and save writes decrease and the loads increase. Once the merge is done, journal and save writes rates should increase and the loads should decrease. If no merge is taking place, then a process outside of MarkLogic Server may be taking away I/O bandwidth.
- What is the XDQP rates and loads pattern? In this context, *rates* refers to amount of data hosts are currently reading from or writing to other hosts and *loads* refers to the execution time of the current read and write requests, including those in the wait queue. A decrease in rates and an increase in loads may indicate that there is network problem.

- What are the cache hit/miss rates? Lots of cache hits means not having to read fragments off disk, so there is less I/O load. An increasing cache miss rate may indicate a need to increase the cache size, write queries that take advantage of indexes to reduce the frequency of disk reads, or adjust the fragment size to better match that of the queried data.
- How many concurrent updates and reads are in progress? An increase of both updates and reads may indicate that there are queries that are doing too many updates and reads concurrently. The potential problem is lock contention between the updates and reads on the same fragments, which degrades performance.
- How many database merges are in progress? Merges require both I/O and disk resources. If too many database merges are taking place at the same time, it may be necessary to coordinate merges by creating a merge policy or establishing merge blackout periods, as described in Understanding and Controlling Database Merges in the *Administrator's Guide*.
- How many reindexes are in progress? Database reindexing is periodically done automatically in the background by MarkLogic Server and requires both CPU and disk resources. If there are too many reindexing processes going on at the same time, you may need to adjust when reindexing is done for particular databases, as described in Text Indexing in the *Administrator's Guide*.
- How many backups and/or restores are in progress? Backup and restore processes can impact the performance of applications and other background tasks in MarkLogic Server, such as merges and indexing. Backups with point-in-time recovery enabled have an even greater impact on performance. If backup and/or restore processes are impacting system performance, it may be necessary to reschedule them, as described in Backing Up and Restoring a Database in the *Administrator's Guide*.

1.6.4 Are There Signs of a Serious Problem?

If you are encountering a serious problem in which MarkLogic Server is unable to effectively service your applications, some questions to ask are:

- Did MarkLogic Server abort or fail to start? This may indicate that there not enough disk space for the log files on the log file device. If this is the cause, you will need to either add more disk space or free up enough disk space for the log files.
- Is an application unable to update data in MarkLogic Server? This may indicate that you have exceeded the 64-stand limit for a forest. This could be the result of running out of merge space or that merges are suppressed.
- Are queries failing with SVC-MEMALLOC messages? This indicates that there is not enough memory or swap space. You may need to add memory or reconfigure your swap memory, as described in Tuning Query Performance in MarkLogic Server in the *Query Performance and Tuning Guide*
- Are there any forests in the async replicating state? This state indicates that a primary forest is asynchronously catching up to its replica forest after a failover or that a new replica forest was added to a primary forest that already contains content. If a forest has failed over, see Scenarios that Cause a Forest to Fail Over in the *Scalability, Availability, and Failover Guide* for possible causes.
- Are there any serious messages in the error logs? The various log levels are described in Understanding the Log Levels in the *Administrator's Guide*. All log messages at the error level and higher should be investigated, whereas lower-level messages, such as warnings and debug messages are mostly informational. Log messages that indicate a particularly serious problem include:
    - Repeated server restart messages. Possible causes include a corrupted forest, segmentation faults, or some problem with the host's operating system.
    - XDQP disconnect. Possible causes include an XDQP timeout or a network failure.
    - Forest unmounted. Possible causes include the forest is disabled, it has run out of merge space, or the forest data is corrupted.
    - SVC-* errors. These are system-level errors that result from timeouts, socket connect issues, lack of memory, and so on.
    - XDMP-BAD errors. These indicate serious internal error conditions that shouldn't happen. Look at the error text for details and the logs for context and contact MarkLogic Support.

1.7 Default Monitor Metrics

The following table lists the metrics that are monitored by the Nagios default templates and HP Operations Manager default policies, along with their default frequencies and thresholds. If the default frequency differs between HP Operations Manager and Nagios, the differences are flagged by (HP) for HP Operations Manage or (Ng) for Nagios. Some metrics do not have a default warning or critical threshold because these thresholds are dependent on your specific deployment of MarkLogic Server.

For details on how to define the threshold values of these metrics, see either "Configuring Nagios to Monitor MarkLogic Server" on page 31 or "Configuring the MarkLogic Smart Plug-in for HP Operations Manager" on page 58.

Metric	Resource	Default Frequency	Default Warning?	Default Critical?	Note
database-count	Cluster	1 hr	current-count +/-1		Signals change
server-count	Cluster	1 hr	current-count +/-1		Signals change
host-count	Cluster	1 hr	current-count +/-1		Signals change
foreign-cluster-count	Cluster	1 hr	current-count +/-1		Signals change
is-bootstrapped	Foreign Cluster	10 min		0 (false)	Database Replication: Are all boostrap hosts bootstrapped?
long-running requests (total-requests)	Server	1 min	>0	>=10	
request-rate	Server	1 min			
expanded-tree-cache-hits	Server	1 min			
expanded-tree-cache-misses	Server	1 min			
query-count	Server	1 min			

Metric	Resource	Default Frequency	Default Warning?	Default Critical?	Note
update-count	Server	1 min			
state	Database	1 min		“unavailable”	
failed-masters	Database	1 min (HP) 10 min (Ng)		>=1	Failover: Down masters
async-replicating	Database	1 min (HP) 10 min (Ng)		>=1	Failover: Forests are catching up
database-replication-active	Database	1 min (HP) 10 min (Ng)		0 (false)	Database Replication
foreign-forests-lag-exceeded	Database	1 min (HP) 10 min (Ng)	1 (true)		Database Replication
backup-count	Database	1 min			
compressed-tree-cache-hit-rate	Database	1 min			
compressed-tree-cache-miss-rate	Database	1 min			
documents	Database	60 min			
list-cache-hit-rate	Database	1 min			list-cache-hit-rate
list-cache-miss-rate	Database	1 min			list-cache-miss-rate
load-detail/*	Database	1 min			
rate-detail/*	Database	1 min			
merge-count	Database	1 min			
on-disk-size	Database	1 min (HP) 10 min (Ng)			
reindex-count	Database	1 min (HP) 10 min (Ng)			
restore-count	Database	1 min (HP) 10 min (Ng)			

Metric	Resource	Default Frequency	Default Warning?	Default Critical?	Note
state-not-open	Host	1 min		>=1	
max-stands-per-forest	Host	1 min	>=35	>=50	
min-capacity	Host	1 min (HP) 10 min (Ng)	<=15	<=10	
online	Host	1 min		“false”	
compressed-tree-cache-hit-rate	Host	1 min			
compressed-tree-cache-miss-rate	Host	1 min			
list-cache-hit-rate	Host	1 min			
list-cache-miss-rate	Host	1 min			
load-detail/*	Host	1 min			
rate-detail/*	Host	1 min			

2.0 Using the MarkLogic Server Monitoring Dashboard

This chapter describes how to use the Monitoring Dashboard. The Monitoring Dashboard provides task-based views of MarkLogic Server performance metrics in real time. The Monitoring Dashboard is intended to be used alongside the status pages in the Admin Interface and other monitoring tools that monitor application and operating system performance metrics.

The topics in this chapter are:

- Terms Used in this Chapter
- Displaying the Monitoring Dashboard
- Monitoring Specific Resources
- Monitoring Dashboard Sessions
- Setting the Sample Interval
- Viewing Monitoring Sample Details
- Monitoring Query Execution
- Monitoring Rates and Loads
- Monitoring Disk Space
- Exporting Monitoring Data

2.1 Terms Used in this Chapter

The following terms are used in this chapter:

- A *Monitoring Session* is the timeframe since the dashboard page was last refreshed. For example, if you navigate from the Query Execution page to the Rates and Loads page, you have ended the Query Execution session and started the Rates and Loads session.
- A *Monitoring Sample* is a bit of information captured during a refresh interval on a graph. For example, one of the candlesticks captured in the Query Execution graph is a single sample.

2.2 Displaying the Monitoring Dashboard

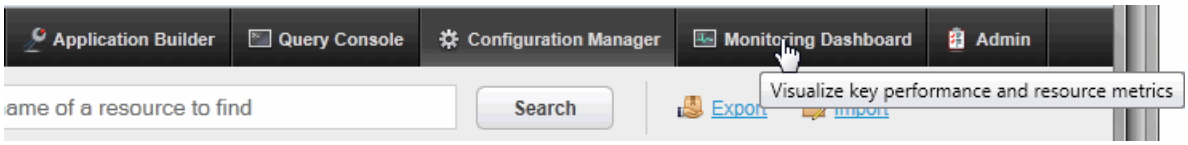
You can display the Monitoring Dashboard by doing the following:

1. Open a browser and enter the URL:

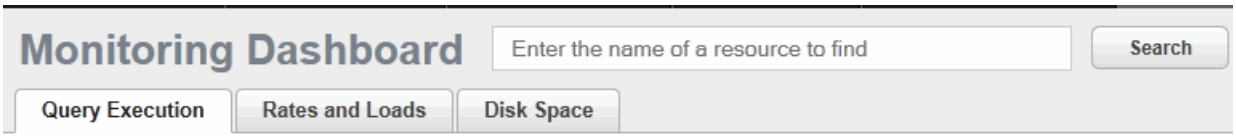
   ```
   http://monitor-host:8002/
   ```

   where *monitor-host* is a host in the cluster you want to monitor

2. At the top of the page, click on Monitoring Dashboard:

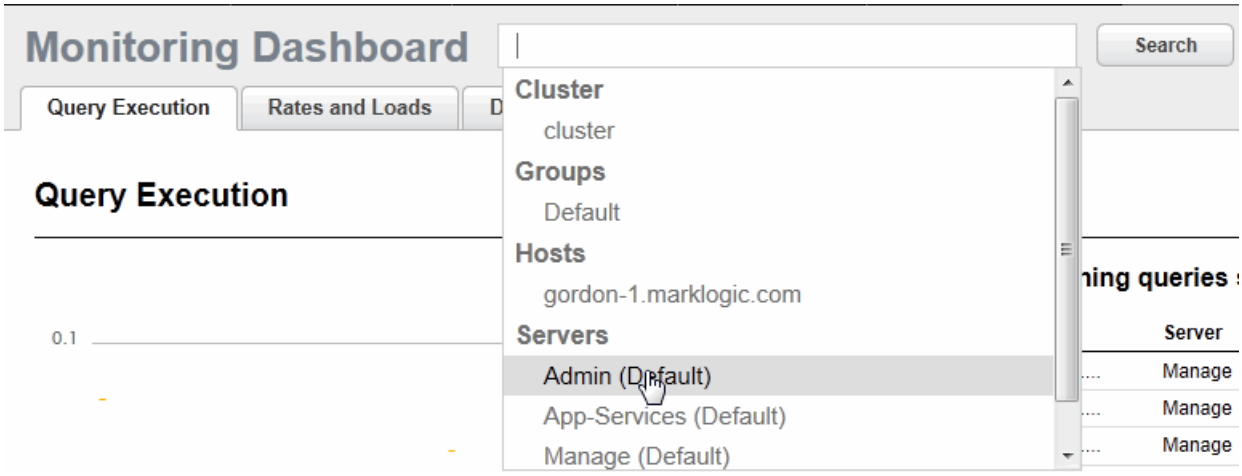


3. The Monitoring Dashboard page appears. From the default Monitoring Dashboard page, you can navigate to any of the pages described in this chapter.



2.3 Monitoring Specific Resources

By default the Monitoring Dashboard monitors the entire cluster. You can use the Search box to select a specific resource to monitor. Clicking on the search field produces a pull-down menu in which you can locate the resource. Alternatively, you can directly locate a resource by entering the name of the resource in the search field.



2.4 Monitoring Dashboard Sessions

Each time you navigate to a new Dashboard page, you end the current monitoring session and begin a new one. The monitoring data from the previous session is lost from that point on. If you want to maintain multiple Dashboard sessions, you can open each page in a separate browser tab or window.

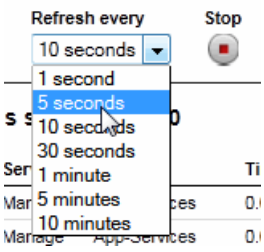
You can freeze the monitoring data for a Dashboard page by clicking on the Stop button in the upper right-hand portion of the page and restart the data by pressing Start. When you stop a page, you will lose any monitoring data between the time the page is stopped and the time it is restarted. If you have multiple Dashboard pages open, the sessions continue on the other pages; so stopping the monitoring data on one page will not stop the data on the other pages. When you start the stopped page, its session will resume at the current timestamp.



2.5 Setting the Sample Interval

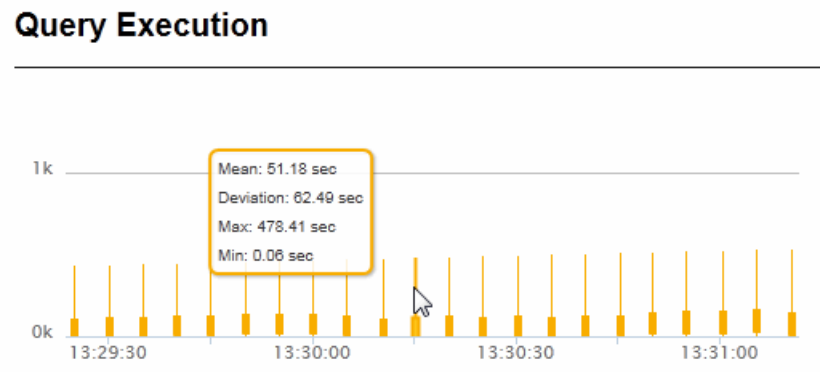
The sample interval specifies the frequency in which the selected resource is monitored. By default, the sample interval is every 10 seconds. Use the Refresh pull-down menu to set the sample interval from anything between once every 1 second to every 10 minutes.

If you have multiple Dashboard pages open in separate tabs or windows, changing the sample interval on one page will not change the interval on the other pages. However, if you switch between pages in the same browser tab or window, the interval will be the same for all pages.



2.6 Viewing Monitoring Sample Details

You can hover your mouse on any monitoring sample to view the details of the sample. For example, to view the details of a query execution sample, hover on the bar graphic as shown below.

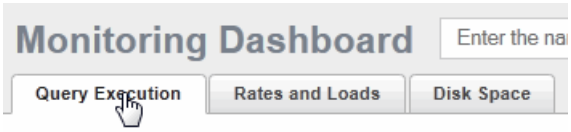


2.7 Monitoring Query Execution

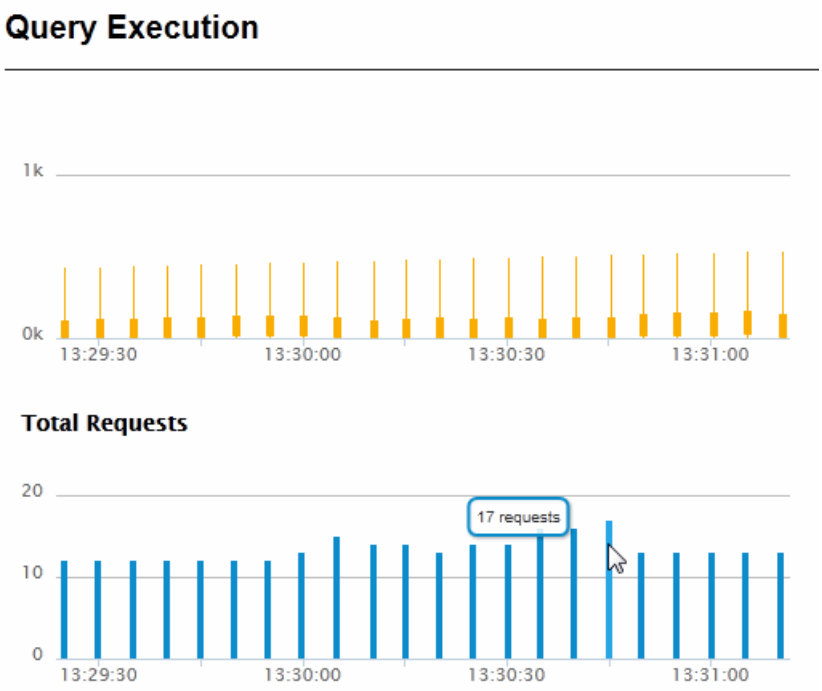
Query execution data gives you insight into the number of queries currently taking place and the execution time of these queries. Two important query execution metrics to monitor are:

- Query Execution Time — Longer than usual query execution times may indicate a bottleneck, such as a slow host or problems with XDQP communication between hosts. Other possible problems include increased loads following a failover or more than the usual number of total requests.
- Total Requests — A consistent increase in the total number of outstanding requests may indicate the need to add more capacity and/or load balance. Decreases in total requests may indicate some “upstream” problem that needs to be addressed.

To display monitoring data related to query execution, select the Query Execution tab in the top left-hand portion of the Monitoring Dashboard.



The left side of the Query Execution page displays the maximum execution time (in seconds) of the current queries and the number of requests captured at each sample interval. You can hover a query execution sample to view the mean, maximum, and minimum execution times and the standard deviation from the mean.



The right side of the Query Execution page displays the five longest running queries since the beginning of the session and the longest running queries at the current time.

5 longest running queries since 13:29:24

Host	Server	Module	Time
gordon-3.marklogic....	App-Servi...	...le/endpoints/eval.xqy	533.47s
gordon-3.marklogic....	App-Servi...	...le/endpoints/eval.xqy	523.46s
gordon-3.marklogic....	App-Servi...	...le/endpoints/eval.xqy	518.46s
gordon-3.marklogic....	App-Servi...	...le/endpoints/eval.xqy	513.45s
gordon-3.marklogic....	App-Servi...	...le/endpoints/eval.xqy	503.46s

Longest running queries at 13:31:07

Host	Server	Module	Time
gordon-3.marklogic....	App-Servi...	...le/endpoints/eval.xqy	533.47s
gordon-3.marklogic....	Documents...	/apply.xqy	124.65s
gordon-3.marklogic....	TaskServer	...ansaction-manager.xqy	72.71s
gordon-1.marklogic....	TaskServer	...push-local-forest.xqy	65.87s
gordon-1.marklogic....	TaskServer	...push-local-forest.xqy	64.61s

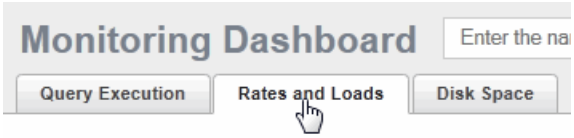
2.8 Monitoring Rates and Loads

In general, rates and loads measure how efficiently data is exchanged between applications and MarkLogic Server. Rates and loads are defined as follows:

- Rates — The amount of data (MB per second) currently being read from or written to MarkLogic Server.
- Loads — The execution time (in seconds) of current read and write requests, which includes the time requests spend in the wait queue when maximum throughput is achieved.

For details on how to interpret rates and loads, see “What is Happening on the MarkLogic Server Cluster Now?” on page 10.

To display monitoring data related to rates and loads, select the Rates and Loads tab in the top left-hand portion of the Monitoring Dashboard.



There are three types of rates and loads monitoring data. Select the type of rates and loads data by clicking on one of the three buttons displayed under Rates and Loads:

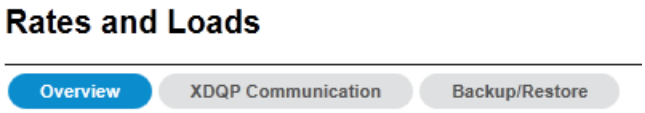


The monitoring data displayed by each of these buttons is described in the following sections:

- Overview
- XDQP Communication
- Backup/Restore

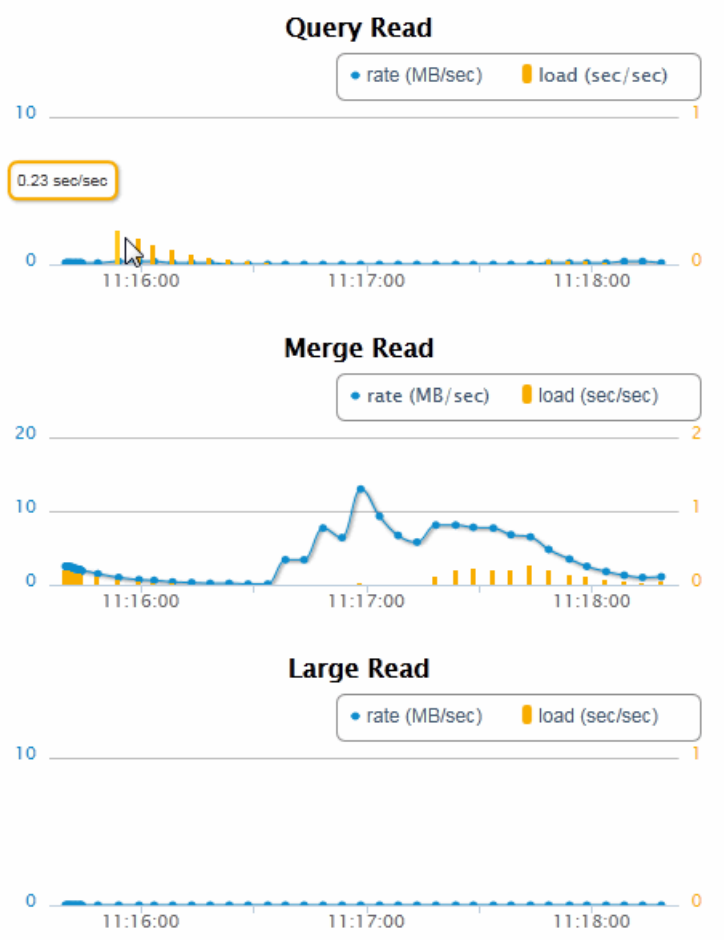
2.8.1 Overview

To obtain rates and loads data for queries, merges, and large data, click on the Overview button:

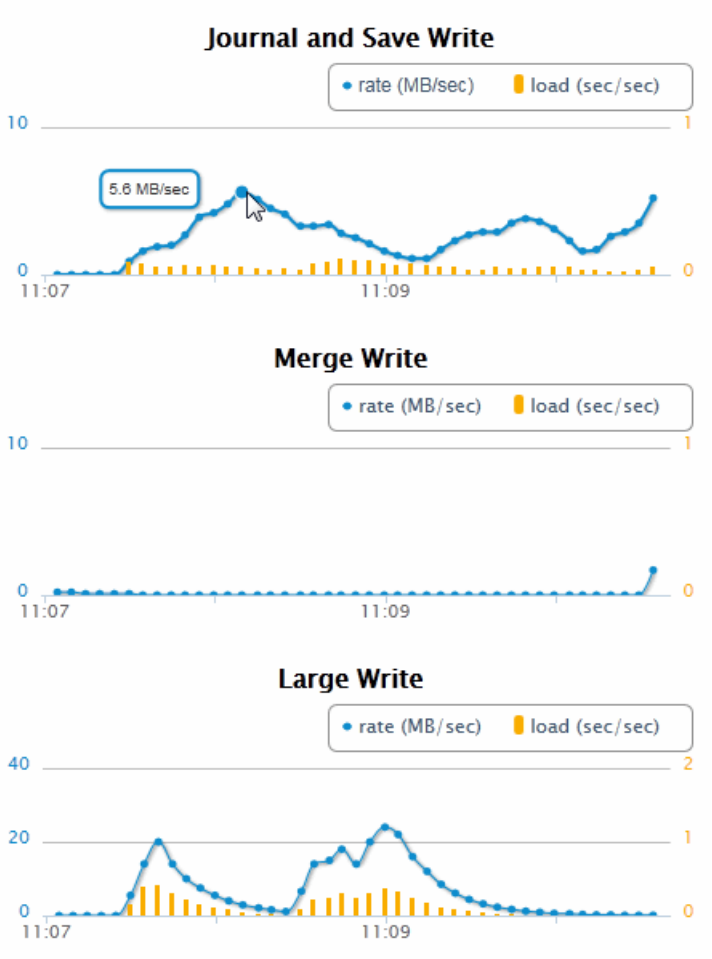


The left-hand side of the Rates and Loads Overview page displays the monitoring data related to query, merge, and large data reads.

Note: For details on Large Data, see Working With Binary Documents in the *Application Developer's Guide*.



The right-hand side of the Rates and Loads Overview page displays the monitoring data related to journal and save, merge, and large data writes.



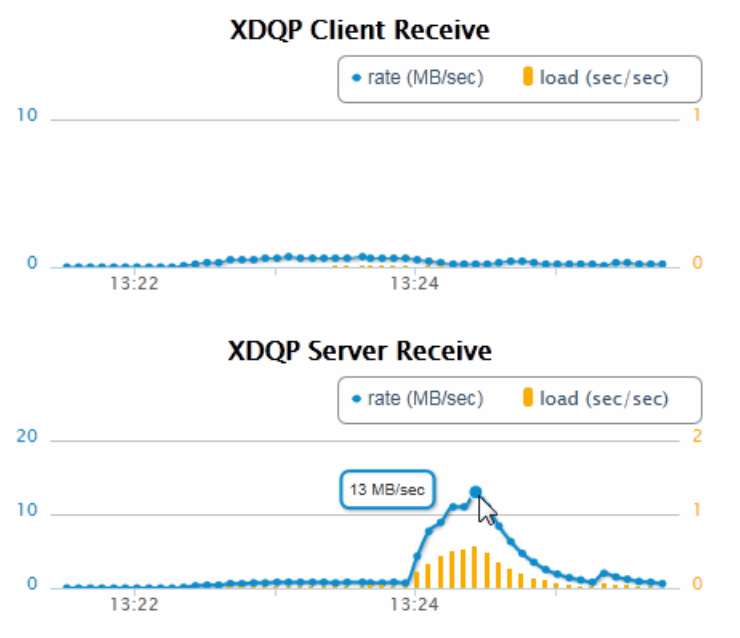
2.8.2 XDQP Communication

Communication between MarkLogic Server hosts within a cluster and between hosts in different clusters is done using the XDQP protocol. Both the rate and load are displayed for each sample interval. Unusually high XDQP loads may indicate a network connection problem.

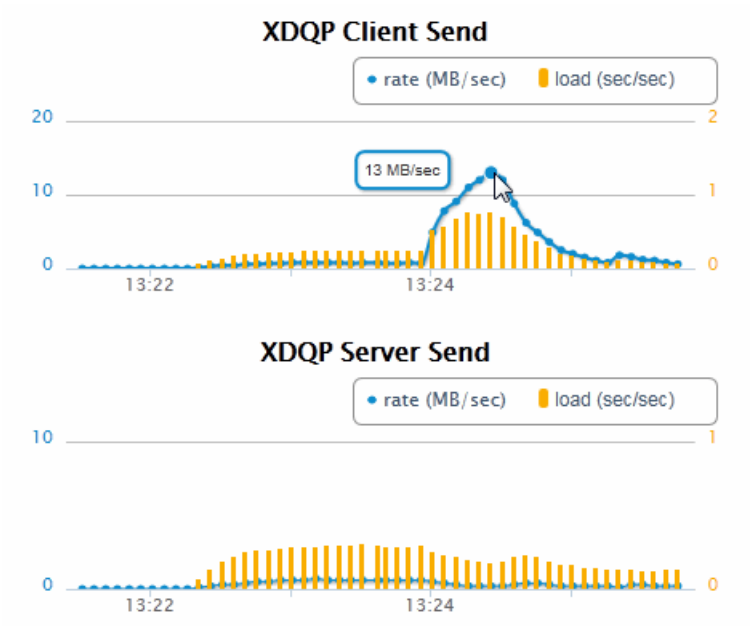
To monitor the rates and loads related to XDQP communication, click on the XDQP Communication button:



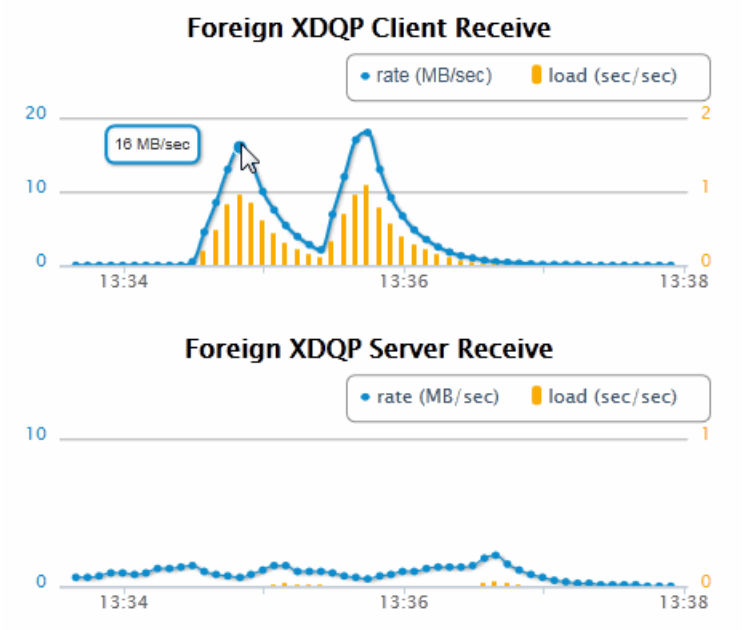
The upper left-hand side of the XDQP Communication page displays the monitoring data related to XDQP data received by the client and server.



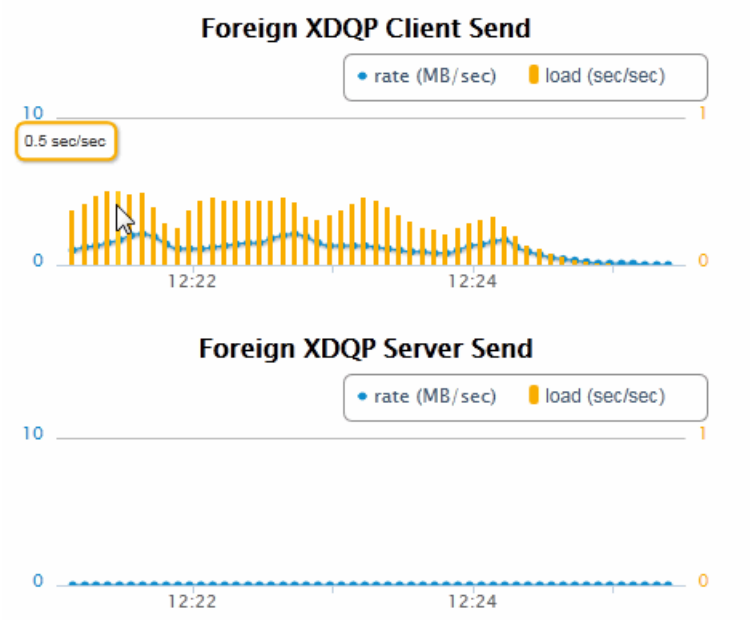
The upper right-hand side of the XDQP Communication page displays the monitoring data related to XDQP data sent by the client and server.



The lower left-hand side of the XDQP Communication page displays the monitoring data related to XDQP data received by the client and server from a foreign cluster.



The lower right-hand side of the XDQP Communication page displays the monitoring data related to XDQP data sent by the client and server to a foreign cluster.



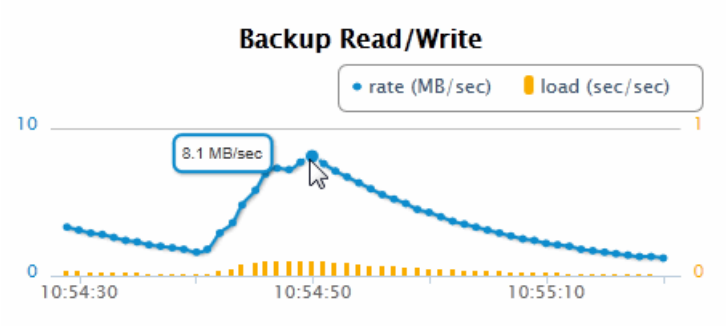
2.8.3 Backup/Restore

Backup and restore processes can impact the performance of applications and other background tasks in MarkLogic Server, such as merges and indexing.

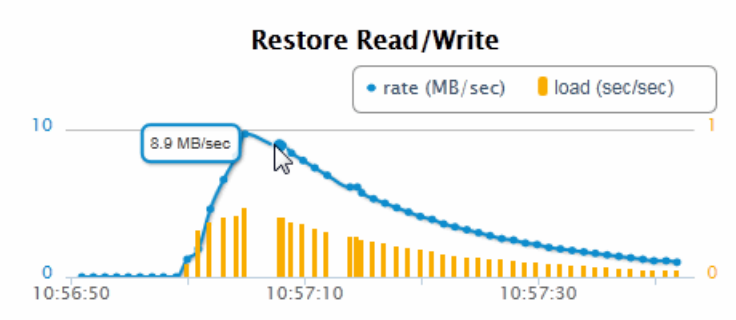
To monitor the rates and loads related to backup and restore operations, click on the Backup/Restore button:



The left-hand side of the Backup/Restore page displays the monitoring data related to Backup reads and writes.



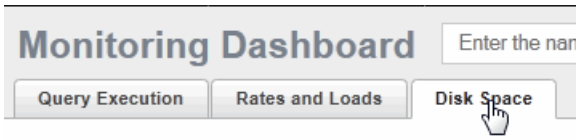
The right-hand side of the Backup/Restore page displays the monitoring data related to Restore reads and writes.



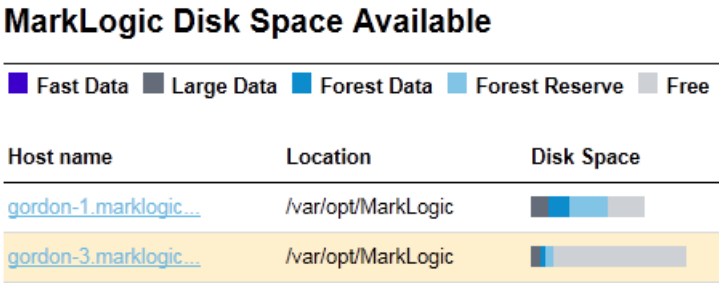
2.9 Monitoring Disk Space

Disk space usage is a key monitoring metric. In general, forest merges require twice as much disk space than that of the data stored in the forests. If a merge runs out of disk space, it will fail. In addition to the need for merge space on the disk, 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. Also, if there is no disk space available to add messages to the log files, MarkLogic Server will fail to start.

To display monitoring data related to disk space, select the Disk Space tab in the top left-hand portion of the Monitoring Dashboard.



The data displayed on the Disk Space is for a specific host. You can select the host in the upper-left-hand section of the Disk Space page. The hosts in this list are sorted by those with the least available disk space at the top.

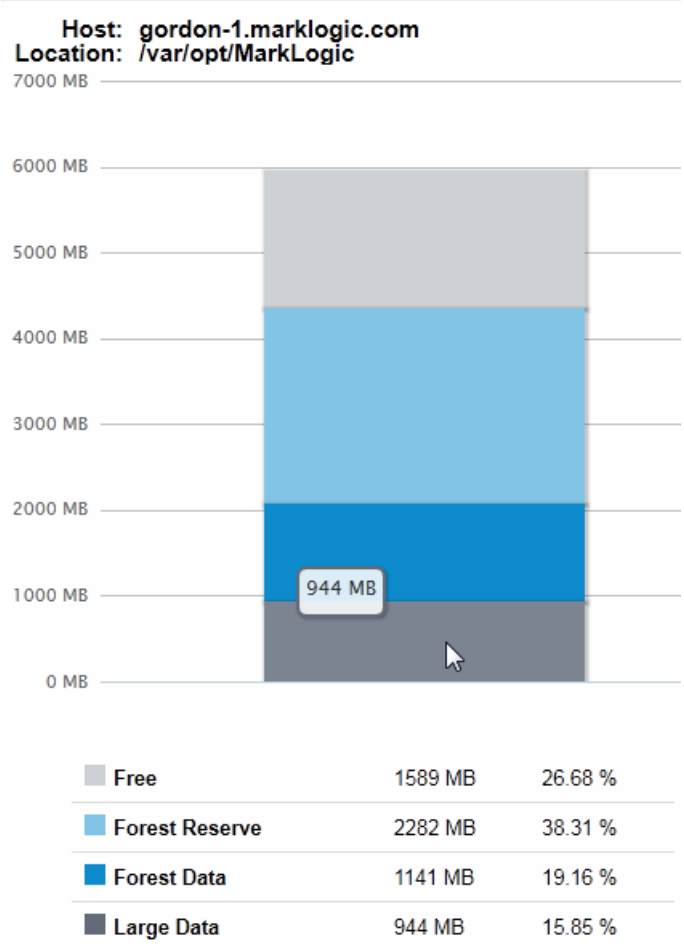


The disk space monitoring metrics are:

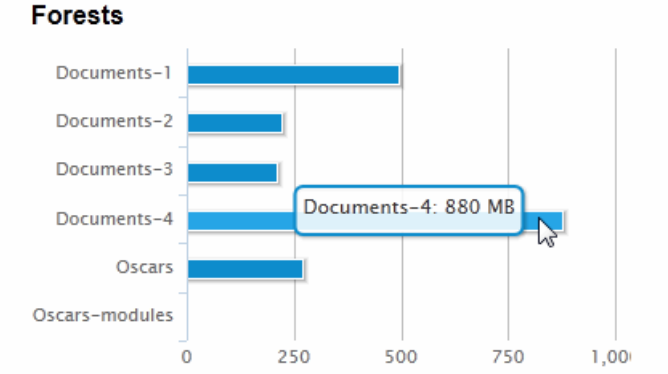
- Fast Data — The amount of disk space used by the forests' Fast Data Directory. The Fast Data Directory is typically mounted on a specialized storage device, such as a solid state disk. Fast data consists of transaction journals and as many stands that will fit on the fast storage device. For more information on Fast Data, see Fast Data Directory on Forests in the *Query Performance and Tuning Guide*.
- Large Data — The amount of disk space used by the forests' Large Data Directory. The Large Data Directory contains binary files that exceed the 'large size threshold' property set for the database. Large Data is not subjected to merges so, unlike Forest Data, Large Data does not require any additional Forest Reserve disk space. For more information on Large Data, see Working With Binary Documents in the *Application Developer's Guide*.
- Forest Data — The amount of disk space used by the data in the forest stands. This data is subject to periodic merges.

- Forest Reserve — The amount of free disk space that should be held in reserve to enable MarkLogic Server to merge the Forest Data.
- Free — The amount of free space on the disk that remains after accounting for the Forest Reserved space.

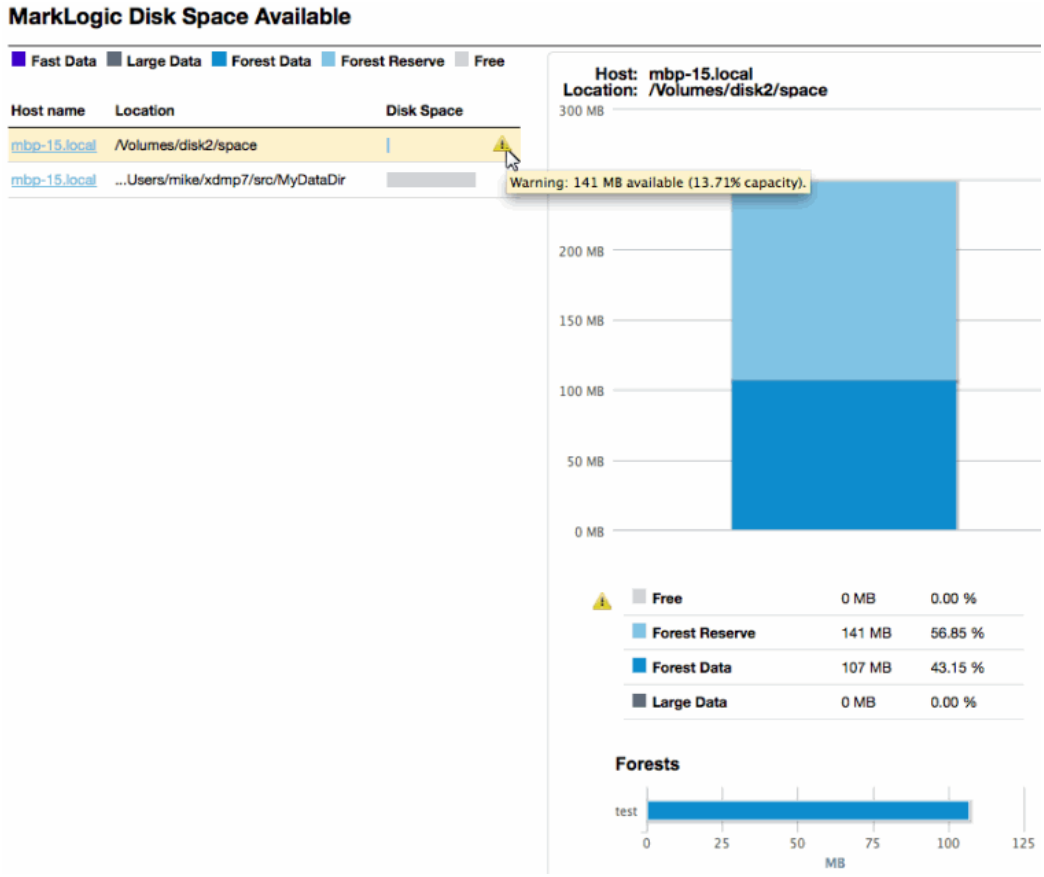
The upper right-hand section of the Disk Space page displays the amount of free space on the disk, along with how much reserve space is reserved for forest merges and the actual amount of space currently used by the forests and large data.



The lower right-hand section of the Disk Space page displays the amount of space on the disk used by the individual forests.

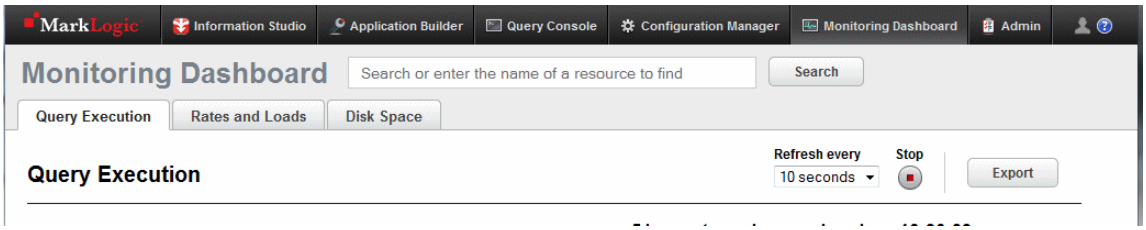


If your disk has less than 15% capacity a warning message is generated, as shown below. If the capacity falls to less than 10%, a critical message is generated.



2.10 Exporting Monitoring Data

Each of the three tabbed Monitoring Dashboard pages (Disk Space, Query Execution, Rates and Loads) has an Export button in its upper right corner, on the same line as the current tab's name. When clicked, it exports the page's data to a local XML file, formatted to be openable in Excel.



The exported files have tab-specific names incorporating a timestamp of when the file was exported. For example:

```
disk-space-20120210-160945.xml
```

indicates that it contains a page of data from the Disk Space tab, exported on February 10th, 2012 (2012 02 10) at 4:09:45 p.m. (16 09 45) (spaces added in this paragraph for clarity).

The exported data is from a JavaScript cache that automatically accumulates data as the page is drawn and refreshed. Two of the tabbed pages, Query Execution and Rates and Loads, accumulate data over time. A maximum1000 latest data points are cached for each of these pages, no matter how long the monitor page runs.

By default, data is cached every 10 seconds. This rate depends on the polling interval, which is set on the Dashboard page within the Refresh drop-down menu. See "Setting the Sample Interval" on page 18.

When using the Export button, remember these caveats:

- The cache is not in a persistent file, so manually refreshing the browser clears it of all accumulated data. Immediately after a manual browser refresh, there is no data to export.
- Clicking Export returns only the data from the current tab's page. For example, if you are on the Query Execution tab, clicking Export only writes out data from Query Execution and does not write out data from the Rates and Loads or Disk Space tabs. To get the values from all three tabs, you have to go to each tab and click its Export button, resulting in three separate files.
- However, when clicking Rates and Loads' Export button, the file does contain the data from all three of Rates and Loads' sub-tabs (Overview, XDQP Communication, and Backup/Restore).

Previously, you had to turn on caching this data with a `debug=true` parameter in the browser URL. Now, data is cached by default.

3.0 Configuring Nagios to Monitor MarkLogic Server

Nagios is a popular open source application for monitoring computer systems and networks. MarkLogic provides a Nagios plugin that makes it easy to use Nagios to monitor your MarkLogic Server cluster. This chapter describes how to configure and use Nagios to monitor your MarkLogic Server cluster.

The main topics in this chapter are:

- Terms Used in this Chapter
- Overview of the Nagios Plugin Package
- Nagios Plugin Requirements
- Installing the Nagios Plugin
- Configuring Nagios for use with MarkLogic Server
- Understanding the Generated Object Definition File
- Using Nagios

3.1 Terms Used in this Chapter

The following terms are used in this chapter:

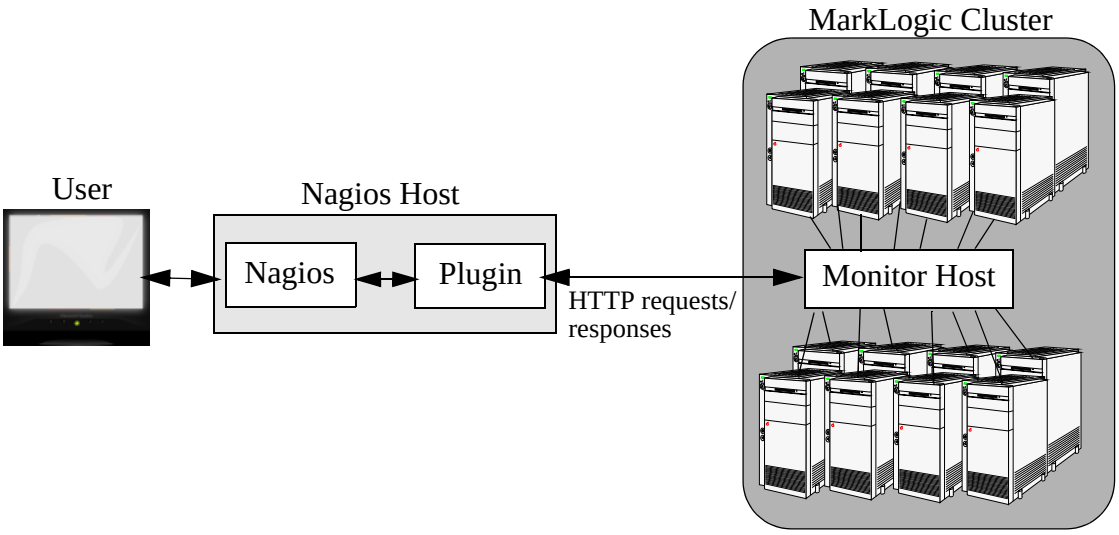
- The *Nagios plugin* is a generic Perl script that plugs into your Nagios environment to manage the requests and responses between Nagios and MarkLogic Server. Nagios uses the results returned from the plugin to display the current status of objects in a MarkLogic cluster.
- An *Nagios object* is a particular resource in MarkLogic Server, such as a cluster, host, App Server, or database.
- The *Nagios host* is the computer on which you have installed Nagios and the Nagios plugin.
- The *Monitor Host* is the host in the MarkLogic Server cluster that communicates with the Nagios plugin and returns monitoring information for the objects in the cluster.
- A *resource path* is a URL sent to MarkLogic Server to return monitoring information for an object. The resource paths are described in “Using the Management API” on page 112.
- A *service* describes what to monitor and how to monitor one or more objects in a MarkLogic cluster. Services can define warning and critical thresholds for alerting and can monitor one or more objects in MarkLogic Server.
- A *service group* is a group of one or more services.
- A *host* describes a MarkLogic Server object, such as a host, database, App Server, or cluster.
- A *host group* is a group of one or more MarkLogic Server objects.

3.2 Overview of the Nagios Plugin Package

This section describes the contents of the Nagios Plugin Package for MarkLogic Server, which you can download from developer.marklogic.com.

The recommended configuration is to install Nagios and the MarkLogic Nagios Plugin on a host outside the MarkLogic cluster. The Nagios plugin communicates via HTTP with a single MarkLogic host that is designated to monitor the entire cluster. This MarkLogic host is referred to as the *monitor host* in this chapter. The Nagios plugin translates a Nagios query into an HTTP request that contains a resource path to be executed by the Management API on the monitor host. The monitor host responds to the request with an XML node that is formatted by the plugin into status and performance data for display by Nagios.

Note: Nagios and the MarkLogic Nagios Plugin must be installed on the same host, unless you use the Nagios Remote Plugin Executor (NRPE) add-on to enable Nagios to communicate remotely with the MarkLogic Nagios Plugin.



For more information on the use of the MarkLogic resource paths, see “Using the Management API” on page 112.

The table below shows the contents of the Nagios Plugin for MarkLogic Server package.

File	Description
`ml_generic.cfg`	Contains the default settings for hosts and services that are inherited by all of the host and service definitions, as well as defines the command to execute the MarkLogic Nagios plugin.
`ml_default_installation.cfg`	An object definition file that contains all of the default settings for monitoring an out-of-the-box MarkLogic Server installation.
`check_marklogic.pl`	The MarkLogic Nagios Plugin.
`generate_marklogic_config.pl`	The script used to automatically generate the object definition file for your cluster. This is used in the configuration procedure described in “Configuring Nagios for use with MarkLogic Server” on page 37.
`ml_input_template.xml`	An XML file that defines the services to be monitored. This is used in the configuration procedure described in “Configuring Nagios for use with MarkLogic Server” on page 37.

3.3 Nagios Plugin Requirements

The computer on which you install Nagios and the plugin is referred to in this chapter as the *Nagios host*. This section describes the following requirements for the Nagios Host:

- Nagios Host Supported Platforms
- Nagios Host Library Requirements

3.3.1 Nagios Host Supported Platforms

The Nagios host must be one of the following platforms:

- Red Hat Enterprise Linux 5 (x64)
- SUSE Linux Enterprise Server 11 (x64)
- CentOS 5 (x64)
- Sun Solaris 10 (x64)

The Nagios plugin is not supported on Windows and Mac OS platforms.

The Nagios host can be used to monitor a MarkLogic cluster built on any supported platform.

3.3.2 Nagios Host Library Requirements

Before you can set up Nagios for monitoring your MarkLogic cluster, you must have the following libraries installed on your Nagios host:

- Download and install the cURL executable on your host computer and make sure the executable directory is in your PATH environment variable.

  http://curl.haxx.se/download.html

- Download and install Nagios Core and the Official Nagios Plugins on your Nagios host computer:

  http://www.nagios.org/download

- To execute the Plugin and the configuration generation script, you will need Perl, version 5 and the following modules, which are available from CPAN:
  - Nagios::Plugin::Threshold, version 0.35: http://search.cpan.org/CPAN/authors/id/T/TO/TONVOON/Nagios-Plugin-0.35.tar.gz
  - Getopt::Long, version 2.38: http://search.cpan.org/CPAN/authors/id/J/JV/JV/Getopt-Long-2.38.tar.gz
  - List::Util, version 1.23: http://search.cpan.org/CPAN/authors/id/G/GB/GBARR/Scalar-List-Utils-1.23.tar.gz
  - XML::LibXML, version 1.88: http://search.cpan.org/CPAN/authors/id/S/SH/SHLOMIF/XML-LibXML-1.88.tar.gz

- List::MoreUtils, version 0.33:
  http://search.cpan.org/CPAN/authors/id/A/AD/ADAMK/List-MoreUtils-0.33.tar.gz

3.4 Installing the Nagios Plugin

Below is the procedure to install the Nagios plugin for MarkLogic Server. The `ml_default_installation.cfg` file described in this procedure only recognizes the out-of-the-box objects in your initial installation of MarkLogic Server and should only be used to confirm your installation. In order to monitor any objects created after your initial installation, follow the procedure described in either "Configuring Nagios for use with MarkLogic Server" on page 37.

Note: The following procedure assumes you have installed Nagios in the default location.

1. Unzip the Nagios Plugin package you downloaded from developer.marklogic.com.

2. Move the `check_marklogic.pl` plugin to the directory:

```
/usr/local/nagios/libexec
```

3. Confirm that you have executable permission on the `check_marklogic.pl` plugin.

4. Move the `ml_default_installation.cfg` and `ml_generic.cfg` files to the directory:

```
/usr/local/nagios/etc/objects
```

5. From the `/usr/local/nagios/libexec` directory, execute the `check_marklogic.pl` script as shown below to verify that you have all of the dependencies installed and can connect to the Management API on the Monitor host. The `check_marklogic.pl` command uses the following parameters, where *`user:pwd`* are the credentials for a user with the `manage-user` role and *`hostname`* is the name of the monitor host in the MarkLogic cluster:

```
perl check_marklogic.pl -a user:pwd -H hostname -p 8002 --path
/manage/v1/databases
```

Note: All text is case-sensitive.

It should return `OK`; if it does not, use the `--verbose` parameter, as follows, to troubleshoot the problem:

Verbose Level	Description
`--verbose 1`	Adds the user input in the status message. Good for debugging the Nagios object definition files. This parameter can also be used in the `check_command` of a host or service definition in order to debug any problems. For details, see "The check_command Parameter" on page 43.

Verbose Level	Description
`--verbose 2`	Adds limited debug messages. Best for command line debugging.
`--verbose 3`	Adds all debug messages. Best for command line debugging and detecting problems with the plugin.

6. Modify the `/usr/local/nagios/etc/nagios.cfg` file to point to your object definition and generic files:

```
cfg_file=/usr/local/nagios/etc/objects/ml_default_installation.cfg
cfg_file=/usr/local/nagios/etc/objects/ml_generic.cfg
```

7. Edit the `ml_default_installation.cfg` file as follows:

   - Replace all instances of `myhost.marklogic.com` with the name or IP address of your monitor host.
   - Replace `user:pw` with the username and password of a user with the `manage-user` role.

8. Test that there are no errors in the object definition file:

```
/usr/local/nagios/bin/nagios -v /usr/local/nagios/etc/nagios.cfg
```

9. Restart Nagios

```
sudo /etc/rc.d/init.d/nagios restart
```

10. Confirm that you can access Nagios, as described in “Using Nagios” on page 53

3.5 Configuring Nagios for use with MarkLogic Server

This section describes how to automatically generate a Nagios object definition file for your cluster.

This section includes the following topics:

- The generate_marklogic_config.pl Script
- The Monitoring Services File

3.5.1 The generate_marklogic_config.pl Script

You can use the `generate_marklogic_config.pl` script to automatically generate an object definition file for your cluster. The script reads the `services.xml` file described in "The Monitoring Services File" on page 40, which defines the services to be monitored, the threshold information for each service, and any objects you want to exclude from monitoring. When you run the `generate_marklogic_config.pl` script, it reads all of the objects (hosts, databases, and App Servers) from your cluster and creates an object definition file that can be used by the Nagios plugin to monitor all of the objects using the services specified in the XML services file.

To run the script, make sure you have the required Perl modules, as described in "Nagios Plugin Requirements" on page 34, open a shell window, and enter a command with the following syntax:

```
perl generate_marklogic_config.pl -a user:pwd -H monitorHost -verbose 1
-f services.xml -u prefix -c clusterName -p port >
MyObjectDefinition.cfg
```

Warning The username and password you enter will be stored in the `MyObjectDefinition.cfg` file.

In the above example, the output from the script is directed to the object definition file, `MyObjectDefinition.cfg`. Reference this file from your `nagios.cfg` file and restart Nagios, as described in "Installing the Nagios Plugin" on page 36.

The parameters for `generate_marklogic_config.pl` are described in the table below.

Parameter	Description	Example	Required?
`-a (--authentication)` *`user:pwd`*	The authentication credentials for the user on the host machine selected to monitor the cluster. This user must have the `manage-user` role.	`-a myName:myPwd`	yes
`-H (--host)` *`monitorhost`*	The name or IP address of the Monitor host selected to monitor the cluster.	`-H Monitor-Host`	yes
`-p (--port)` *`portNumber`*	The port number to be used by the Monitor host.	`-p 8002`	yes
`-c (--clustername)` *`clusterName`*	The name you want to assign to the host group. For information on host groups, see the Host Group Definition in the Nagios documentation.	`-c ML-Cluster1`	yes

Parameter	Description	Example	Required?
`-u (--uniqueshortcut)` *`prefix`*	A unique string to be used as a prefix for every object in your cluster. This prefix must be unique across all clusters.	`-u Clu-1`	yes
`-f (--filename)` *`servicesFile`*`.xml`	The name of the XML file that contains the configuration settings for your monitoring services. The best practice is to copy the `ml_input_template.xml` file and edit the copy, which is referred to in this chapter as `services.xml`.	`-f services.xml`	yes
`-h (--help)`	Display help information.	`-h`	no
`-t (--timeout)` *`seconds`*	Set the number of seconds to wait for each service to execute before returning an error message.	`-t 60`	no
`-V (--version)`	Display the Nagios plugin version.	`-V`	no
`-v (--verbose) {0 | 1 | 2 | 3}`	`-v 0` Returns no debugging information. `-v {1 | 2 | 3}` Validates the check_commands by executing them against the plugin and reports any problematic services. Higher verbosity levels report more debugging information. The plugin must be in the same directory as the `check_marklogic.pl` and `generate_marklogic_config.pl` scripts.	`-v 1`	no

Parameter	Description	Example	Required?
`-ssl (--ssl) {0|1}`	Determine whether to enable SSL on the monitor host. 0: Do not enable SSL (default) 1: Enable SSL. The `manage` App Server must have SSL enabled and use the same certificate as specified by the `-cert` parameter below. For details on configuring SSL, see Configuring SSL on App Servers in the *Administrator's Guide*.	`-ssl 1`	no
`-cert (--sslcertificate)` *`mycertificate`*`.crt`	Specifies the location of the certificate used for SSL access to MarkLogic Server. Only specify this parameter when SSL is enabled (`-ssl 1`). The `manage` App Server must use the same certificate specified by this parameter. For details creating an SSL certificate, see Configuring SSL on App Servers in the *Administrator's Guide*.	`-cert cert.crt`	Only if SSL is enabled

3.5.2 The Monitoring Services File

The `ml_input_template.xml` file describes the services used to monitor your cluster in XML format. It is recommended that you copy this file and make changes on the copy. This copy is referred to in this chapter as `services.xml`. You can edit the `services.xml` file to add or modify services, add or modify the sample intervals and thresholds, and to exclude specific resources from any of the services used to monitor your cluster.

This section describes the following topics:

- Globally Excluding Resources
- Service Definitions
- The check_command Parameter
- Defining and Setting Thresholds and Ranges

3.5.2.1 Globally Excluding Resources

At the top of the `ml_input_template.xml` file is a global `excludes` element that lists the resources (by name) to be excluded by all of the monitoring services. For example, to exclude the `Schemas` database, the `Admin` App Server in the `Default` group, and the `myhost` host from being monitored by all of the services, the `excludes` element would look like the following:

```
<excludes>
    <exclude resourcetype="Databases" name="Schemas" />
    <exclude resourcetype="Servers" name="Admin?group-id=Default"/>
    <exclude resourcetype="Hosts" name="myhost.marklogic.com"/>
</excludes>
```

3.5.2.2 Service Definitions

The purpose of this section is to describe the service definitions used by the `generate_marklogic_config.pl` script to generate the object definition file described in "Understanding the Generated Object Definition File" on page 50.

For example, the following XML defines the `backup-count` service in the `services.xml` file:

```
<service-template type="Databases" refresh="default">
    <service_description>backup-count</service_description>
    <service_note>Number of backups in progress</service_note>
    <check-command>
        check_command check_marklogic.pl! -a $_HOSTMLUSERPW$
-port $_HOSTMLPORT$  --host $_HOSTMLIP$
-path  /manage/v1/databases/$_HOSTMLALIAS$/status -key $SERVICEDESC$
    </check-command>
    <exclude name="Security"></exclude>
</service-template>
```

Assuming you have a default installation of MarkLogic Server, the resulting `backup-count` service generated from the `generate_marklogic_config.pl` script will look like the following in your generated object definition file:

```
define service{
    use             ML-generic-service
    host_name       MyCl-App-Services, MyCl-Documents, MyCl-Fab,
MyCl-Last-Login, MyCl-Modules, MyCl-Schemas, MyCl-Triggers
    service_description backup-count
    notes           Number of backups in progress
    servicegroups   MyCl-Databases
    check_command   check_marklogic.pl! -a $_HOSTMLUSERPW$
-port $_HOSTMLPORT$  --host $_HOSTMLIP$
-path  /manage/v1/databases/$_HOSTMLALIAS$/status
-key $SERVICEDESC$ $_HOSTMLSSL$ $_HOSTMLTIMEOUT$
}
```

Note: Nagios uses the `$SERVICEDESC$` variable to hold the value of `service_description`. Consequently, in order to use `$SERVICEDESC$` in the resource path, the `service_description` must be an element in the node returned by `-path`.

The `service-template` element contains three attributes:

- `type` — The `servicegroups` value in the resulting service definition is made up from the string specified by the `generate_marklogic_config.pl -u` parameter and the value specified by the `type` attribute. The `type` can be any object you can specify in a resource path, such as `Local-Cluster, Foreign-Clusters`, `Groups`, `Servers`, `Hosts`, or `Databases`. For details on resource paths in the Management API, see “Resource Addresses” on page 114.
- `refresh` — Specifies the value for `check_interval` in the resulting service definition. If set to `default`, Nagios uses the value from the `ml_generic.cfg` file.

The `check_command` is described in detail in “The check_command Parameter” on page 43.

The `exclude` element is used to exclude certain resources from this specific service. For example, to exclude the `Security` database from this service, use:

```
<exclude name="Security"></exclude>
```

For details on how to exclude resources on a global level, see “Globally Excluding Resources” on page 41.

3.5.2.3 The check_command Parameter

This section provides more detail on the `check_command` defined in a service definition in the `services.xml` file.

The `check_command` contains the information used by the plugin to construct the HTTP request sent to MarkLogic Server. For example, the `update-count` service defines the `check_command` as follows:

```
check_command   check_marklogic.pl! -a $_HOSTMLUSERPW$
-port $_HOSTMLPORT$  --host $_HOSTMLIP$
-path  /manage/v1/requests?server-id=$_HOSTMLALIAS$
-key $SERVICEDESC$
```

The `check_marklogic.pl` script identifies the plugin that translates the monitoring requests and responses between MarkLogic Server and Nagios. The name of the plugin is defined in the `ml_generic.cfg` file and it must be specified in every `check_command`.

The `check_command` contains what Nagios refers to as *custom variable macros* and *standard macros*. The custom variable macros used by the `-a`, `--port`, and `--host` parameters represent the login credentials, port number, and MarkLogic Server host name (or IP address) set in the `MyCl_abstract` host definition. The `--path` specifies the resource path to be sent to MarkLogic Server. The `$_HOSTMLALIAS$` is a custom variable macro that specifies the resource to be monitored.

The `$SERVICEDESC$` macro is a standard macro used by Nagios to hold the value of the `service_description` (see the definition of the $SERVICEDESC$ macro in the Nagios documentation). The `service_description` in this example is `update-count`, so when Nagios checks the `update-count` service for the four App Servers, it will replace the macros and call the plugin four times with the following resource addresses:

```
http://localhost:8002/manage/v1/requests?server-id=Admin
http://localhost:8002/manage/v1/requests?server-id=App-Services
http://localhost:8002/manage/v1/requests?server-id=Manage
http://localhost:8002/manage/v1/requests?server-id=TaskServer
```

The Management API is described in detail in “Using the Management API” on page 112. See Understanding Macros and How They Work and Standard Macros in Nagios in the Nagios documentation for details on custom variable and standard macros.

The following table lists the possible parameters that can be used by `check_marklogic.pl` script. The parameters that are required in the `check_marklogic.pl` script are flagged by "(Required)." The parameters related to thresholding and ranges are described in more detail in "Defining and Setting Thresholds and Ranges" on page 47.

Parameters	Description
`-a (--authentication)` *`user:pwd`* (Required)	Authentication for the user on the monitor host that is set in the abstract host definition (required). This is captured in the `$_HOSTMLUSERPW$` macro.
`-H (--host)` *`hostname`* (Required)	Name or IP address of the monitor host that is set in the abstract host definition (required). This is captured in the `$_HOSTMLIP$` macro.
`-p (--port)` *`port`* (Required)	Port number used by the Management API on the monitor host that is set in the abstract host definition (required). This is captured in the `$_HOSTMLPORT$` macro.
`--path` *`path`* (Required)	The resource address path (required).
`-ssl (--ssl) {0 | 1}`	0: Do not enable SSL (default) 1: Enable SSL
`-cert (--sslcertificate)` *`mycertificate`*`.crt`	Specifies the location of the certificate used for SSL access to MarkLogic Server. Only specify this parameter when SSL is enabled (`-ssl 1`).
`-t (--timeout)` *`timeout-seconds`*	The timeout, in seconds, that defines the maximum time to wait for a response to a service. (default is 10).
`-v (--verbose) {0 | 1 | 2 | 3}`	0: Return no debugging information (default 0). 1: Returns the user input in the status message. {2 | 3}: Returns various levels of additional debugging information.

Parameters	Description
`-k (--key) element` (Required if thresholds are used)	The element value to return for the specified resource. You can specify any simple or complex element in the node returned in response to the resource path. You can optionally use the `-op` parameter below to inspect the element value and determine a threshold.
`-op (--operator) {range | {{eq | ne}=string}}`	Specifies that a threshold is to be determined by a range of values, or by the presence or absence of a specific string. The default for -`op` is `range`. For details, see "Defining and Setting Thresholds and Ranges" on page 47.
`-w (--warning) range`	The range used to inspect the threshold value to determine whether flag the object with a warning. For details, see "Defining and Setting Thresholds and Ranges" on page 47.
`-c (--critical) range`	The range used to inspect the threshold value to determine whether flag the object as critical. For details, see "Defining and Setting Thresholds and Ranges" on page 47.

You can test the results of a `check_command` by navigating to the `/usr/local/nagios/libexec` directory and executing the `check_marklogic.pl` script using the following format:

```
perl check_marklogic.pl  -a user:pwd --port 8002  --host hostName
--path /manage/v1/URI [--key resource [-op range -w value -c value]]
```

For example, to return the `database-counts` node, you can enter:

```
perl check_marklogic.pl  -a admin:admin  -port 8002
--host gordon-1 -path  /manage/v1/databases/Documents/counts
```

You will see a result like:

```
OK - Documents-elapsed-time=0.057927s |
Documents-elapsed-time=0.057927s; Documents-documents=590003;
Documents-directories=5; Documents-active-fragments=118001 1;
Documents-deleted-fragments=16; Documents-nascent-fragments=0;
```

To return the `documents` value in the `database-counts` node, you can enter:

```
perl check_marklogic.pl  -a admin:admin  -port 8002
--host gordon-1 -path  /manage/v1/databases/Documents/counts
--key documents
```

You will see a result like:

```
OK - Documents-documents=590003 | Documents-documents=590003;;
```

To test a threshold, set a threshold value higher than the document count. For example, enter:

```
perl check_marklogic.pl  -a admin:admin  -port 8002  --host gordon-1
-path  /manage/v1/databases/Documents/counts -key documents -op range
-w 10000: -c 3000000:
```

This should result in a critical message like the following:

```
Critical - documents=590003 [critical(3000000:)][warning(10000:)] |
Documents-documents=590003;10000:;3000000:;
```

You can use the same approach to test other thresholds.

3.5.2.4 Defining and Setting Thresholds and Ranges

The syntax for setting thresholds and ranges in Nagios is described in the Threshold and ranges section in the Nagios documentation. The purpose of this section is to describe the parameters that are specific to setting thresholds and ranges in the object definition file used by the `check_marklogic.pl` plugin.

> **Note:** Thresholds and ranges should be set in the `services.xml` file, as described in "The Monitoring Services File" on page 40

The parameters used by `check_command` to set thresholds and ranges in the `services.xml` file are shown in the following table.

Parameter	Description
`-k (--key) element`	The element value to inspect and determine a threshold. This can be any element in the node returned in response to the resource path.
`-op (--operator) {range | {{eq | ne}=string}}`	Defines whether the threshold is to be determined by a range of values, or by the presence or absence of a specific string. This must follow a `--key` parameter.

The thresholds are set in a separate `threshold` element and they are:

Threshold Element	Description
`<default-warning>`	The range used to inspect the threshold value to determine whether flag the object with a warning.
`<default-critical>`	The range used to inspect the threshold value to determine whether flag the object as critical.
`<custom-threshold name="resource" warning="value" critical="value">`	The threshold ranges to apply to a specific resource.

For example, the `documents` service defined in the `services.xml` file returns the number of documents stored in the databases. The default service definition in your `services.xml` file is:

```
<service-template type="Databases" refresh="60">
    <service_description>documents</service_description>
    <service_note>
        Document count for attached forests (excluding replicas)
    </service_note>
    <check-command>check_command   check_marklogic.pl!
-a $_HOSTMLUSERPW$  -port $_HOSTMLPORT$  --host $_HOSTMLIP$
-path  /manage/v1/databases/$_HOSTMLALIAS$/counts
-key $SERVICEDESC$
    </check-command>
</service-template>
```

If you want to define a threshold to generate a warning if the document count on any of the databases monitored by the `documents` service exceeds 1000 documents and a critical warning if the document count exceeds 10000, then you can add a `threshold` element to your `services.xml` file as follows:

```
<service-template type="Databases" refresh="60">
    <service_description>documents</service_description>
    <service_note>
        Document count for attached forests (excluding replicas)
    </service_note>
    <check-command>check_command   check_marklogic.pl!
-a $_HOSTMLUSERPW$  -port $_HOSTMLPORT$  --host $_HOSTMLIP$
-path  /manage/v1/databases/$_HOSTMLALIAS$/counts
-key $SERVICEDESC$
    </check-command>
    <threshold>
       <default-critical>10000</default-critical>
       <default-warning>1000</default-warning>
    </threshold>
</service-template>
```

Note: The `check_command` must be defined as a continuous line with no returns.

The `check_commend --key` parameter specifies the XML element on which to apply a threshold. In the above example, the name of this element is `documents`, which is defined in the `service_description` and stored in the `$SERVICEDESC$` macro (see the definition of the $SERVICEDESC$ macro in the Nagios documentation). The `--key` element must be an element in `database-counts` node returned by the resource path, `/manage/v1/databases/Documents/counts`.

Another type of threshold is to determine either the presence or absence of a particular string. The -`op eq=`*`string`* and `-op ne=`*`string`* parameters evaluate values as strings, calculating equality or inequality, respectively, and convert the value to either 0 (false) or 1 (true). For example, the `state` service used to detect the state of the databases makes use of the `-op ne=unavailable` parameter and the `default-critical @0:0` element defined for each database to generate a critical flag on any database that is not in the 'available' state (boolean value of 0).

```
<service-template type="Databases" refresh="default">
    <service_description>state</service_description>
    <service_note>State of the database</service_note>
    <check-command>check_command   check_marklogic.pl!
-a $_HOSTMLUSERPW$  -port $_HOSTMLPORT$  --host $_HOSTMLIP$
-path  /manage/v1/databases/$_HOSTMLALIAS$/status
-key $SERVICEDESC$ -op ne=unavailable
    </check-command>
    <threshold>
    <default-critical>@0:0</default-critical>
    </threshold>
</service-template>
```

You can also set a custom threshold for specific objects. For example, below, the `default-critical` and `default-warning` elements set the thresholds for all of the objects, which are databases in this example. The `custom-threshold` element overrides the default thresholds and sets the given thresholds for only the `Documents` database.

```
<threshold>
    <default-critical>@10:399</default-critical>
    <default-warning>@400:1000</default-warning>
    <custom-threshold name="Documents" warning="@5000:7000"
critical="@7001:10000"></custom-threshold>
</threshold>
```

You can create a custom threshold for more than a single object. For example, the `query-count` service calls the following method to return the query count for each App Server in the cluster:

```
-path /manage/v1/requests?server-id=$_HOSTMLALIAS$ -key $SERVICEDESC$
```

You have a number of App Servers in your cluster and you want to set a different threshold for the Admin App Server `Default` group. To set a custom threshold for the Admin App Servers in the `Default` group, you can define a `custom-threshold` with the following name (note the need to escape '`&`'):

```
<custom-threshold name="Admin\\&group-id=Default" ....
```

This would result in a resource path like the one below to determine whether the query count for the Admin App Server in the `Default` group has fallen outside of the specified thresholds.

```
-path /manage/v1/requests?server-id=Admin\\&group-id=Default
-key $SERVICEDESC$
```

3.6 Understanding the Generated Object Definition File

This section describes the object definition file generated by the following call to the `generate_marklogic_config.pl` script on a default installation of MarkLogic Server on a single host.

```
perl generate_marklogic_config.pl -a admin:admin -H gordon-1
-f ml_input_template.xml -u MyCl -c ML-Cluster1
-p 8002 > MyObjectDefinition.cfg
```

Note: Manual edits to an object definition file are not recommended, as it is easy to introduce errors and changes are hard to manage.

The Nagios process that monitors an object in MarkLogic Server is called a *service*. Services are grouped into *service groups*. The `MyObjectDefinition.cfg` object definition file generated from the above script defines the following service groups:

- MyCl-Servers
- MyCl-Databases
- MyCl-Hosts
- MyCl-Local-Cluster
- MyCl-Foreign-Cluster

The `on-disk-size` service looks like the following:

```
define service{
    use             ML-generic-service
    host_name       MyCl-App-Services, MyCl-Documents, MyCl-Fab,
MyCl-Last-Login, MyCl-Modules, MyCl-Schemas, MyCl-Security,
MyCl-Triggers
    service_description on-disk-size
    notes           Total size of forest data on disk (MB)
    servicegroups   MyCl-Databases
    check_command   check_marklogic.pl! -a $_HOSTMLUSERPW$
-port $_HOSTMLPORT$  --host $_HOSTMLIP$
-path  /manage/v1/databases/$_HOSTMLALIAS$/status
-key $SERVICEDESC$   $_HOSTMLSSL$ $_HOSTMLTIMEOUT$
    check_interval 10
}
```

The second line in the service definition, `use ML-generic-service`, specifies that the service inherits default settings from the `ml_generic.cfg` file. These default settings are used unless expressly overridden for a host or service in your `MyObjectDefinition.cfg` file. For example, the value of the default `check_interval` setting is 1, but the definition of the `on-disk-size` service above overrides the default and sets the `check_interval` value to 10.

The `host_name` portion of the service definition defines which objects are monitored by this service. The `service_description` is the name of the service that appears in the Nagios UI and `notes` is a simple comment that describes the service.

Nagios allows you to aggregate services into groups, known as *service groups*. The `servicegroups` value specifies that this service belongs to the group of database monitoring services, known as `MyCl-Databases`, for the cluster, `MyCl`. See Service Group Definition in the Nagios documentation for more detail on service groups. The `check_command` portion defines the specific monitoring method to be used on those objects. The `check_command` is described in more detail in "The check_command Parameter" on page 43.

The abstract host definition shown below defines the port number, login credentials, and name of the monitor host in the cluster to execute the monitoring methods and relay the results to the plugin for use by Nagios.

For example, the abstract host definition might look like:

```
define host{
    use         ML-generic-host
    name        MyCl_abstract
    _MLPORT     8002;
    _MLUSERPW   usr:pwd;
    _MLIP       gordon-1;
    _MLSSL
    _MLTIMEOUT
    register    0;
}
```

Like the services, the abstract host definition inherits default settings from the `ml_generic.cfg` file, which are used unless they are expressly overridden in the host definition in your `MyObjectDefinition.cfg` file.

Following the abstract host definition is a series of `define host` definitions that describe which objects to monitor on the cluster.

> **Note:** In Nagios, every object, whether it is a MarkLogic Server host, App Server, database, or the overall cluster, is referred to as a 'host', so you must distinguish between the abstract host definition for the monitor host, `MyCl_abstract`, above from the host definitions for the MarkLogic Server objects, like the one shown for the `Documents` database below.

For example, the definition that directs Nagios to monitor the `Documents` database looks like:

```
define host{
    use                     MyCl_abstract
    host_name               MyCl-Documents
    address                 MyCl-Documents
    hostgroups              ML-Cluster1
    check_command check_marklogic.pl!  -a $_HOSTMLUSERPW$
-port $_HOSTMLPORT$  --host $_HOSTMLIP$ $_HOSTMLSSL$ $_HOSTMLTIMEOUT$
-path /manage/v1/databases/Documents/status -key state -c @0:0
-op eq=available
    _MLALIAS    Documents

    _CRITICAL-STATE @0:0
    _CRITICAL-FAILED-MASTERS 0:0
    _CRITICAL-ASYNC-REPLICATING 0:0
    _CRITICAL-DATABASE-REPLICATION-ACTIVE @0:0
    _WARNING-FOREIGN-FORESTS-LAG-EXCEEDED 0:0
}
```

The `use MyCl_abstract` parameter specifies that the host definition is to inherit all of the setting from the abstract host, `gordon-1`, defined earlier. The `host_name` identifies this database as `MyCl-Documents`. The `MyCl` prefix is used to distinguish between different `Documents` databases in different clusters.

The `address` value, `MyC1-Documents`, is a dummy value that is required by Nagios but is not used in this configuration. The `hostgroups` value, `ML-Cluster1`, specifies that this database belongs to this group of hosts. You don't need to be concerned with this name, unless you are going to include more than one object definition file in your `nagios.cfg` file, in which case the names specified for your `host_names` and `hostgroups` in each object definition file must be unique.

The `check_command` is used to report whether the resource is enabled. The `_MLALIAS` variable specifies that the object to be monitored is the `Documents` database. This variable can be used as part of a macro (`_HOSTMLALIAS`) in the resource path so that multiple objects can be monitored by a single service, as described in "The check_command Parameter" on page 43.

The macros `_CRITICAL-STATE`, `_CRITICAL-FAILED-MASTERS 0:0`, `CRITICAL-ASYNC-REPLICATING 0:0`, `_CRITICAL-DATABASE-REPLICATION-ACTIVE @0:0`, and `_WARNING-FOREIGN-FORESTS-LAG-EXCEEDED 0:0` define the thresholds for this resource. Thresholding is described in "Defining and Setting Thresholds and Ranges" on page 47.

3.7 Using Nagios

For details on how to use the Nagios User Interface, see the Nagios Core 3.x Documentation. To access Nagios, enter a URL with the following format:

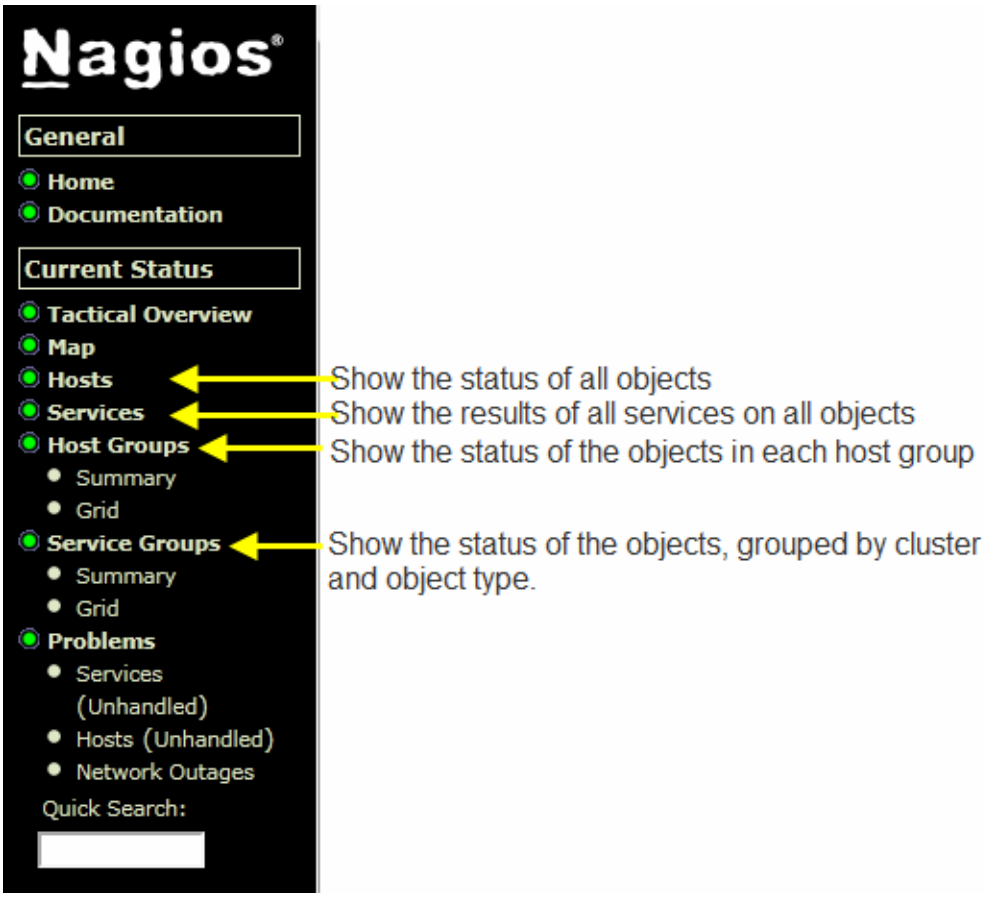
```
http://hostName/nagios/
```

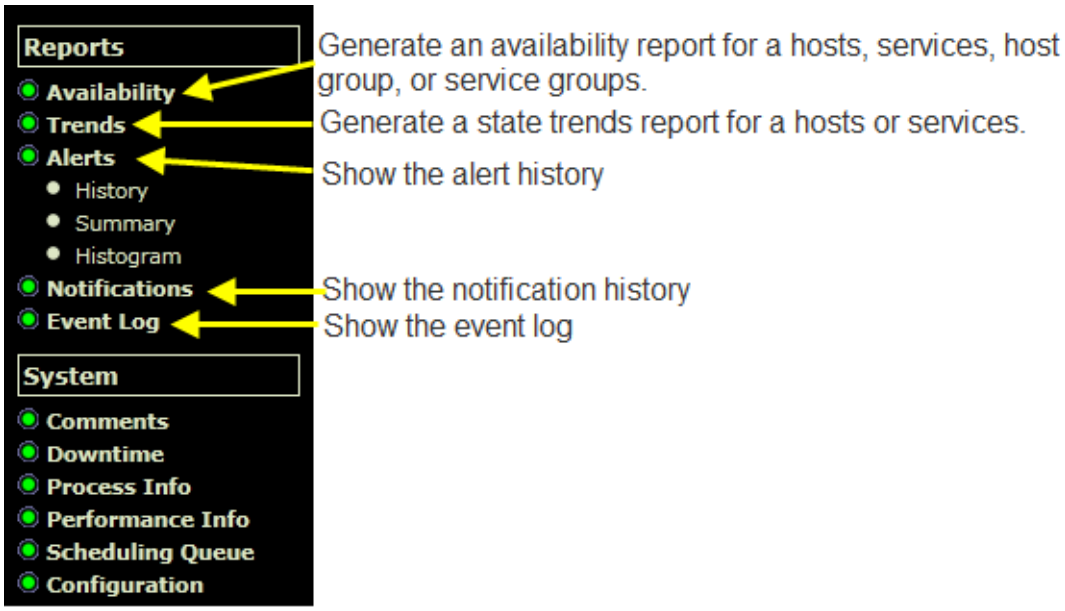
This section contains the following topics:

- Nagios Navigation Panels
- Host Groups
- Service Groups
- Service Status Details for a Resource

3.7.1 Nagios Navigation Panels

The navigation panel on the left side of the page looks like:




Reports
Availability
Trends
Alerts
History
Summary
Histogram
Notifications
Event Log
System
Comments
Downtime
Process Info
Performance Info
Scheduling Queue
Configuration
Generate an availability report for a hosts, services, host group, or service groups.
Generate a state trends report for a hosts or services.
Show the alert history
Show the notification history
Show the event log

3.7.2 Host Groups

The Host Groups page displays all of the Nagios services grouped by cluster. For example, the Host Groups below displays all the services for both the `FooMe` and `FooYou` clusters.



Service Overview For All Host Groups

ML-FooM (FooMe)

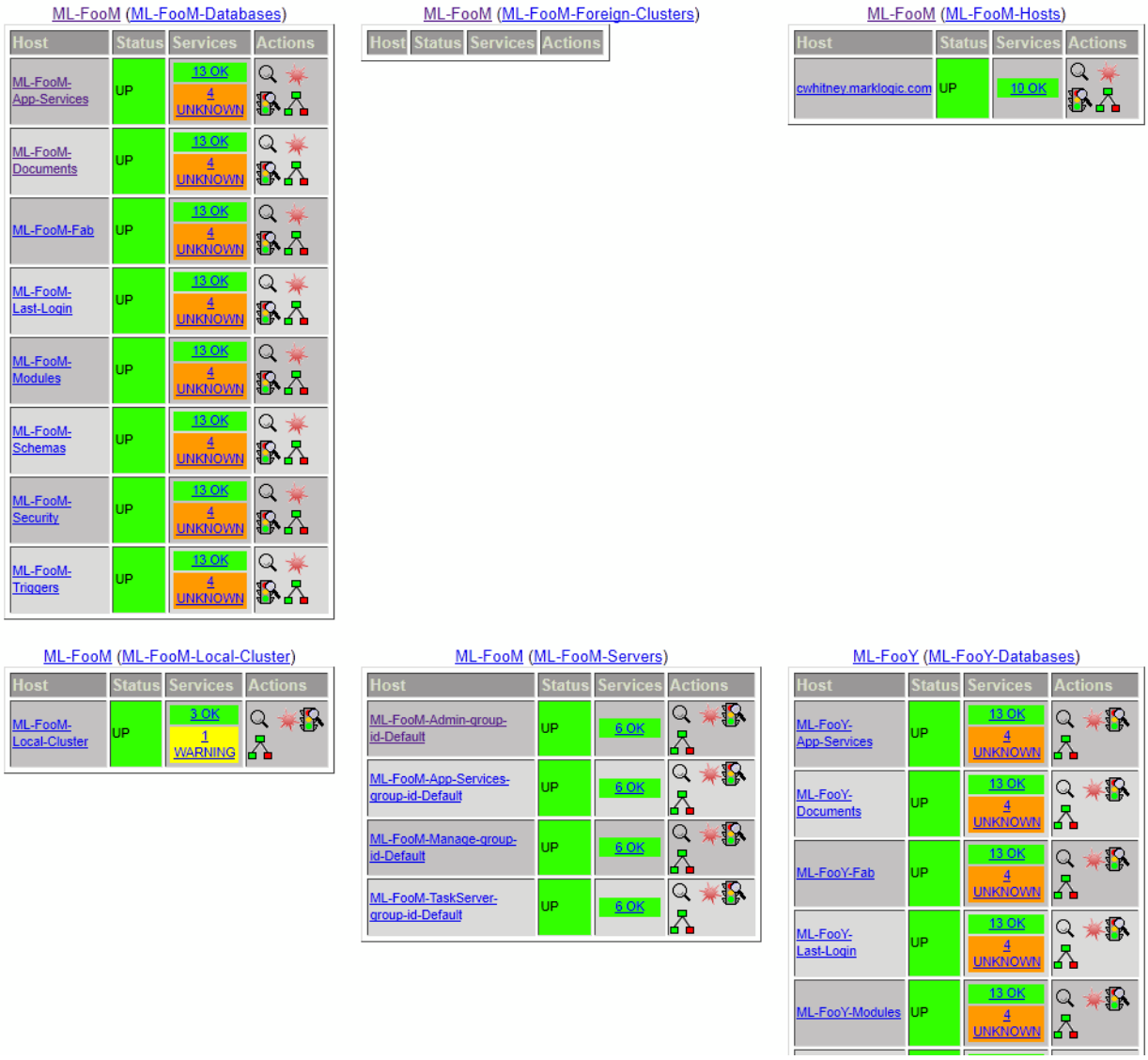
Host	Status	Services	Actions
ML-FooM-Admin-group-id-Default	UP	6 OK	
ML-FooM-App-Services	UP	13 OK 4 UNKNOWN	
ML-FooM-App-Services-group-id-Default	UP	6 OK	
ML-FooM-Documents	UP	13 OK 4 UNKNOWN	
ML-FooM-Fab	UP	13 OK 4 UNKNOWN	
ML-FooM-Last-Login	UP	13 OK 4 UNKNOWN	
ML-FooM-Local-Cluster	UP	3 OK 1 WARNING	
ML-FooM-Manage-group-id-Default	UP	6 OK	
ML-FooM-Modules	UP	13 OK 4 UNKNOWN	
ML-FooM-Schemas	UP	13 OK 4 UNKNOWN	
ML-FooM-Security	UP	13 OK 4 UNKNOWN	
ML-FooM-TaskServer-group-id-Default	UP	6 OK	
ML-FooM-Triggers	UP	13 OK 4 UNKNOWN	
cwhitney.marklogic.com	UP	10 OK	

ML-FooY (FooYou)

Host	Status	Services	Actions
ML-FooY-Admin-group-id-Default	UP	6 OK	
ML-FooY-App-Services	UP	13 OK 4 UNKNOWN	
ML-FooY-App-Services-group-id-Default	UP	6 OK	
ML-FooY-Documents	UP	13 OK 4 UNKNOWN	
ML-FooY-Fab	UP	13 OK 4 UNKNOWN	
ML-FooY-Last-Login	UP	13 OK 4 UNKNOWN	
ML-FooY-Local-Cluster	UP	3 OK 1 WARNING	
ML-FooY-Manage-group-id-Default	UP	6 OK	
ML-FooY-Modules	UP	13 OK 4 UNKNOWN	
ML-FooY-Schemas	UP	13 OK 4 UNKNOWN	
ML-FooY-Security	UP	13 OK 4 UNKNOWN	
ML-FooY-TaskServer-group-id-Default	UP	6 OK	
ML-FooY-Triggers	UP	13 OK 4 UNKNOWN	
wlan31-13-237.marklogic.com	UP	10 OK	

3.7.3 Service Groups

The Service Groups page displays the services groups by resource type on each cluster. For example, the Service Groups below displays the services for the databases, servers, hosts, and local/foreign clusters in separate groups for both the `FooMe` and `FooYou` clusters.



3.7.4 Service Status Details for a Resource

Below is a detailed view of the services for the Documents database in the `FooMe` cluster. Note that some services may report a status of UNKNOWN. This indicates that the monitoring metric is unavailable to Nagios for that resource. For example, the `database-replication-active` service shown below reports UNKNOWN because Database Replication is not configured for the database. In this example, if you do not have a license for Database Replication, you can remove the `database-replication-active` service from your `services.xml` file. If you have a license for Database Replication, but only have the feature enabled for some databases, you can exclude the databases that are not replicated from being monitored, as described in "The Monitoring Services File" on page 40.

Service Status Details For
Host 'ML-FooM-Documents'

Host	Service	Status	Last Check	Duration	Attempt	Status Information
ML-FooM-Documents	async-replicating	UNKNOWN	10-11-2011 12:08:56	0d 23h 10m 3s	10/10	The XML element (async-replicating) was not present in the data returned by the API (curl "http://cwhitney:8002/manage/v1/databases/Documents/status" --max-time 10 --user [username:pw] -i --anyauth -H "Accept: application/xml" --silent 2>&1)
	backup-count	OK	10-11-2011 12:10:36	0d 4h 51m 23s	1/10	OK - Documents-backup-count=0
	compressed-tree-cache-hit-rate	OK	10-11-2011 12:10:11	0d 4h 51m 48s	1/10	OK - Documents-compressed-tree-cache-hit-rate=0
	compressed-tree-cache-miss-rate	OK	10-11-2011 12:10:48	0d 4h 51m 11s	1/10	OK - Documents-compressed-tree-cache-miss-rate=0
	database-replication-active	UNKNOWN	10-11-2011 12:01:26	0d 23h 7m 33s	10/10	XML element (database-replication-active) was not present in the data returned by the API
	documents	OK	10-11-2011 12:04:04	0d 20h 6m 55s	1/10	OK - Documents-documents=0
	failed-masters	UNKNOWN	10-11-2011 12:02:41	0d 23h 6m 18s	10/10	The XML element (failed-masters) was not present in the data returned by the API (curl "http://cwhitney:8002/manage/v1/databases/Documents/status" --max-time 10 --user [username:pw] -i --anyauth -H "Accept: application/xml" --silent 2>&1)
	foreign-forests-lag-exceeded	UNKNOWN	10-11-2011 12:03:18	0d 23h 5m 41s	10/10	The XML element (foreign-forests-lag-exceeded) was not present in the data returned by the API (curl "http://cwhitney:8002/manage/v1/databases/Documents/status" --max-time 10 --user [username:pw] -i --anyauth -H "Accept: application/xml" --silent 2>&1)
	list-cache-hit-rate	OK	10-11-2011 12:09:56	0d 4h 51m 3s	1/10	OK - Documents-list-cache-hit-rate=0
	list-cache-miss-rate	OK	10-11-2011 12:10:36	0d 4h 51m 23s	1/10	OK - Documents-list-cache-miss-rate=0
	load-detail	OK	10-11-2011 12:10:08	0d 4h 51m 51s	1/10	OK - Documents-total-query-read-load=0
	merge-count	OK	10-11-2011 12:10:42	0d 4h 51m 17s	1/10	OK - Documents-merge-count=0
	on-disk-size	OK	10-11-2011 12:10:19	0d 18h 30m 40s	1/10	OK - Documents-on-disk-size=0MB
	rate-detail	OK	10-11-2011 12:09:57	0d 4h 51m 2s	1/10	OK - Documents-total-query-read-rate=0
	reindex-count	OK	10-11-2011 12:01:35	0d 20h 19m 24s	1/10	OK - Documents-reindex-count=0
	restore-count	OK	10-11-2011 12:02:12	0d 18h 38m 47s	1/10	OK - Documents-restore-count=0
	state	OK	10-11-2011 12:10:50	0d 4h 51m 9s	1/10	OK - state=1 [critical(@0:0)]

4.0 Configuring the MarkLogic Smart Plug-in for HP Operations Manager

HP Operations Manager is a software solution designed to help administrators monitor, recognize, solve, and prevent problems reported by applications, networks, and systems. HP Operations Manager uses the MarkLogic Smart Plugin (SPI) to monitor, recognize, solve, and prevent problems reported by MarkLogic Cluster components.

This chapter provides information about installing, configuring, managing, and uninstalling MarkLogic SPI. This chapter contains the following topics:

- Overview
- Terms Used in this Chapter
- Smart Plug-in for HP Operations Manager Requirements
- Installing MarkLogic Smart Plugin
- Configuring the MarkLogic Smart Plug-in
- Uninstalling the MarkLogic Smart Plugin
- Configuration Files
- MarkLogic Default Policies
- Modifying Policies and Creating New Policies
- Reconfiguring HP Operations Manager After Changes to MarkLogic Server
- Performance Data Source

4.1 Overview

The MarkLogic SPI consists of a set of extensions to and customizations of HP Operations Manager, which allow for quick and seamless integration with MarkLogic. The SPI monitors the status and collects performance data of the following MarkLogic components:

- MarkLogic clusters
- MarkLogic databases
- MarkLogic application servers
- MarkLogic hosts

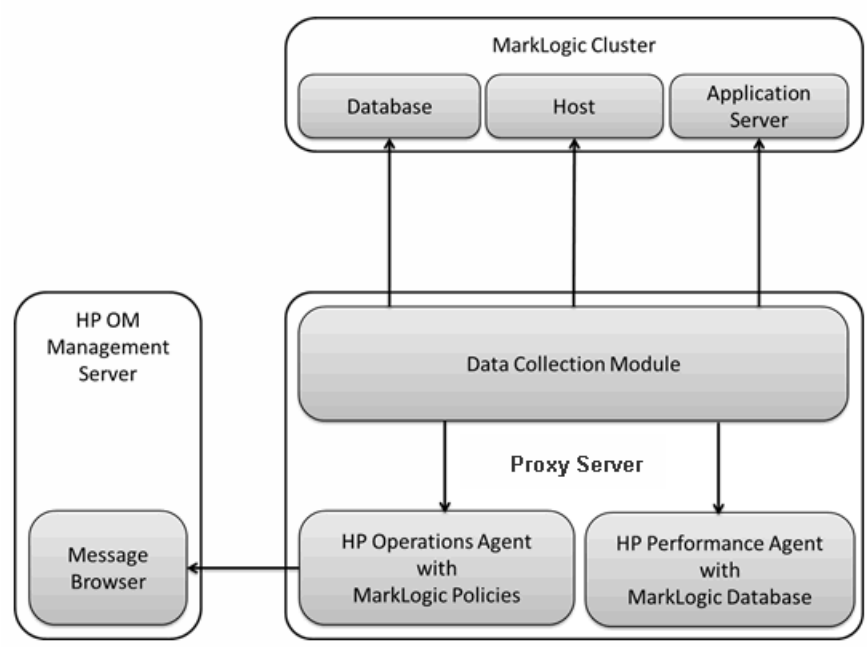
HP Operations Manager reports only those MarkLogic components statuses that are relevant to the administrator, which are defined by the policies described in “MarkLogic Default Policies” on page 92.

In environments where MarkLogic components are distributed across several systems, the SPI has the following advantages:

- Centralized monitoring of MarkLogic components status
- Real-time monitoring of MarkLogic components status

A key component of the MarkLogic SPI is the Data Collection Module (DCM), which queries MarkLogic Clusters for monitored information. The DCM passes the gathered information to the HP Operations agent, which filters the gathered information based on the MarkLogic SPI policies. The HP Operations agent then passes the filtered information to HP Operations Manager message browser to display as HP Operations messages.

A proxy server is a computer system in a network that hosts the MarkLogic SPI. The following figure shows the general architecture of a proxy server set up to monitor a MarkLogic cluster and pass the data to an HP Operations Manager message browser.



The MarkLogic SPI includes the following extensions:

- Information sources to access and relay information about the status of MarkLogic components to HP Operations Manager
- Policies defining which MarkLogic software statuses are to be reported and the granularity of monitored objects

The policies are implemented as monitor threshold policy types.

4.2 Terms Used in this Chapter

- *SPI* — Short for Smart Plug-in.
- *Proxy Server* — The server that hosts the MarkLogic SPI. The Proxy Server is identified in the HP Operations Manager as the *Management Server.*
- *MarkLogic SPI Operator User Role* — MarkLogic SPI installs a user role with the privileges to view and administer messages belonging to the MarkLogic SPI message group.
- *MarkLogic Node Group* — An HP Operations Manager node group contains several managed nodes which belong together. For example, if a policy monitors managed nodes for the same type of resource, the managed nodes can be grouped into a node group. During installation, you apply a set of policies to a node group. A single managed node can belong to several node groups.

  MarkLogic SPI installs the MarkLogic node group.

- *MarkLogic Service Tree* — An HP Operations Manager service contains several sub-services that belong together. A service status is recalculated when a new message arrives in the message browser. The MarkLogic SPI installs the MarkLogic service, which is a root for all MarkLogic cluster services. The MarkLogic discovery probes add a subservice for each discovered MarkLogic cluster.
- *MarkLogic Message Groups* — An HP Operations Manager message group contains messages that belong to the same task or have some logical connection. For example, if messages relate to the same type of managed nodes or to a certain domain, they can be grouped into a message group. The MarkLogic SPI installs the MarkLogic SPI message group.
- *MarkLogic Policy Groups* — An HP Operations Manager policy specifies the collected or monitored messages and values, scheduled actions, filters to expose or suppress messages, and the logging options used after messages and values interception.

  The MarkLogic SPI installs following policy groups to monitor MarkLogic:

  - Data Collection
  - Dynamic Resource Allocation
  - Internal Messages
  - MarkLogic Health Monitor
  - Resource Availability

  The policies in these groups are described in “MarkLogic Default Policies” on page 92.

4.3 Smart Plug-in for HP Operations Manager Requirements

The computer on which you install the Smart Plug-in for HP Operations Manager must conform to the following requirements:

- Microsoft Windows Server 2008 R2

Additionally, before you install the MarkLogic SPI, install and configure the following products on your supported platform:

- HP Operations Manager Management Server.
- HP Operations Agent on the system which will be used as MarkLogic SPI Proxy.
- ActiveState Perl 5.8 or later must be installed on all MarkLogic SPI Proxies.
- CPAN Log-log4perl Perl module must be installed on MarkLogic SPI Proxies because ActiveState Perl standard module repository does not contain it.
- The MarkLogic Cluster you want to monitor.

4.4 Installing MarkLogic Smart Plugin

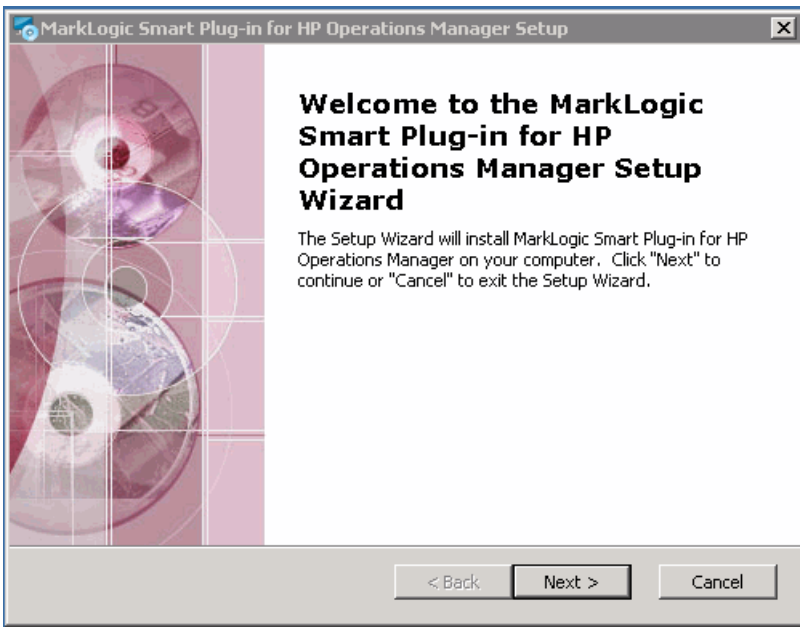
This section describes how to install the MarkLogic SPI for HP Operations Manager.

> **Note:** All text is case-sensitive.

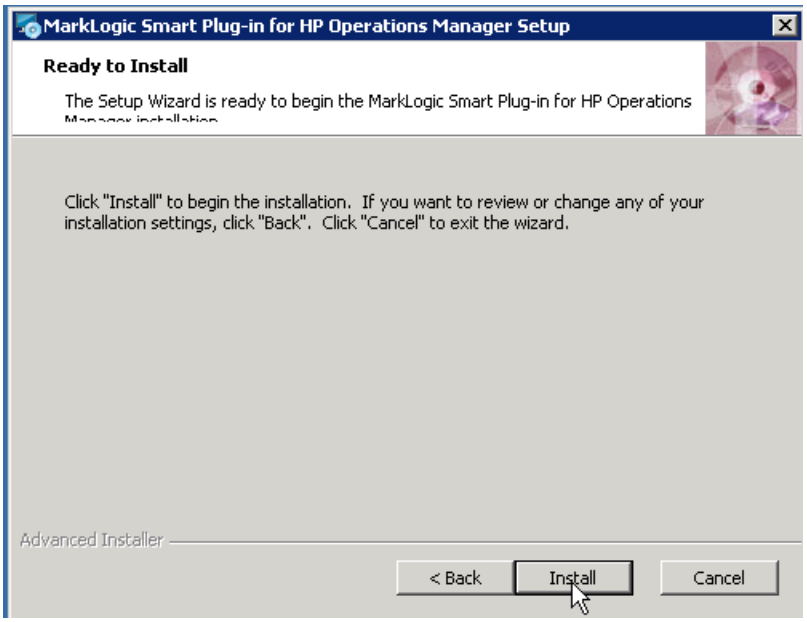
You must have the `manage-user` role on the MarkLogic Server cluster to be monitored in order to complete this procedure.

1. Download the MarkLogic Smart Plugin for HP Operations Manager package from the MarkLogic Support site.

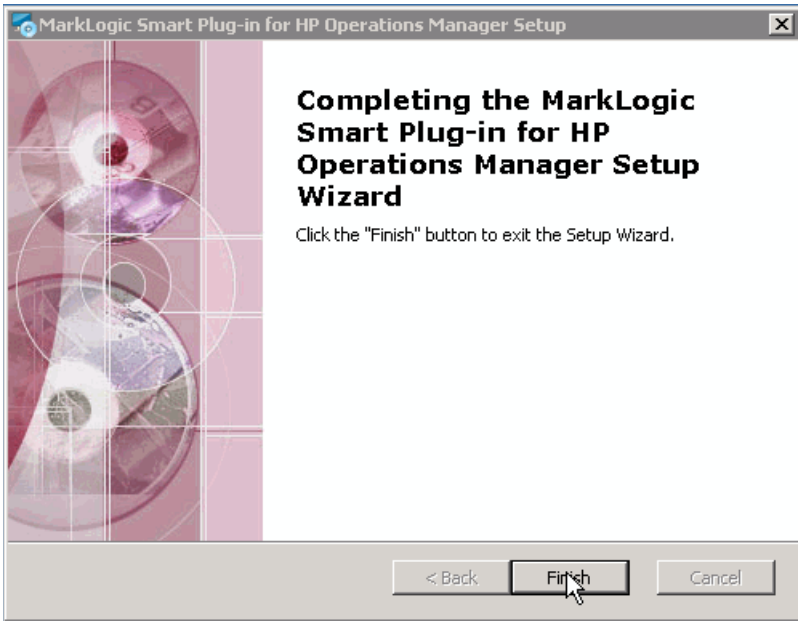
2. Locate the `MarkLogic-HPOM-SPI-1.0-1.exe` file and double click to launch the Setup Wizard. Click Next.



3. In the Ready to Install window, click Install.



4. When the Setup Wizard is done, click Finish.



4.5 Configuring the MarkLogic Smart Plug-in

The configuration procedure consists of the following steps:

To configure the MarkLogic Smart Plug-in, complete the following steps:

- Edit the Configuration File
- Enabling SSL between the Proxy Server and MarkLogic Monitor Host
- Add MarkLogic Proxy nodes to the Node Group
- Deploy MarkLogic Smart Plug-in Instrumentation
- Launch the MarkLogic SPI Setup Tools
- Deploy the MarkLogic Policies to the Plugin Proxy

4.5.1 Edit the Configuration File

Locate the `%OvShareDir%\Instrumentation\Categories\MarkLogicSPI\mlspi_config.xml` file and edit to change the MarkLogic cluster name, hostname, username and password. You can also exclude specific resources from being monitored. For details, see "Modifying Policies and Creating New Policies" on page 96.

For example, if your cluster name is `Master`, host name is `rh5-intel64-18`, and the login credentials for the user on the MarkLogic Server to be monitored are `usr`/`pwd`, the `mlspi_config.xml` file would look like the following (changes displayed in **bold**):

```
<mlspi>
  <global>
    <polling file="mlspi_urls.xml" />
    <excludes>
      <exclude resourcetype="database"name="MyDatabase" />
      <exclude resourcetype="server" name="MyServer" />
      <exclude resourcetype="host" name="myother.marklogic.com" />
    </excludes>
  </global>
  <clusters>
    <cluster>
      <name>Master</name>
      <hostname>rh5-intel64-18</hostname>
      <port>8002</port>
      <ssl>0</ssl>
      <certificate></certificate>
      <username>usr</username>
      <password>pwd</password>
      <excludes>
        <exclude resourcetype="database"name="MyOtherDatabase" />
        <exclude resourcetype="server" name="MyOtherServer" />
        <exclude resourcetype="host" name="myoldhost.marklogic.com" />
      </excludes>
    </cluster>
  </clusters>
</mlspi>
```

4.5.2 Enabling SSL between the Proxy Server and MarkLogic Monitor Host

The procedure in this section describes how to enable SSL encryption between the HP Proxy Server and the MarkLogic Server Monitor Host named `rh5-intel64-18`.

The procedure for enabling SSL encryption using the Admin interface is as follows:

1. Create an SSL certificate template and add it to the `Manage` AppServer to enable SSL, as described in Procedures for Enabling SSL on App Servers in the *Administrator's Guide*.

2. Test the SSL connection by entering the following URL in your browser and accepting the security exception:

```
https://rh5-intel64-18:8002/manage/v1
```

3. Use the following procedure to download the certificate to your file system:

   a. Click the Security icon in the left frame.

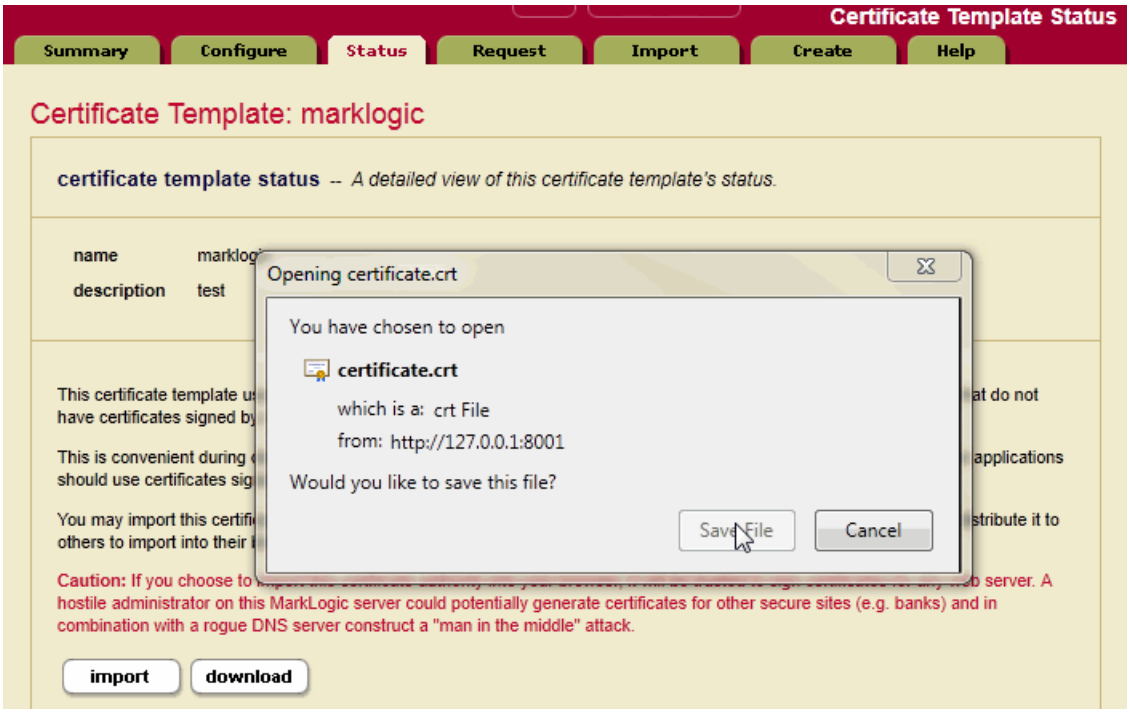
   b. Click the Certificate Templates icon on the left tree menu.

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

   d. Click the Status tab to display the certificate template Status page.

   e. In the certificate template status section of the Status page, click on Download.

   f. Click Save File in the pop-up window.



4. Copy the certificate to a directory in your proxy server. If you have multiple proxy servers, the certificate must be copied to the same location on each proxy server.

5. Edit the `mlspi_config.xml` on the management server to enable SSL (set the value of `<ssl>` to 1) and to provide the pathname to the certificate file:

```
<ssl>1</ssl>
<certificate>c:\mypath\certificate.crt</certificate>
```

6. Deploy instrumentation as described in “Deploy MarkLogic Smart Plug-in Instrumentation” on page 69 and launch setup tools 01 and 02, as described in “Launch the MarkLogic SPI Setup Tools” on page 71.

If you encounter any problems, you can debug by invoking the following cURL command on your proxy server:

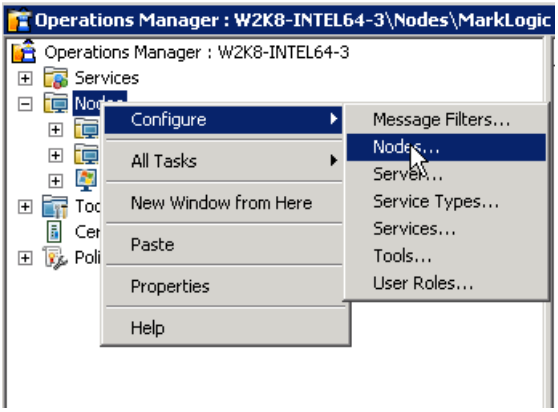
```
cd %OvShareDir%\Instrumentation\Categories\MarkLogicSPI\Windows\x64

curl.exe "https://rh5-intel64-18:8002/manage/v1" --user usr:pwd
--max-time 120  -i --anyauth -H "Accept: application/xml" --cacert
c:\mypath\certificate.crt
```

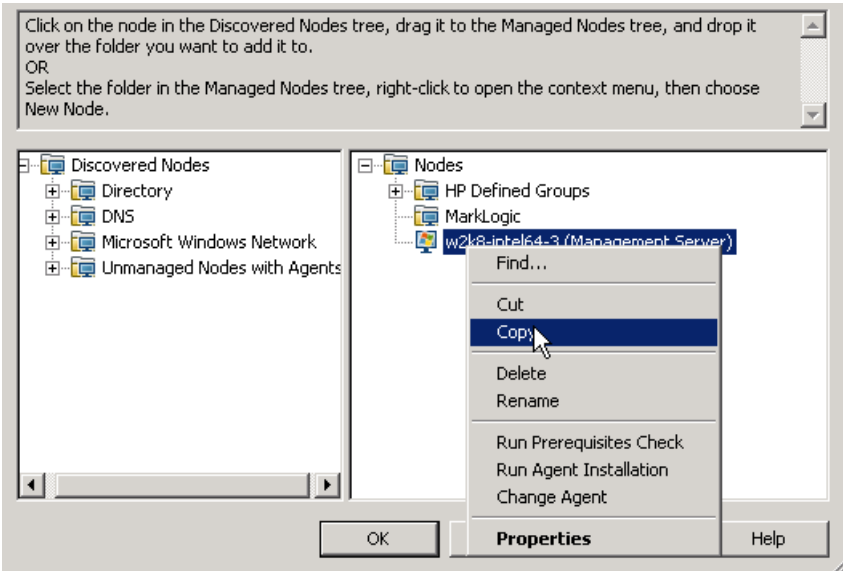
4.5.3 Add MarkLogic Proxy nodes to the Node Group

The procedure described in this section adds the MarkLogic SPI proxy to the HP Operations Manager Node Group. This procedure describes the simplest scenario, where the SPI is installed on the same server as the message browser.

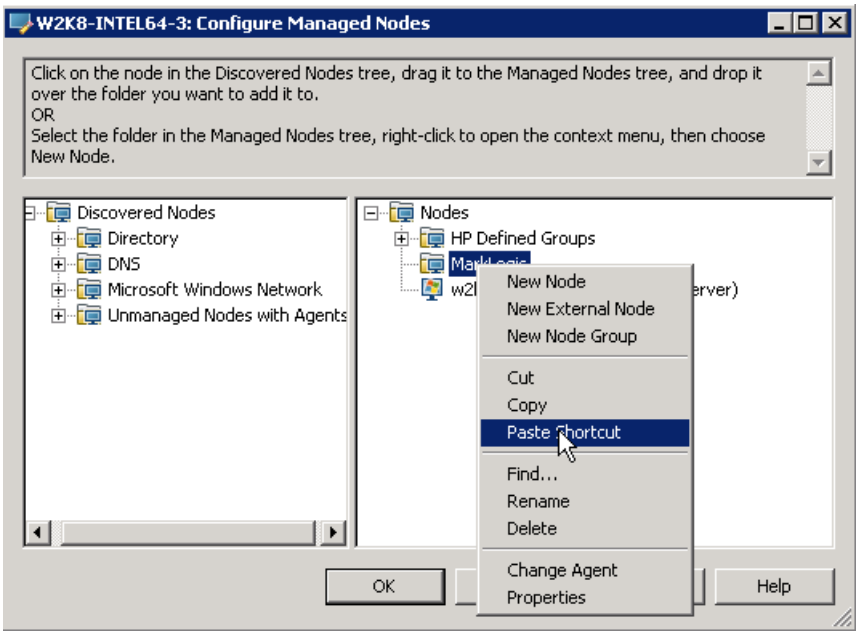
1. Right-click on Nodes in the left-hand navigation window and select Configure > Nodes.



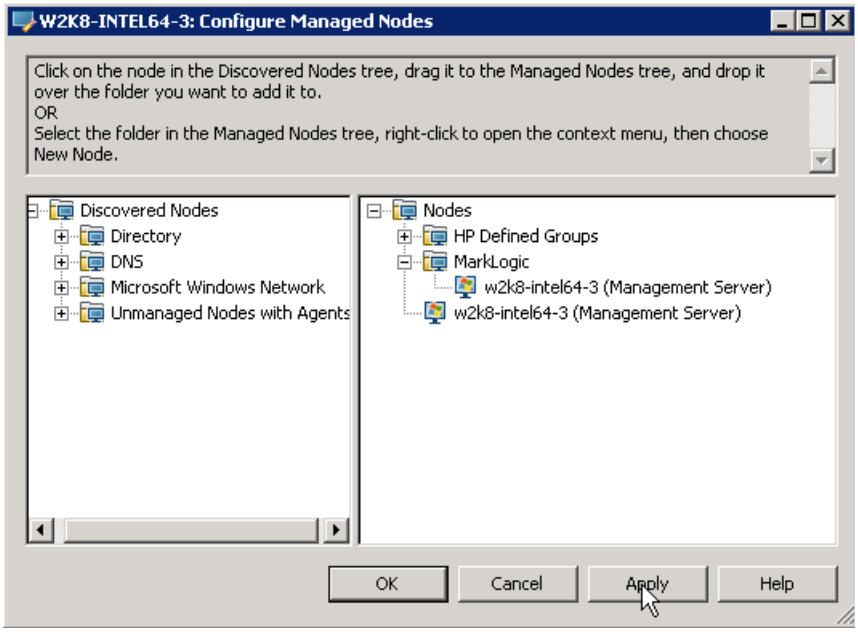
2. Locate the proxy server (in this setup, this is the Management Server) in the right-hand window. Right click and select Copy.



3. In the right-hand window, right click on MarkLogic and select Paste Shortcut.



4. When complete, the proxy server appears under the MarkLogic node. Click OK.

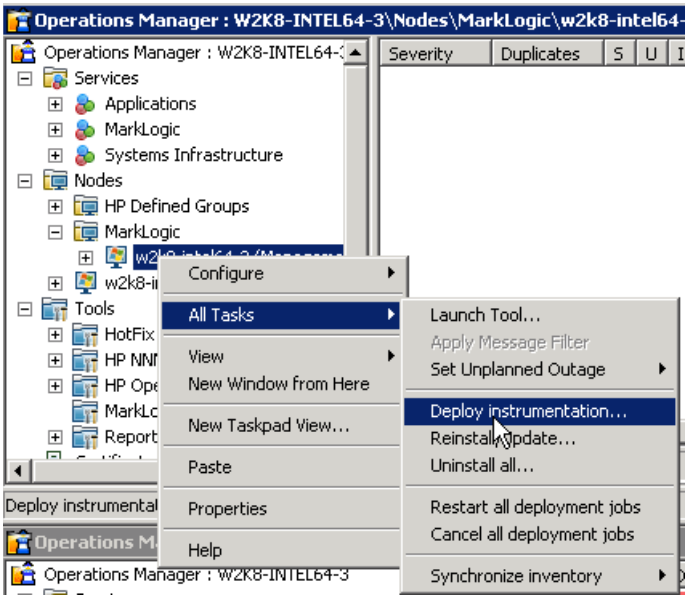


4.5.4 Deploy MarkLogic Smart Plug-in Instrumentation

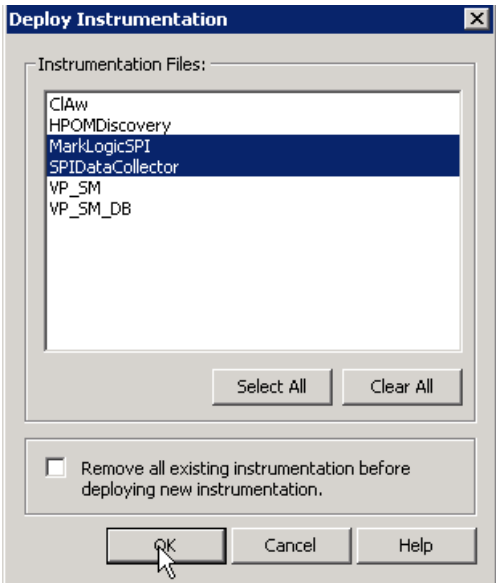
MarkLogicSPI category group contains the data collection module files and the SPI configuration for deployment to the proxies. The MarkLogicSPI depends on the SPIDataCollector, which contains everything required for interacting with the HP Performance Agent.

The procedure described in this section deploys the MarkLogicSPI and SPIDataCollector to the proxy server.

1. Navigate to Nodes > MarkLogic and locate the server that you want to use as the proxy. Right click on the server name and select All Tasks > Deploy Instrumentation.



2. In the Deploy Instrumentation window select MarkLogicSPI and SPIDataCollector. Click OK.

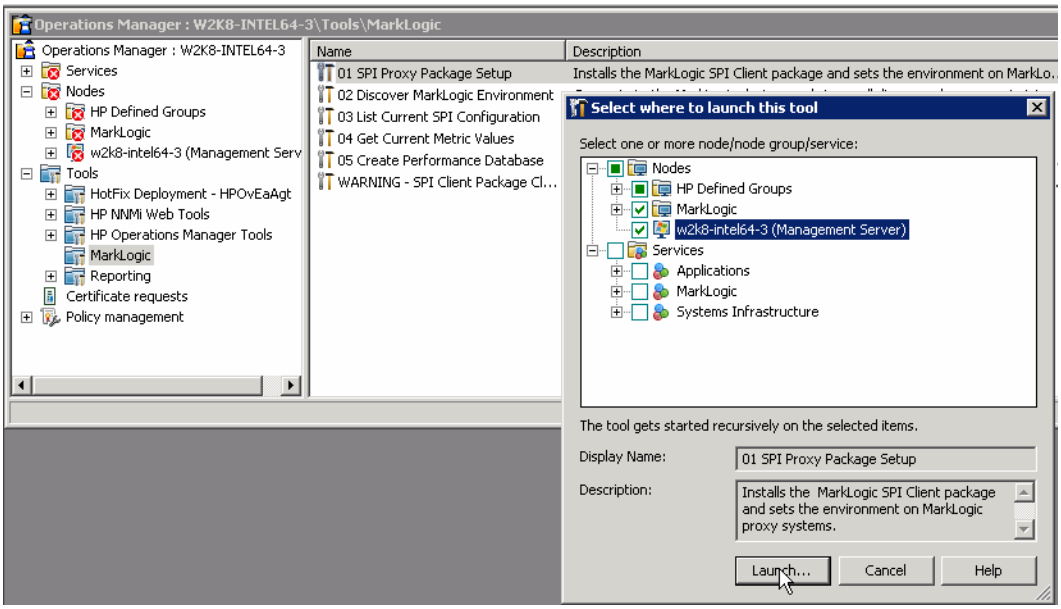


4.5.5 Launch the MarkLogic SPI Setup Tools

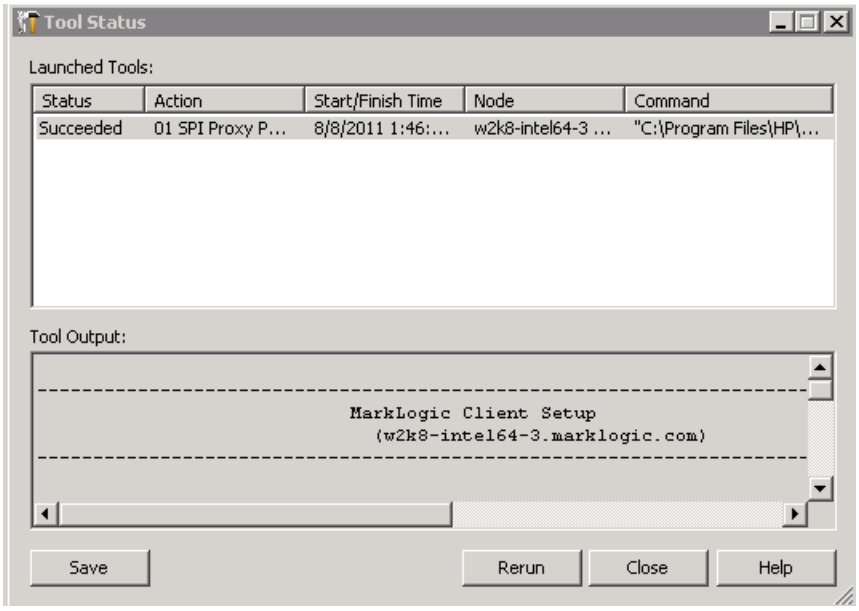
The setup tools run in this section do the following:

- 01 Proxy Package Setup — Prepares the data collection module configuration.
- 02 Discover MarkLogic Environment — Discovers the MarkLogic Server Clusters and locates the resources to be monitored. This tool executes discovery probes on all of the proxy servers. The Tool Output window will display the number of discovered MarkLogic Server resources and clusters. When the discovery is finished, a new MarkLogic Cluster subservice appears in the Operations Manager service tree.
- 03 List Current SPI Configuration — Lists the discovered MarkLogic resources and metrics. Resources that are explicitly excluded in the `mlspi_config.xml` file are not included in the list. For details on the `mlspi_config.xml` file, see “mlspi_config.xml” on page 88.
- 04 Get Current Metric Values — Lists the metrics for all previously discovered MarkLogic resources.
- 05 Create Performance Database — Creates the performance database used by HP Operations Manager to store the performance data. Any data in the database will be overwritten by this tool.

1. Navigate to Tools in the left-hand menu, locate MarkLogic. Double click on MarkLogic. In the right-hand window, click on 01 SPI Proxy Package Setup. In the dialog window, under Nodes, check the box next to the proxy server(s). Click Launch.

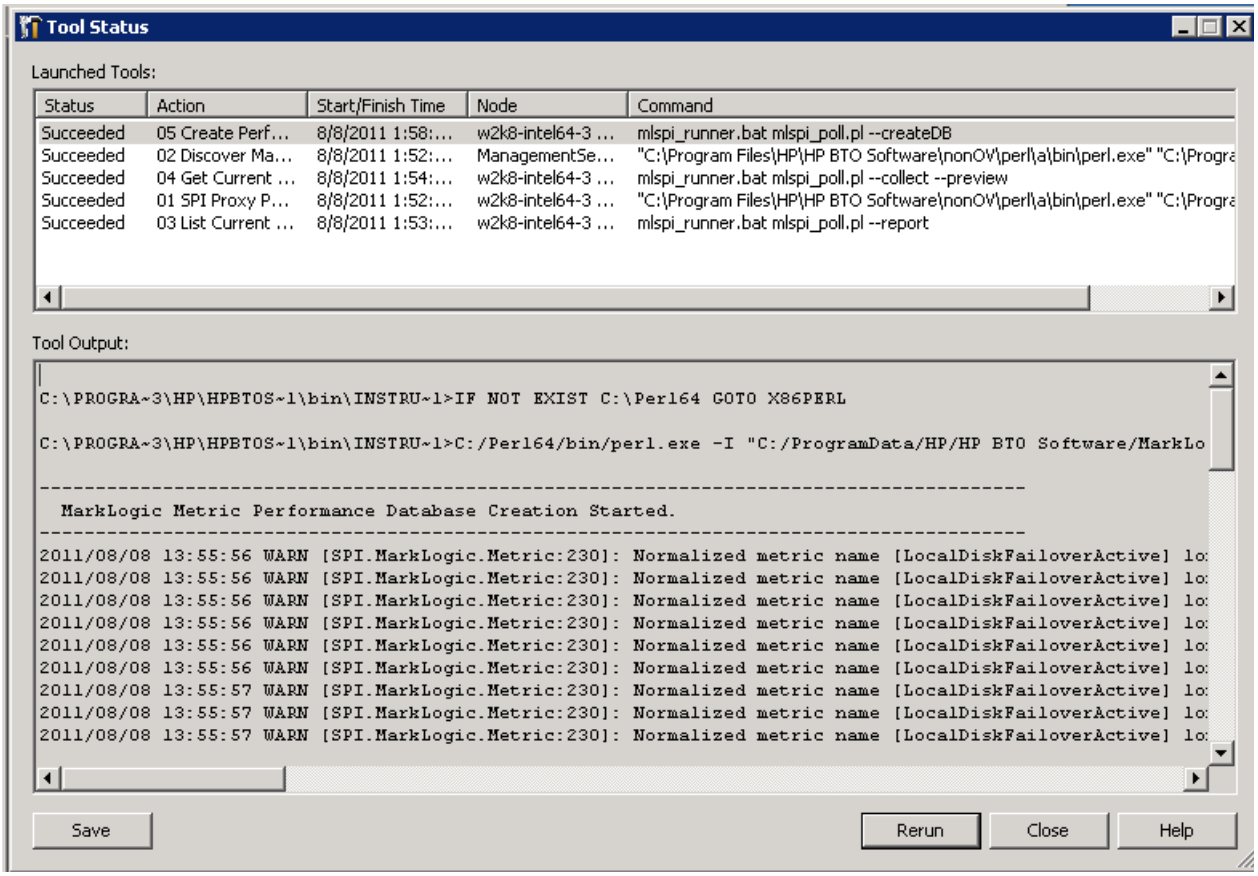


2. If the tool completes successfully, you will see the Succeeded status in the Tool Status window.



3. Repeat Steps 1 and 2 for tools 02 and 05. The status of each tool appears in the Tool Output window. When all of the setup steps are complete, select Close to close the dialog.

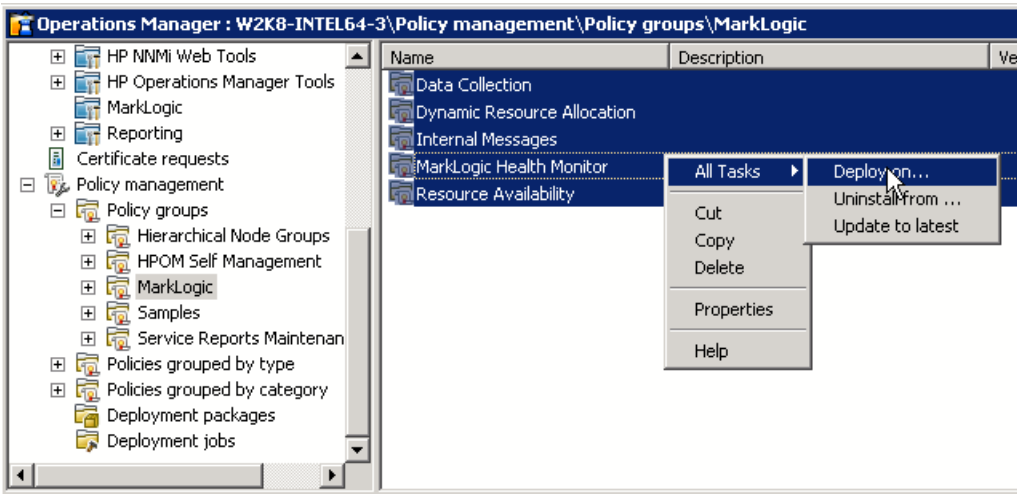
Warning Do not execute the WARNING - SPI Client Package Cleanup tool. The use of this tool is described in “Uninstalling the MarkLogic Smart Plugin” on page 76.



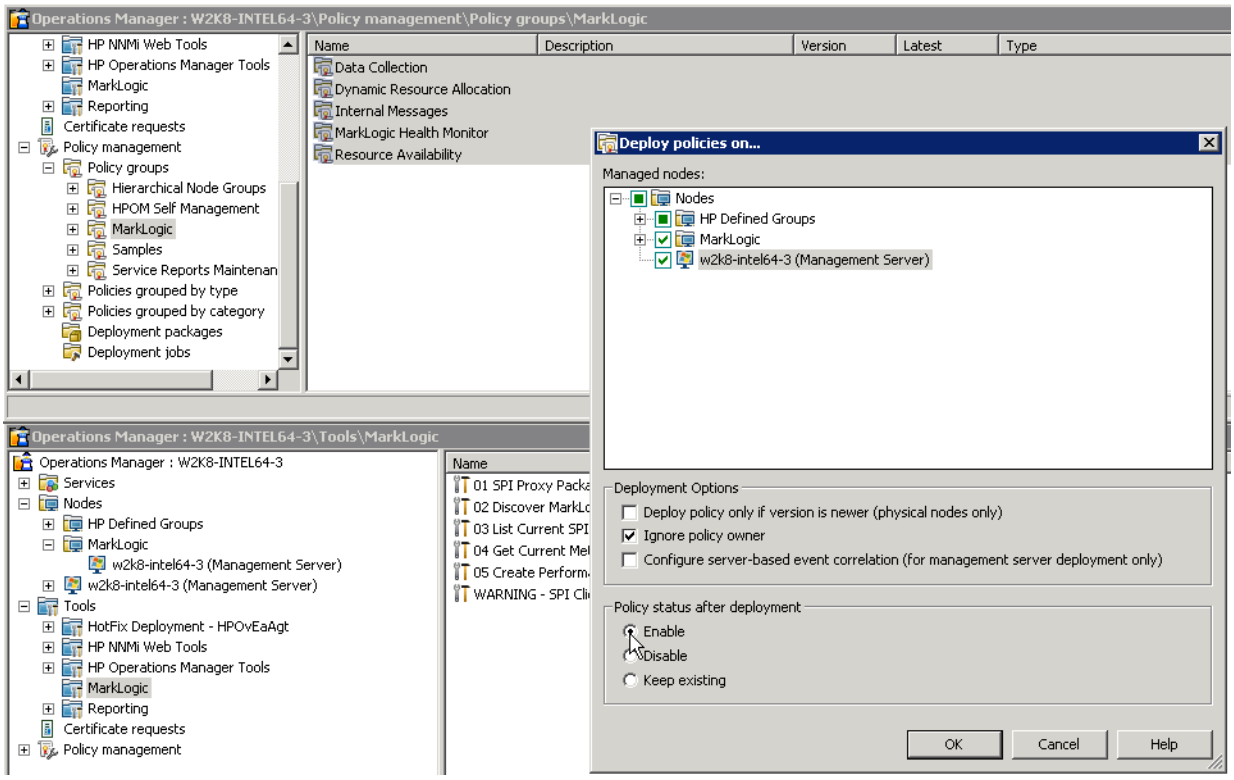
4.5.6 Deploy the MarkLogic Policies to the Plugin Proxy

The procedure described in this section deploys the MarkLogic policies to the plugin proxies and begins collecting monitoring data from the MarkLogic Server cluster.

1. In the left-hand menu, navigate to Policy Management > Policy Groups and select MarkLogic. In the left-hand window, select all of the policy groups. Right click and select All Tasks > Deploy on.



2. In the Deploy Policies On window, select the proxy server. Under Deployment Options, check Ignore Policy Owner (no other options should be checked). Under Policy Status After Deployment, select Enable. Click OK.



Once the policies are deployed, Monitoring data is collected from the MarkLogic Server cluster(s).

4.6 Uninstalling the MarkLogic Smart Plugin

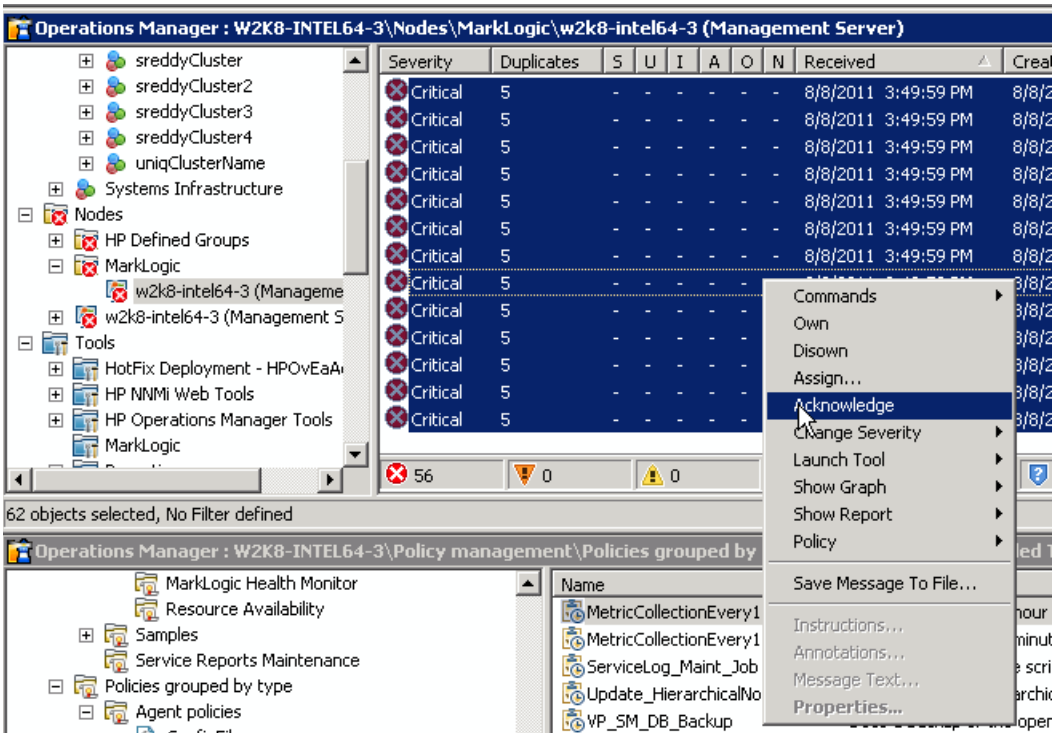
This section describes the procedure for removing the configuration and uninstalling the MarkLogic SPI.

To completely remove MarkLogic SPI configuration, you must complete the following steps:

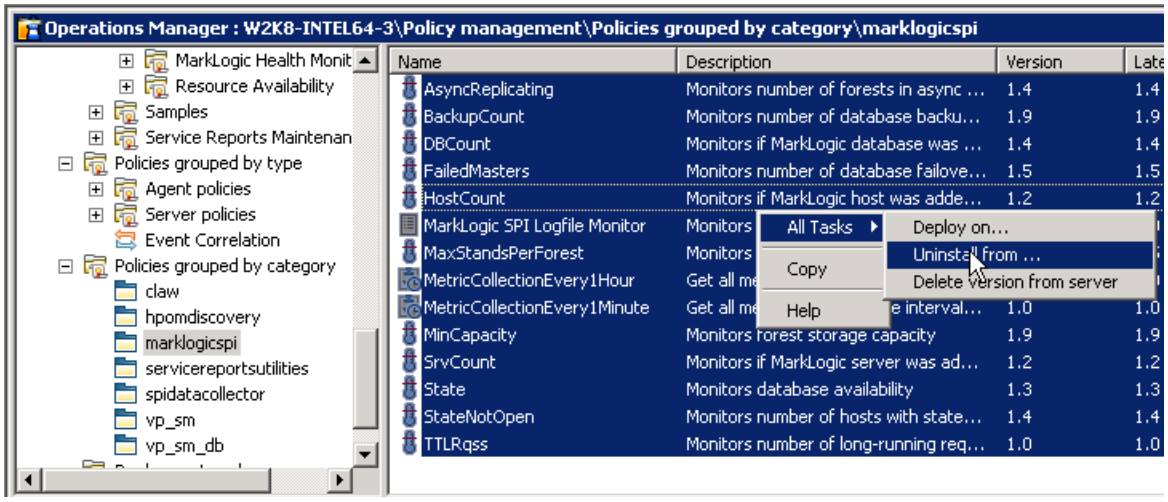
- Acknowledge All MarkLogic Related Messages
- Uninstall all deployed MarkLogic policies from managed nodes
- Delete the MarkLogic policy group
- Delete the orphaned logfile policy
- Delete orphaned monitor threshold policies
- Execute the SPI client package cleanup
- Remove the MarkLogic tools group
- Remove the MarkLogic node group
- Remove the MarkLogic service tree
- Remove the MarkLogicSPI server package

The following procedure describes each step for removing the MarkLogic SPI configuration:

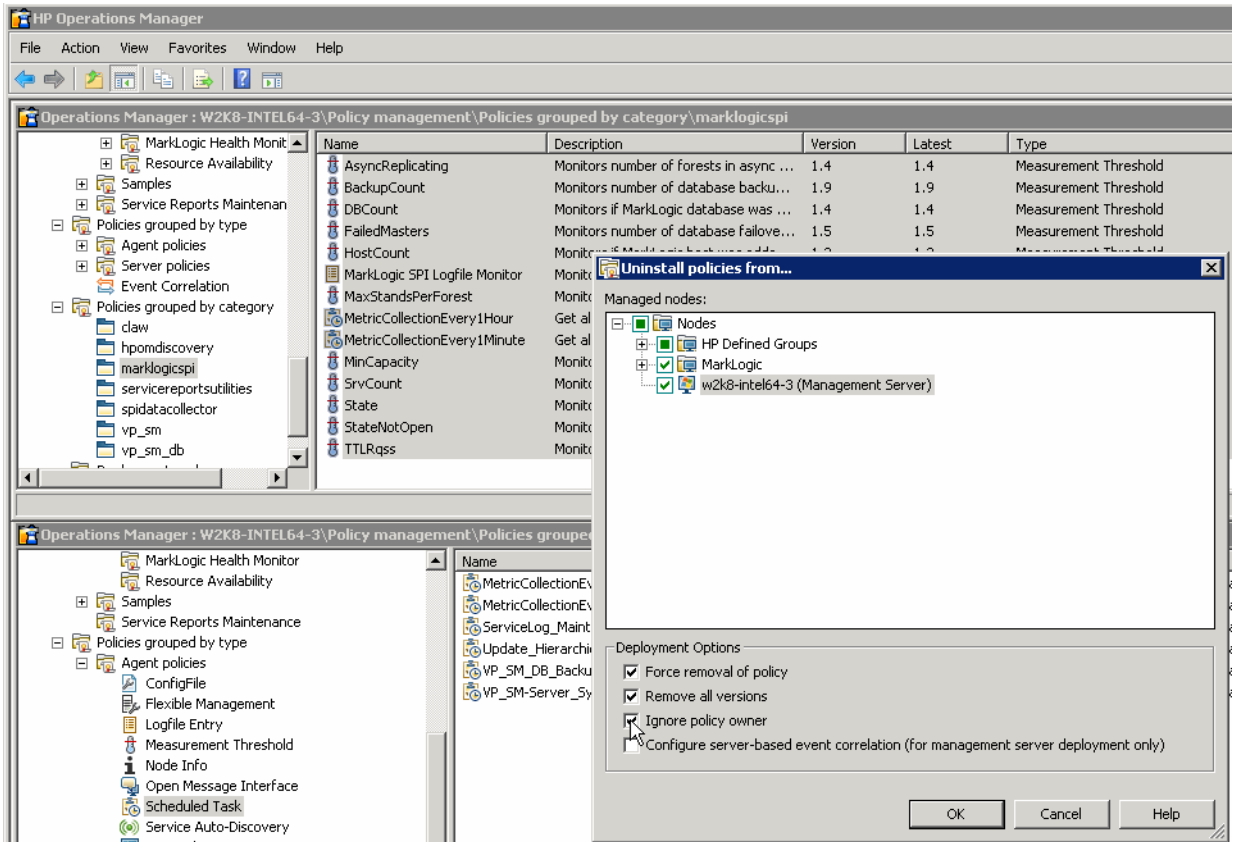
1. In the left-hand menu, navigate to Nodes > MarkLogic and select the proxy server. Select all of the messages in the right-hand menu. Right click and select Acknowledge to remove all of the messages.



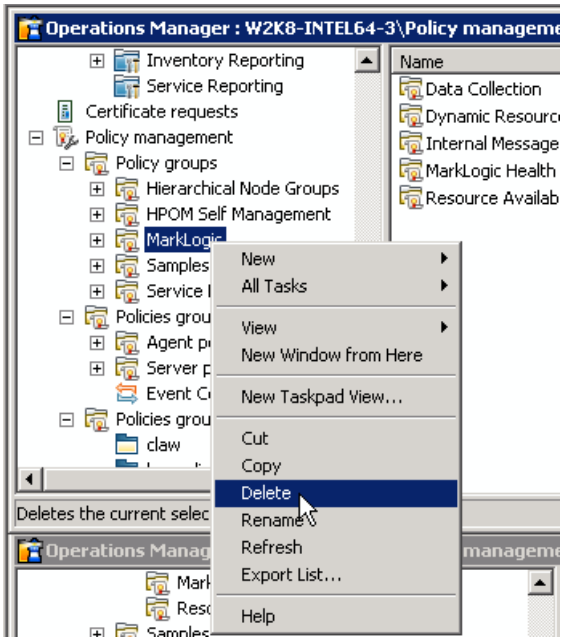
2. In the left-hand menu, navigate to Policies Groups by Category > marklogicspi. Select all of the policy groups in the right-hand window. Right click and select AllTasks > Uninstall From.



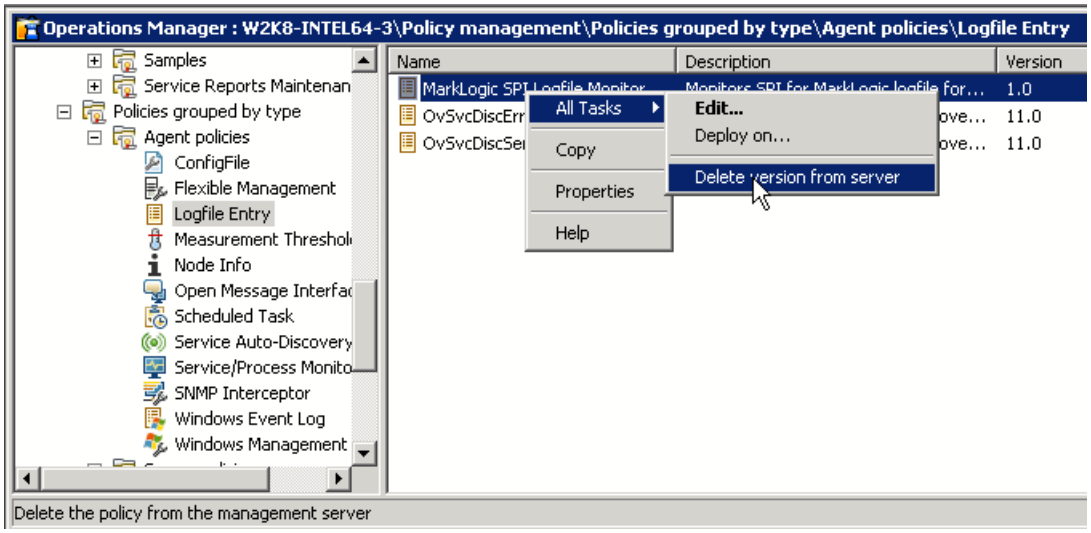
3. In the Uninstall Policies from window, select the proxy server(s). Under Deployment Options, select Force Removal of Policy, Remove all Versions, and Ignore Policy Owner. Click OK.



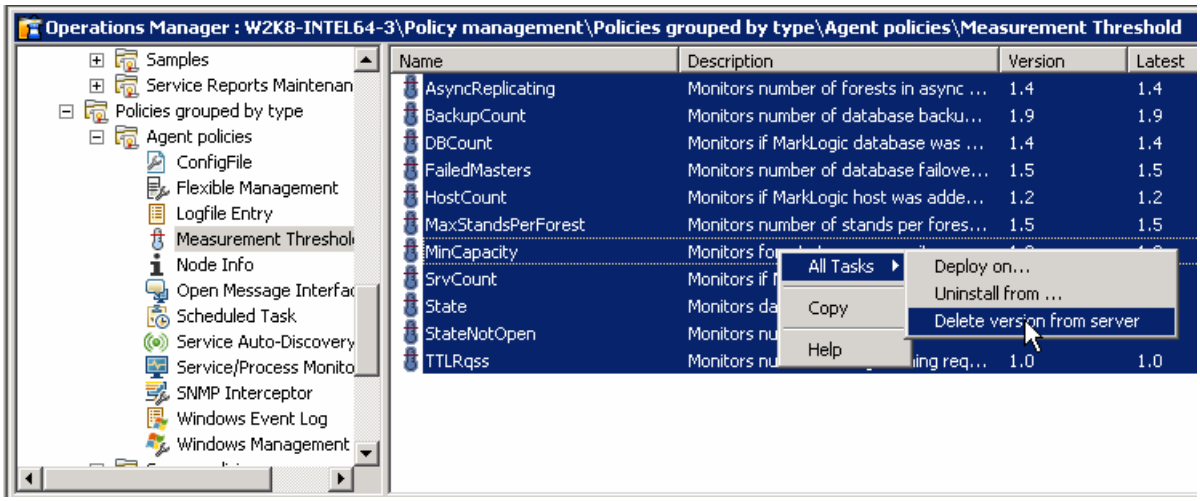
4. In the left-hand menu, navigate to Policy Management > Policies Groups > MarkLogic. Right click on MarkLogic and select Delete to delete the MarkLogic policy group. Click Yes at the prompt.



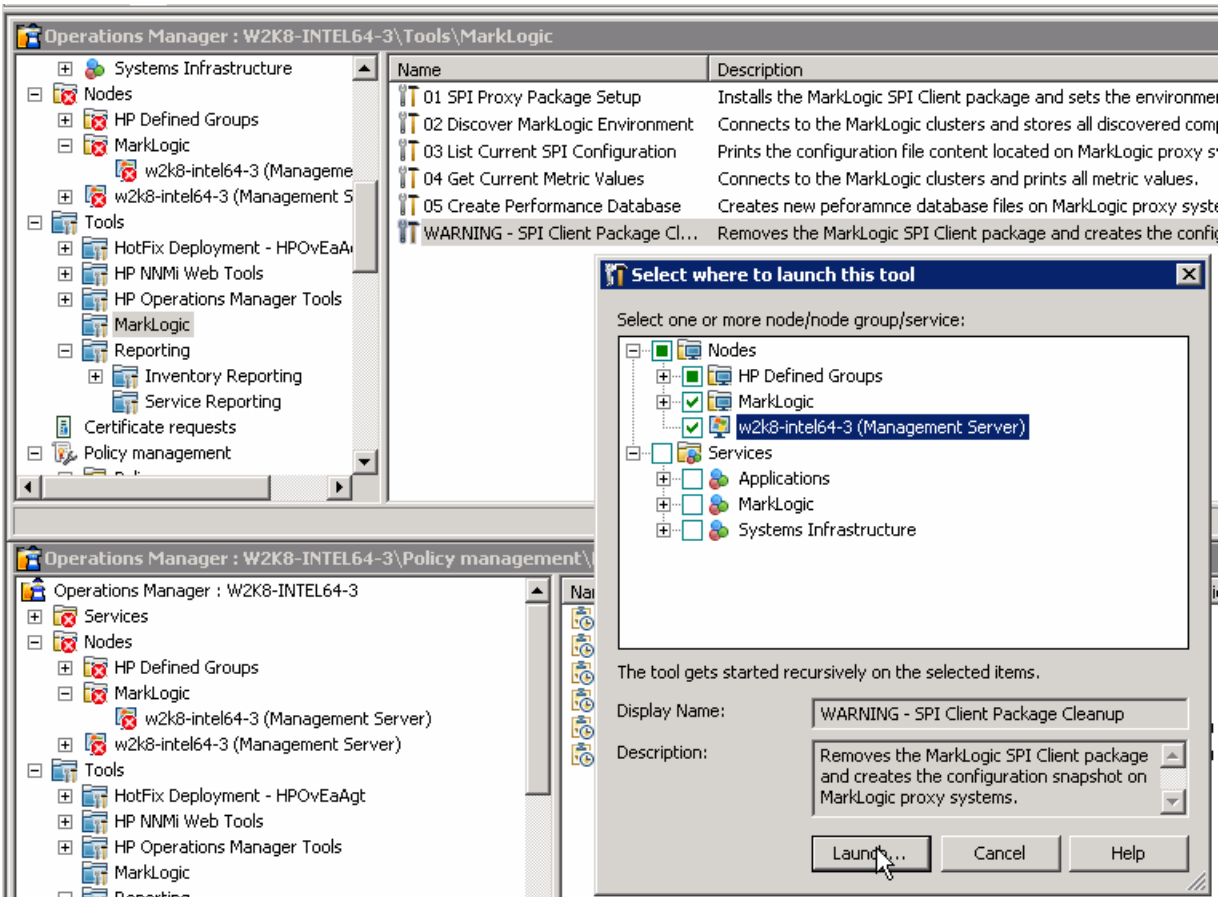
5. In the left-hand menu, navigate to Policies Groups by Type > Agent Policies > Logfile Entry. Select MarkLogic SPI Logfile Monitor in the right-hand window. Right click and select All Tasks > Delete Version from Server.



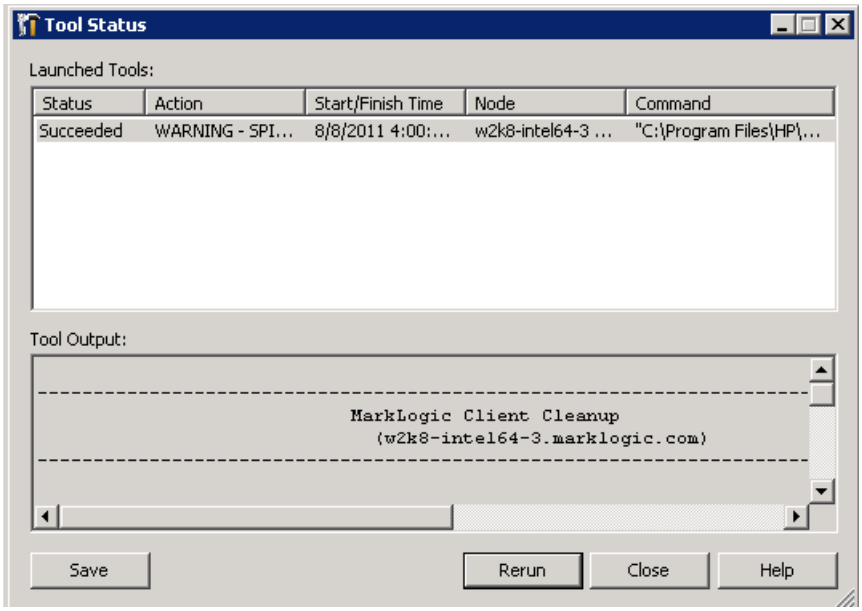
6. In the left-hand menu, navigate to Policies Groups by Type > Agent Policies > Measurement Threshold. Select all of the policy groups in the right-hand window. Right click and select All Tasks > Delete Version from Server.



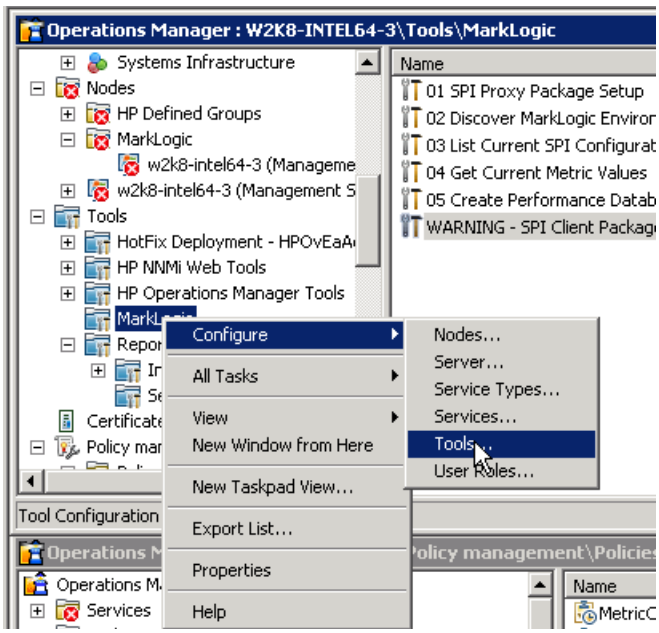
7. In the left-hand menu, navigate to Tools > MarkLogic. In the right-hand menu, click on WARNING - SPI Client Package Cleanup. In the Select Where to Launch This Tool window, select the proxy server(s). Click Launch to remove the MarkLogic Tools Group.



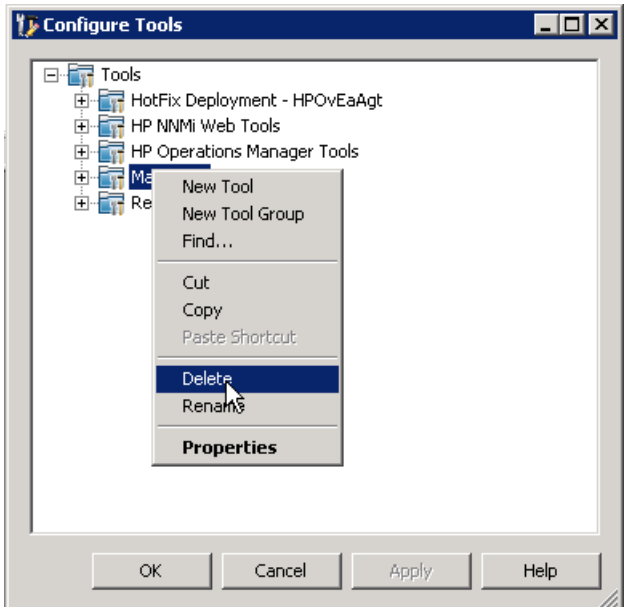
8. If the tool executes successfully, the Tool Status will show Succeeded.



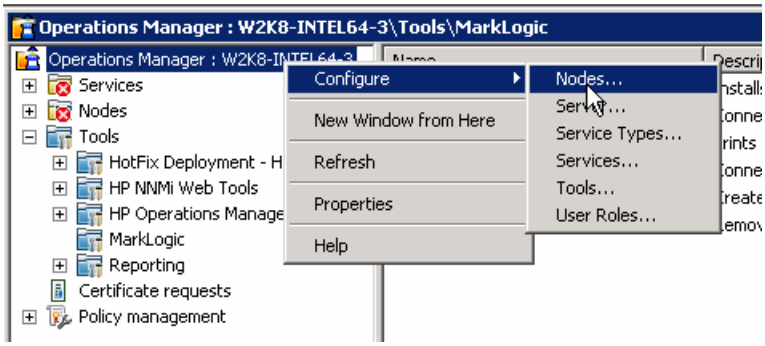
9. In the left-hand menu, navigate to Tools > MarkLogic. Right click on MarkLogic and select Configure > Tools.



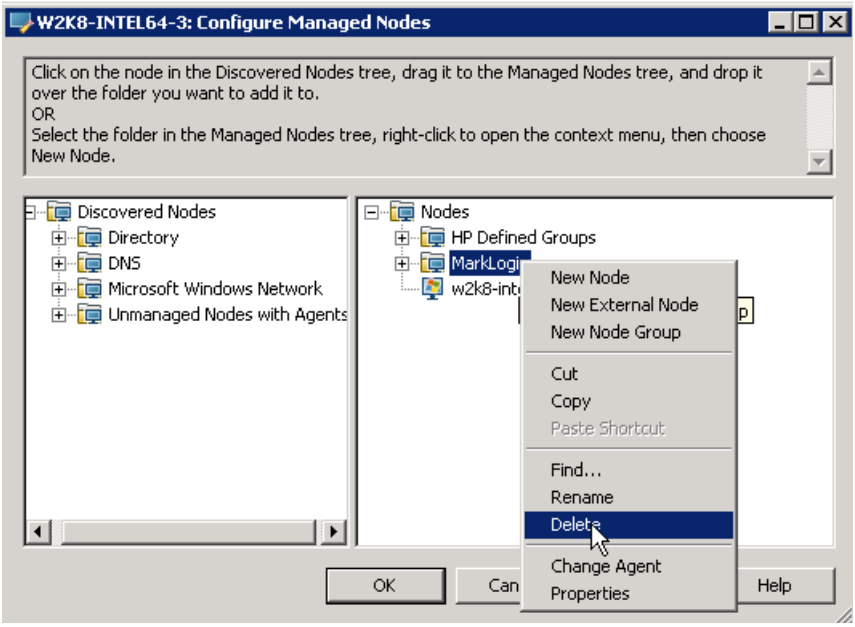
10. In the Configure Tools window, right click on MarkLogic and select Delete. Click OK.



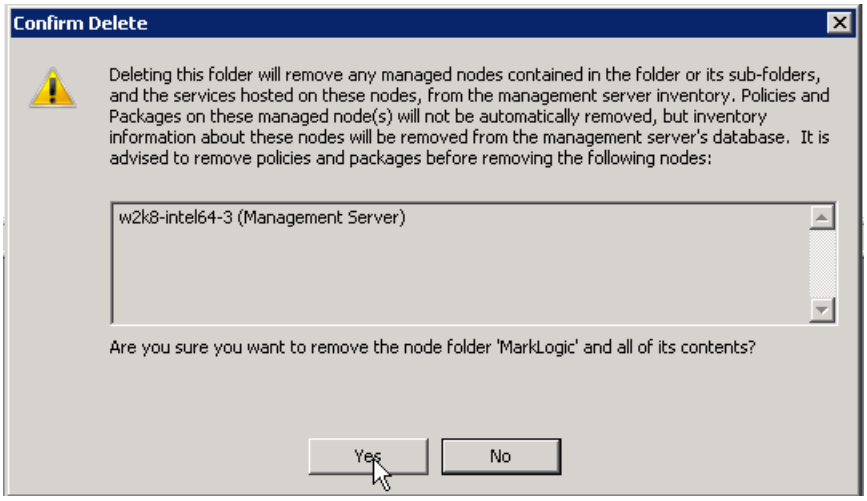
11. In the left-hand menu, right click on Operations Manager *server_name* and select Configure > Nodes.



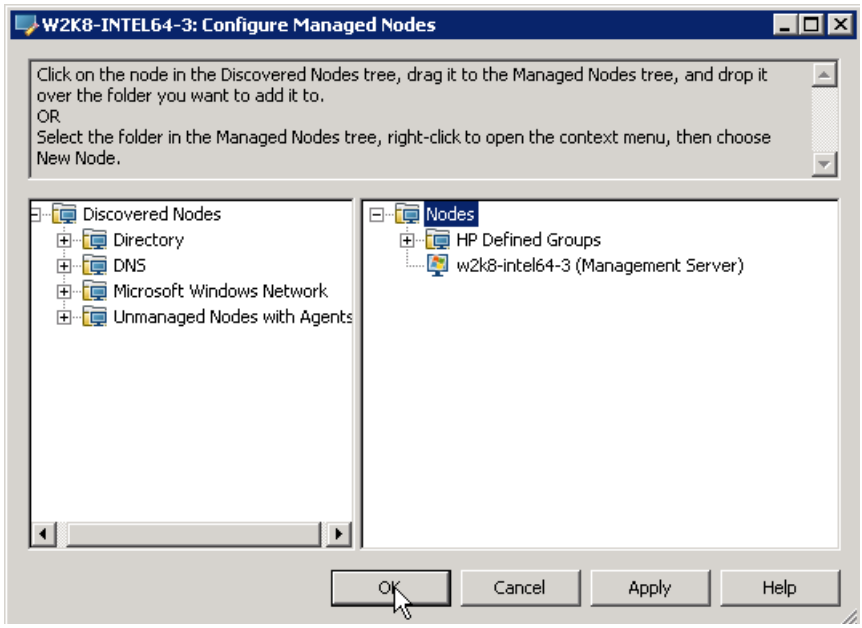
12. In the Configure Managed Nodes window, right-click on MarkLogic and select Delete.



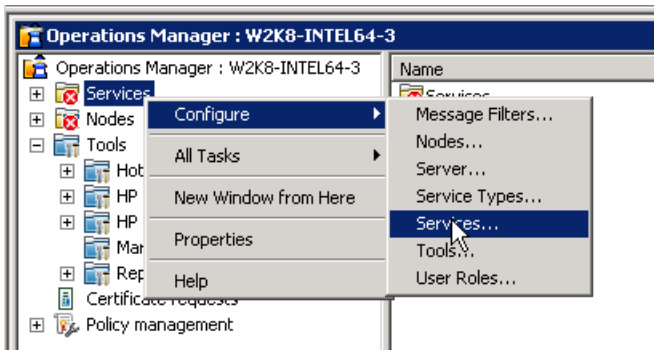
13. In the Confirm Delete window, click Yes.



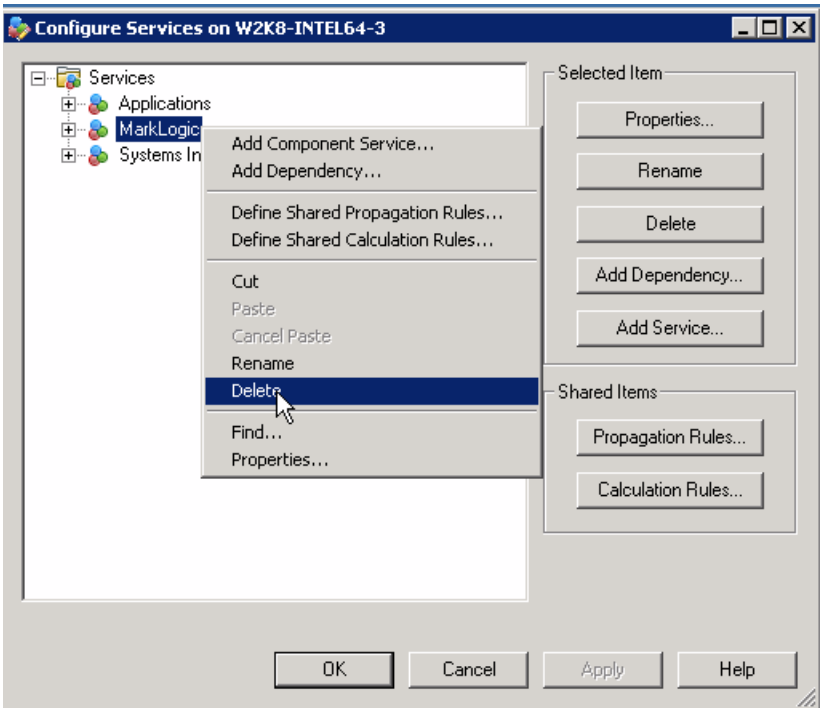
14. The MarkLogic node will be removed from the Configure Managed Nodes window. Click OK.



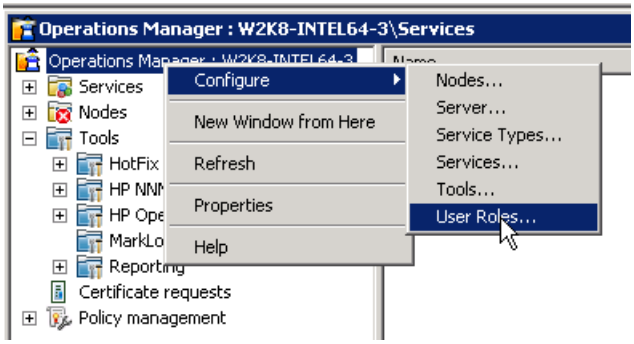
15. In the left-hand menu, right click on Services and select Configure > Services.



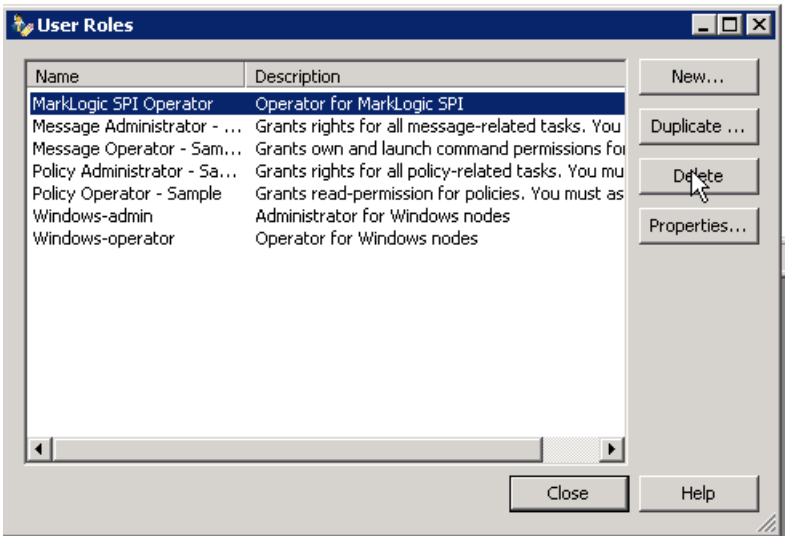
16. In the Configure Services window, right click on MarkLogic and select Delete.



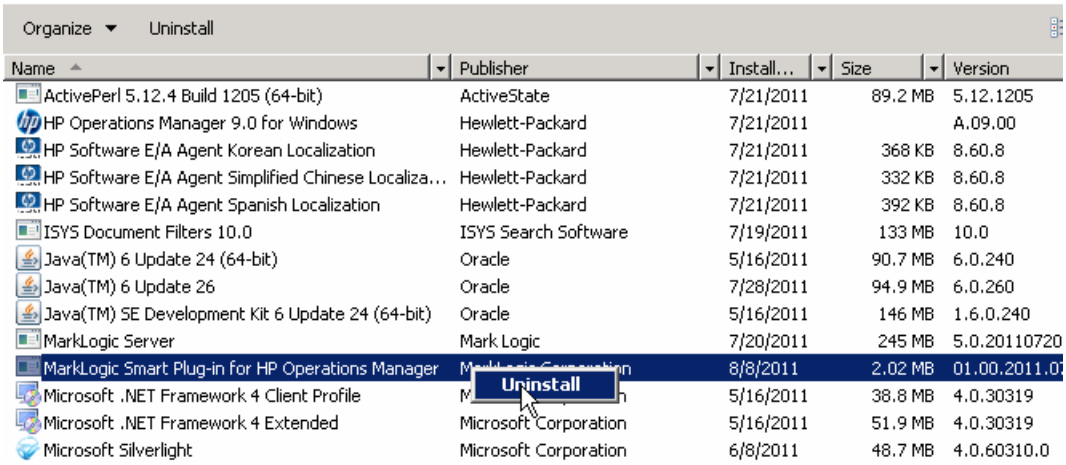
17. In the left-hand menu, right click on Operations Manager *server_name* and select Configure > User Roles.



In The User Roles Window, Select MarkLogic Spi Operator And Click Delete. Click Yes, when prompted. Click Close.



18. Open the Windows Control Panel and select Programs and Features. Locate the MarkLogic Smart Plug-in for HP Operations Manager. Right click and select Uninstall.



4.7 Configuration Files

The following directory contains the configuration files that are specific to the MarkLogic SPI:

```
%OvShareDir%\Instrumentation\Categories\MarkLogicSPI
```

This directory contains the following configuration files:

- mlspi_config.xml
- mlspi_urls.xml

4.7.1 mlspi_config.xml

The `mlspi_config.xml` file identifies the following:

Element	Description
`<name>`	Name of the MarkLogic cluster to be monitored.
`<hostname>`	Name of the monitor host in the MarkLogic Server cluster that sends monitor data to the SPI proxy server.
`<port>`	Port number on the MarkLogic monitor host used by the Management API.
`<ssl>`	Enables/disables SSL communication between the monitor host and the proxy server.
`<username>`	Name of the MarkLogic user with the Admin role.

Element	Description
<password>	Password for the MarkLogic user with the Admin role.
<exclude>	Excludes specific resources from being monitored. The individual resourcetype and name attributes can be used to exclude specific databases, App Servers, and hosts.

By default, all of the MarkLogic cluster resources are monitored. You can also edit the mlspi_config.xml file to exclude specific resources from being monitored. There are two levels of excludes:

- Excluding resources on all clusters. This is done by specifying <exclude> elements in the <global> element.
- Excluding resources on a named cluster. This is done by specifying <exclude> elements in a <cluster> element.

For example, to exclude the Documents database and the Admin App Server from being monitored in all of the clusters, your mlspi_config.xml file would look like the following:

```
<mlspi>
  <global>
    <polling file="mlspi_urls.xml" />
    <excludes>
      <exclude resourcetype="database" name="Documents" />
      <exclude resourcetype="server" name="Admin" />
      <exclude resourcetype="host" name="myother.marklogic.com" />
    </excludes>
  </global>
  <clusters>
    <cluster>
      <name>Master</name>
      <hostname>rh5-intel64-18</hostname>
      <port>8002</port>
      <ssl>0</ssl>
      <username>usr</username>
      <password>pwd</password>
      <excludes>
        <exclude resourcetype="database" name="MyOtherDatabase" />
        <exclude resourcetype="server" name="MyOtherServer" />
        <exclude resourcetype="host" name="myoldhost.marklogic.com" />
      </excludes>
    </cluster>
  </clusters>
</mlspi>
```

4.7.2 mlspi_urls.xml

The `mlspi_urls.xml` file identifies the following:

Element	Description
`<urls>`	Specifies a groups of similar `<url>` elements. A family of URLs can be any of MarkLogic resource type: clusters, databases, servers, and hosts.
`<url>`	Specifies the resource address to be sent by the SPI to the monitor host to obtain monitoring data on the MarkLogic cluster. For details on resource addresses, see “Using the Management API” on page 112. The `interval` attribute specifies how frequently the resource address is sent to the monitor host. You can change the interval for a resource address to either 1 or 60 minutes. If you want to assign a resource address another type of interval, such as 5 minutes, you must create a new task in the Data Collection Policy Group. The `skip` attribute, when set to `true`, will disable the URL so that the resource address is not sent to the monitor host.
`<resources>`	Defines the base URI used by the SPI to retrieve monitoring data.
`<resource>`	Defines a resource family, such as databases, servers, hosts, and so on.

By default, the `mlspi_urls.xml` file is configured to monitor databases, servers, and hosts. You can edit the `mlspi_urls.xml` file and create new policies to monitor additional MarkLogic Server resources.

Note: The `interval` attribute in the `url` element describes the frequency (in minutes) monitoring data is collected on the resource. Each interval must be associated with a matching scheduled task in the Data Collection policy group. If you change or add a new interval value in the `mlspi_urls.xml` file. you must also modify the corresponding Data Collection task or create a new Data Collection task.

The `mlspi_urls.xml` file looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<urls-to-process>
   <urls family="clusters">
      <url interval="60">/manage/v1/databases</url>
      <url interval="60">/manage/v1/servers</url>
      <url interval="60">/manage/v1/hosts</url>
   </urls>
   <urls family="databases">
      <url interval="1">/manage/v1/databases/$dbname/status</url>
      <url interval="60">/manage/v1/databases/$dbname/counts</url>
      </urls>
   <urls skip="false" family="servers">
      <url interval="1">
         /manage/v1/requests/?server-id=$servername&amp;seconds-min=60
      </url>
      <url interval="1">
         /manage/v1/servers/$servername&amp;view=status</url>
   </urls>
   <urls family="hosts">
      <url skip="false" interval="1">
         /manage/v1/hosts/$hostname?view=status</url>
      <url interval="1">
         /manage/v1/forests?host-id=$hostname&amp;view=status</url>
   </urls>
   <resources baseURI="/manage/v1/">
      <resource pattern="$dbname" resourcetype="databases">
         uriref
      </resource>
      <resource pattern="$servername" resourcetype="servers">
         uriref
      </resource>
      <resource pattern="$hostname" resourcetype="hosts">
         uriref
      </resource>
   </resources>
</urls-to-process>
```

4.8 MarkLogic Default Policies

All MarkLogicSPI Monitor Threshold policies are asynchronous (Type = External) which means the data collection module (DCM) collects and feeds the HP Operations Agent with collected values.

To view the MarkLogic monitoring messages, in the left hand-menu, select Services > MarkLogic.

The following MarkLogic components are monitored by the default policies:

- MarkLogic Cluster
- MarkLogic Database
- MarkLogic Application Server
- MarkLogic Host

4.8.1 MarkLogic Cluster

The MarkLogic cluster policies are:

- DBCount - Count databases belonging to the cluster. If the number of databases belonging to the MarkLogic cluster changes, a warning message will be generated.

  All messages must be acknowledged by MarkLogicSPI operators.

- HostCount - Count hosts belonging to the cluster. If the number of hosts belonging to the MarkLogic cluster changes, a warning message will be generated.

  All messages must be acknowledged by MarkLogicSPI operators.

- SrvCount - Count Application Servers belong to the cluster. If the number of application servers belonging to the MarkLogic cluster changes, a warning message will be generated.

  All messages must be acknowledged by MarkLogicSPI operators.

4.8.2 MarkLogic Database

The MarkLogic database policies are:

- AsyncReplicating - The number of forests in asynchronous replicating mode. If there is at least one forest in the asynchronous replicating mode the critical message will be sent to the message browser.

  Thresholds:

  - NORMAL: AsyncReplication = 0
  - CRITICAL: AsyncReplicating >= 1

  All related critical messages are acknowledged when the state for that specific database changes and the value drops below the threshold.

- BackupCount - The number of database backups in progress. If there is at least one active backup job for that specific database a warning message will be sent to the message browser.

  Thresholds:

  - NORMAL: BackupCount = 0
  - WARNING: BackupCount >= 1

  All related warning messages will be acknowledged when there are no active backup jobs for that specific database.

- FailedMasters - The number of database failover events. If there is at least one failover event for that specific database a critical message will be sent to the message browser.

  Thresholds:

  - NORMAL: FailedMasters = 0
  - CRITICAL: FailedMasters >= 1

  All related critical messages will be acknowledged when there are no failover events for that specific database.

- State - The database availability. If the database is unavailable a critical message will be sent to the message browser.

  Thresholds:

  - NORMAL: State = 0
  - CRITICAL: State >= 1

  All related critical messages will be acknowledged when all databases are available again.

4.8.3 MarkLogic Application Server

The MarkLogic App Server policies are:

- TLRqss - The number of long-running requests. If the number of long running queries is too high, the monitor will send the warning message to the message browser.

  Thresholds:

  - NORMAL: TTLRqss = 0
  - WARNING:TTLRqss >=1
  - CRITICAL: TTLRqss >=10

  If the critical threshold is exceeded the warning message will be replaced by critical message and vice versa.

  If there are no long running requests on that specific application server all related warning and critical messages will be acknowledged.

4.8.4 MarkLogic Host

The MarkLogic host policies are:

- MaxStandsPerForest - The number of stands per forest on one host. If the number of stands per forest on one host is too high, the monitor will send the warning message to the message browser.

  Thresholds:

  - NORMAL: MaxStandsPerForest = 0
  - WARNING: MaxStandsPerForest >= 5
  - CRITICAL: MaxStandsPerForest >= 50

  If the critical threshold is exceeded the warning message will be replaced by critical message and vice versa. If there are no stands on that specific host server all related warning and critical messages will be acknowledged.

- MinCapacity - The forest storage capacity. If the storage capacity fails below the threshold, the monitor will send the warning message to the message browser.

  Thresholds:

  - NORMAL: MinCapacity > 15
  - WARNING: MinCapacity <= 15
  - NORMAL: MinCapacity <= 10

  If the critical threshold is exceeded the warning message will be replaced by critical message and vice versa. If there is enough free space on that specific host all related warning and critical messages will be acknowledged.

- StateNotOpen - The number of not-opened forests. If there is at least one unopened forest a critical message is sent to message browser.

  Thresholds:

  - CRITICAL: StateNotOpen = 1
  - NORMAL: StateNotOpen = 0

  All related critical messages will be acknowledged when all forests on that specific host are available.

4.9 Modifying Policies and Creating New Policies

This section describes how to modify and create new policies to monitor a MarkLogic cluster.

> **Note:** The procedures in this section assume you have administer-level experience with HP Operations Manager.

The topics in this section are:

- Policy Rules
- Changing a Policy
- Creating a New Policy

4.9.1 Policy Rules

The logic for a policy is encompassed in its rule. There are several types of policy rules. An example of the simplest type of rule is that used by the StateNotOpen policy in the MarkLogic Health Monitor group. This rule detects if there is at least one unopened forest. If so, a Critical message is sent to message browser.

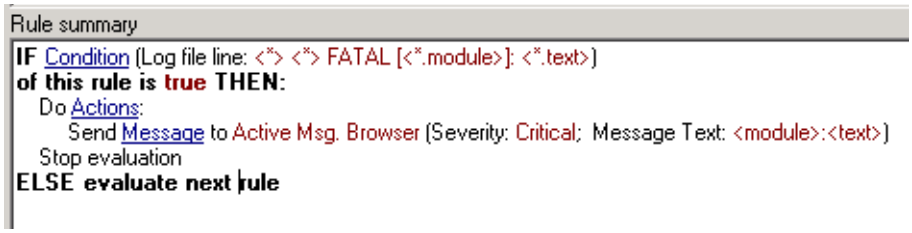


The MaxStandsPerForest policy in the MarkLogic Health Monitor group has a rule that reports a Critical message if the number of stands for a forest is 50 or more; a Warning message if the number of stands is 35 or more, and no message if the number of stands is between 0 and 34.

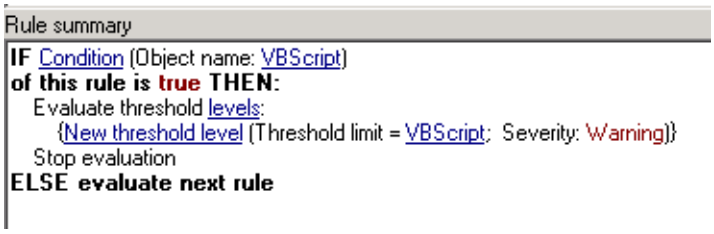


The MarkLogic SPI Logfile Monitor policy in the Internal Messages group has a rule that monitors the condition of the SPI and generates a message if it detects a fatal error.

Note: The SPI Logfile does not rollover, so you must periodically delete it to preserve disk space.



Some policy rules have a script that specifies the threshold. You use a script to determine the threshold policy if the metric is something other than a number or a Boolean value or if you want to evaluate multiple sources. A script allows you to determine if the threshold has been crossed based on your own criteria. For example, the `DBCount` policy calls a VBScript rule that keeps track of the number of databases in the cluster. If the number of databases belonging to the MarkLogic cluster changes, a warning message messages is generated.



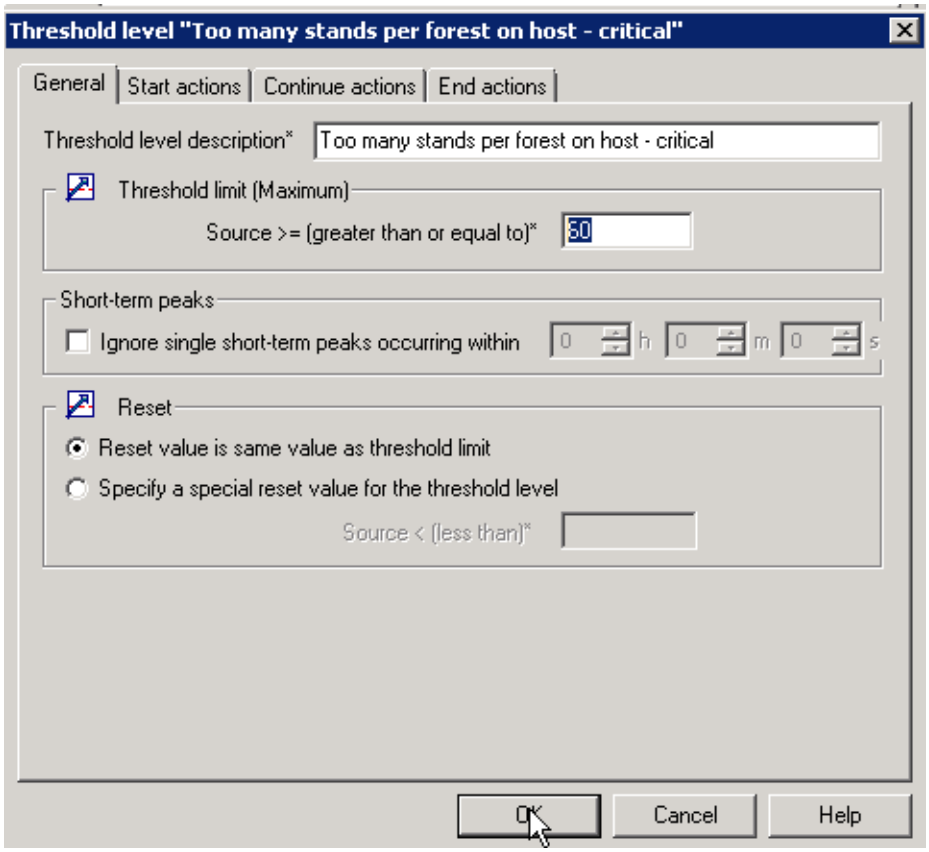
4.9.2 Changing a Policy

Changing a policy typically means changing all or parts of the policy rule. For example, the following procedure shows have to change the MaxStandsPerForest policy in the MarkLogic Health Monitor group to generate a Critical message is the number of stands per forest is over 60, rather than 50.

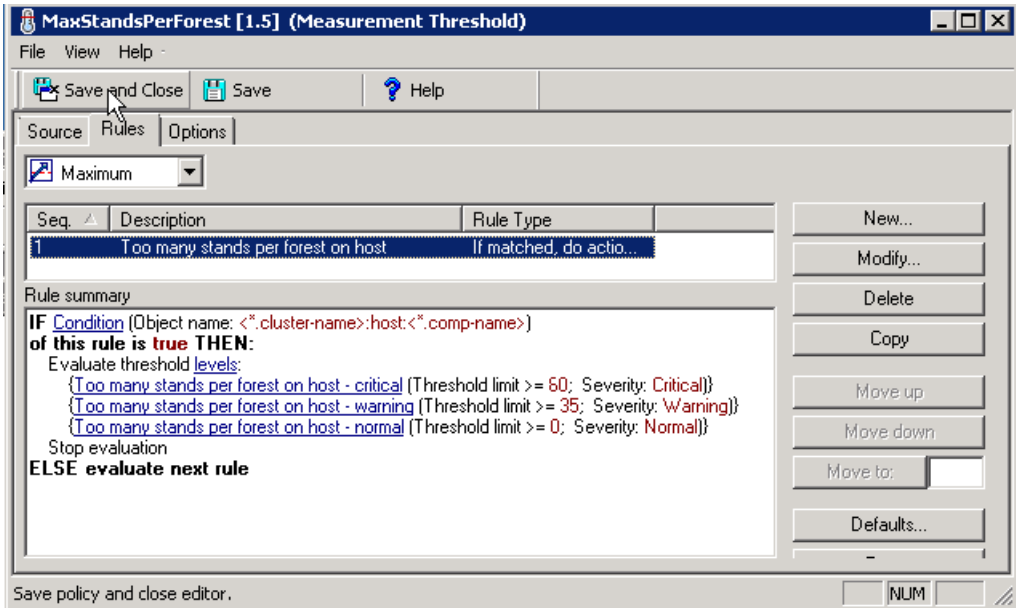
1. In the MaxStandsPerForest policy Rule Summary, click on the `Too many stands per forest on host - critical` action.



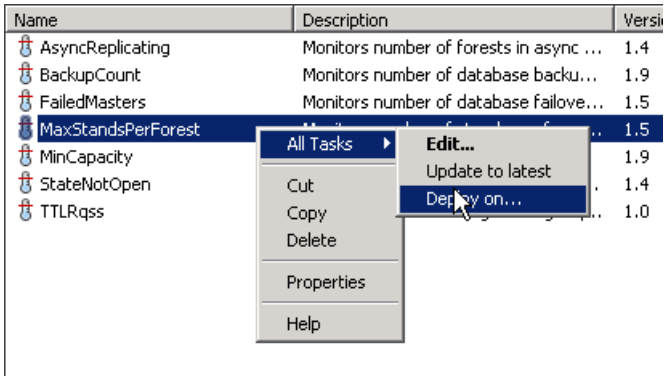
2. Change the Threshold limit (Maximum) value to 60. Click OK.



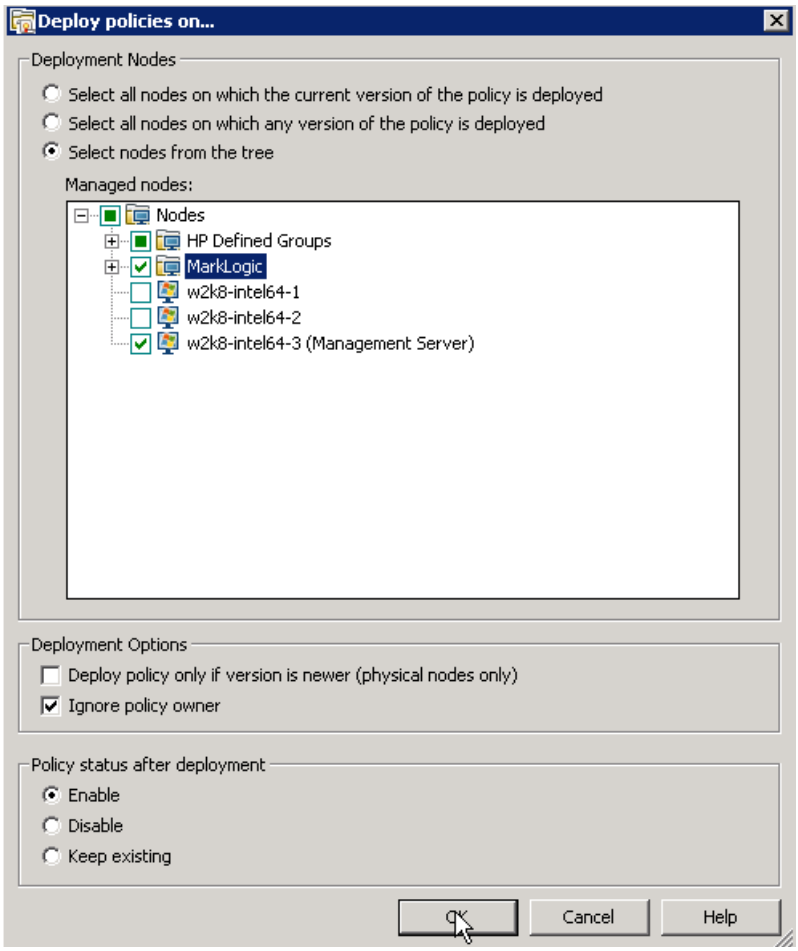
3. The Rule Summary for the policy will appear as follows. Click Save and Close.



4. To deploy the modified policy, right click on the policy and select All Tasks > Deploy on.



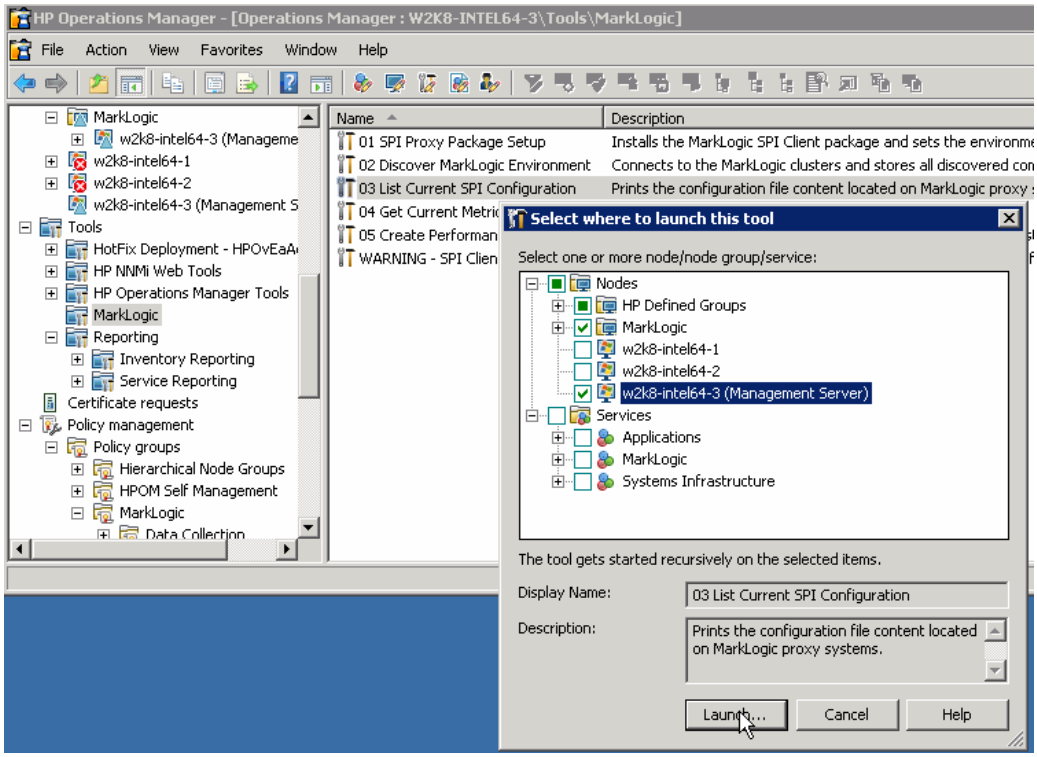
5. In the Deploy Policies On window, select MarkLogic. Under Deployment Options, check Ignore Policy Owner (no other options should be checked). Under Policy Status After Deployment, select Enable. Click OK.



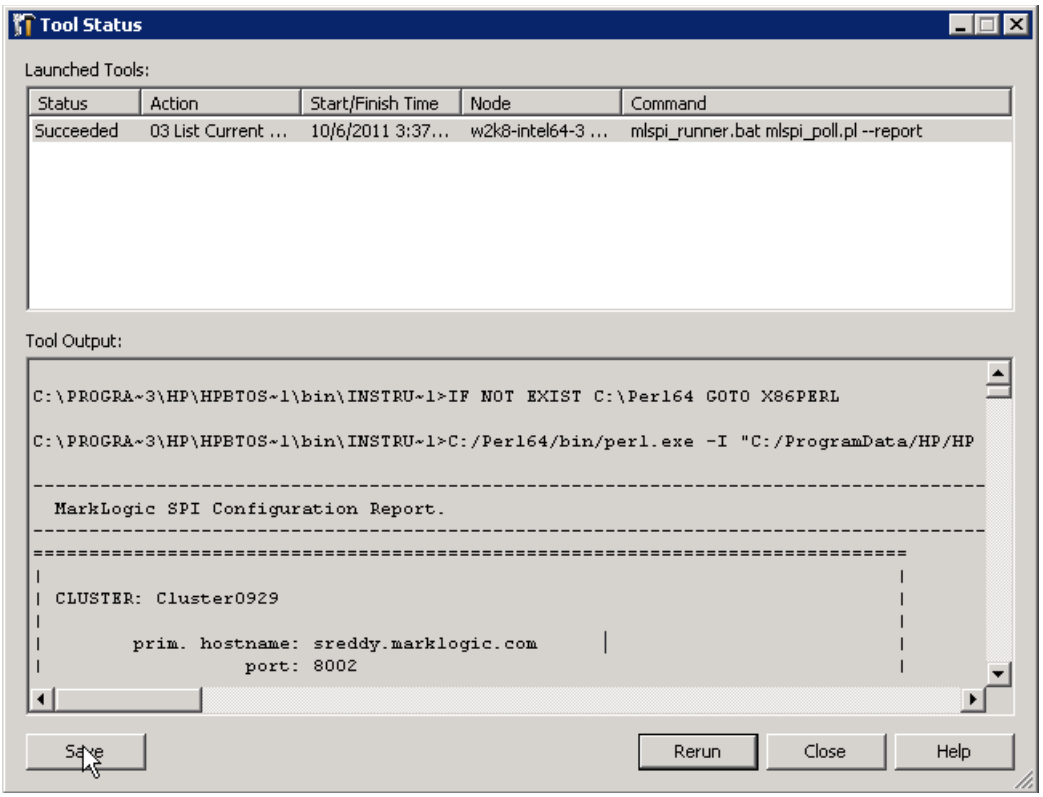
4.9.3 Creating a New Policy

This section describes how to create a new policy that reports when Flexible Replication is enabled on a database in the cluster.

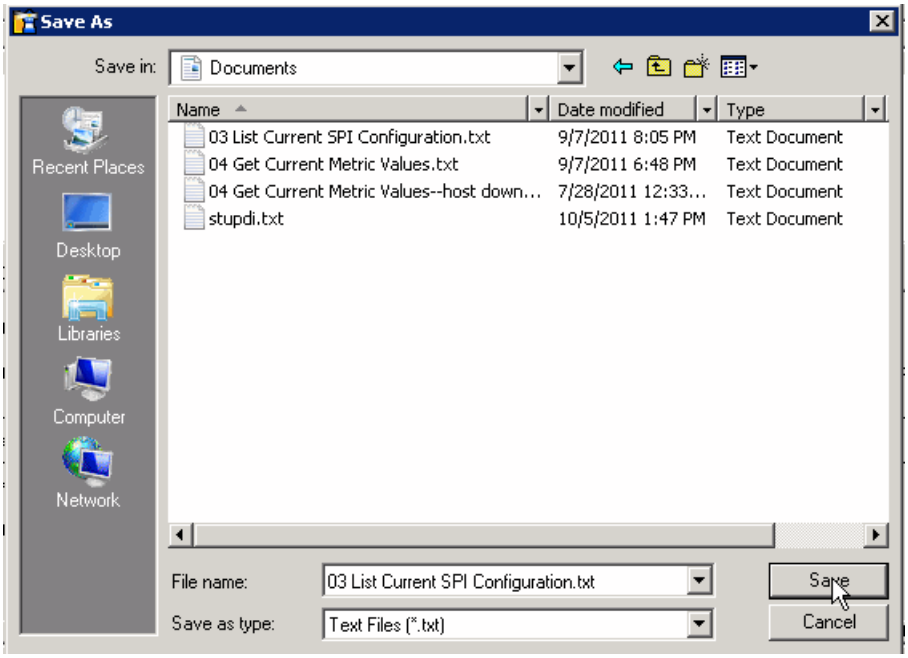
1. Before you create a new policy, obtain the list of metric values for monitoring the MarkLogic cluster. Navigate to Tools in the left-hand menu, locate MarkLogic. Double click on MarkLogic. In the right-hand window, click on 03 List Current SPI Configuration. In the dialog window, under Nodes, check the box next to the proxy server(s). Click Launch.



2. If the tool completes successfully, you will see the Succeeded status in the Tool Status window. Click Save to save the output to a file.



3. In the Save As window, save the output file.



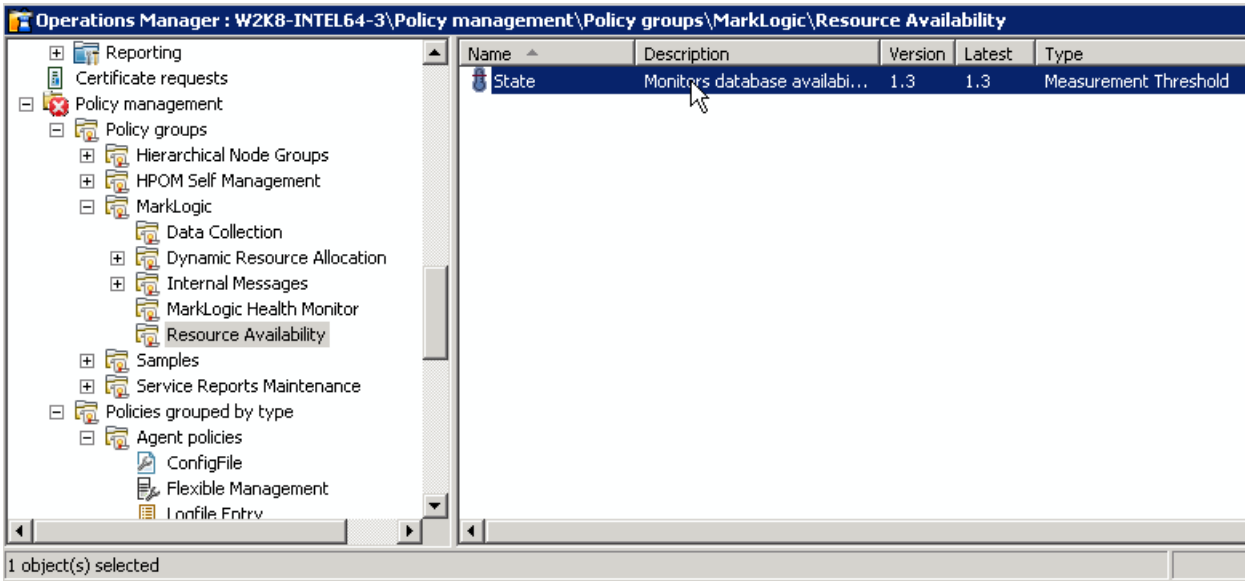
4. Locate the `[flexible-replication-enabled]` metric in the output file. You will need the `norm.name`, `FLEXReplEnabled`, to create your new policy. The `norm.name` is used by the MarkLogic SPI to store the metric value in the HP Operations Manager database.

```
[flexible-replication-enabled]
                norm. name : FLEXReplEnabled
                perf. name : FLEXREPLENABLED
              metric type : string (TEXT)
                comp. type : database
```

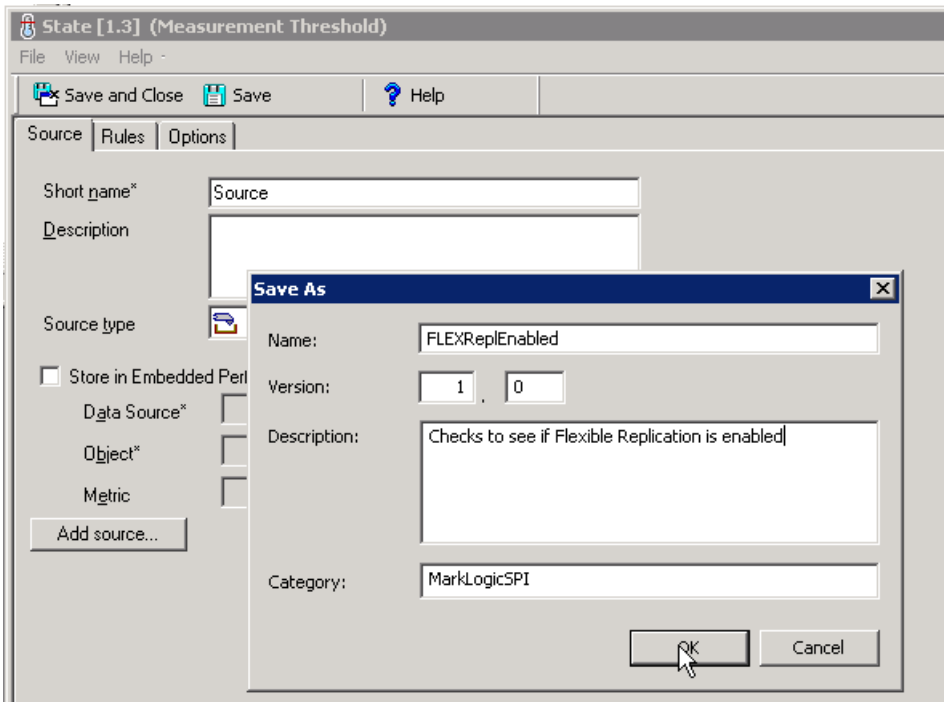
5. The recommended way to create a new policy is to copy an existing policy that is similar to the policy you want to create. In the case of Flexible Replication Enabled policy, the existing database State policy is the closest match because it simply detects whether or not the database is enabled.

```
[state]
                norm. name : State
                perf. name : STATE
              metric type : string (TEXT)
                comp. type : database
```

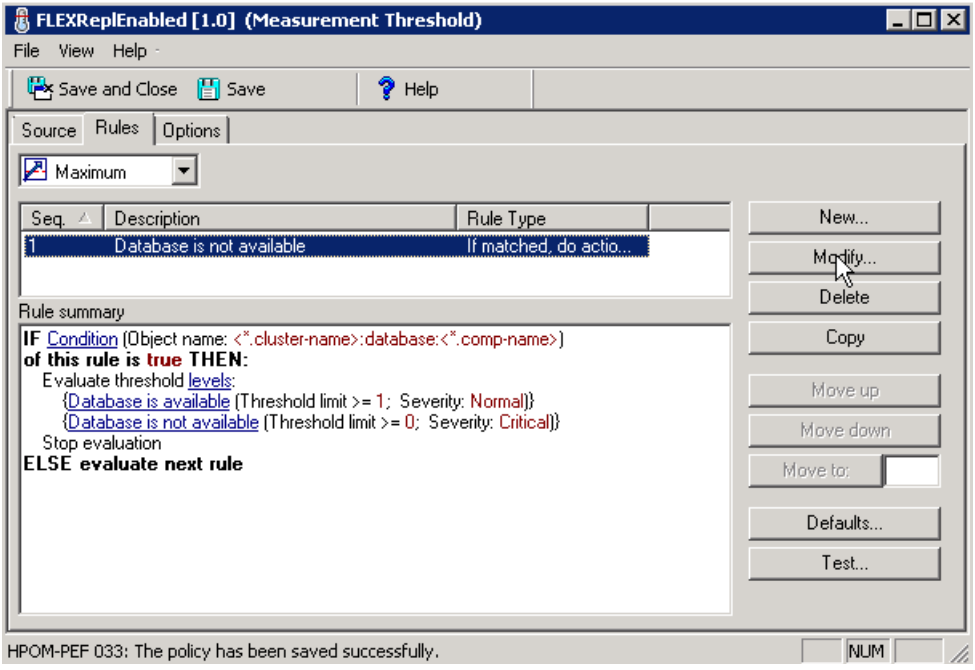
In the left-hand menu, navigate to Policy Management > Policy Groups > MarkLogic and select the Resource Availability policy group. In the right-hand window click on the State policy.



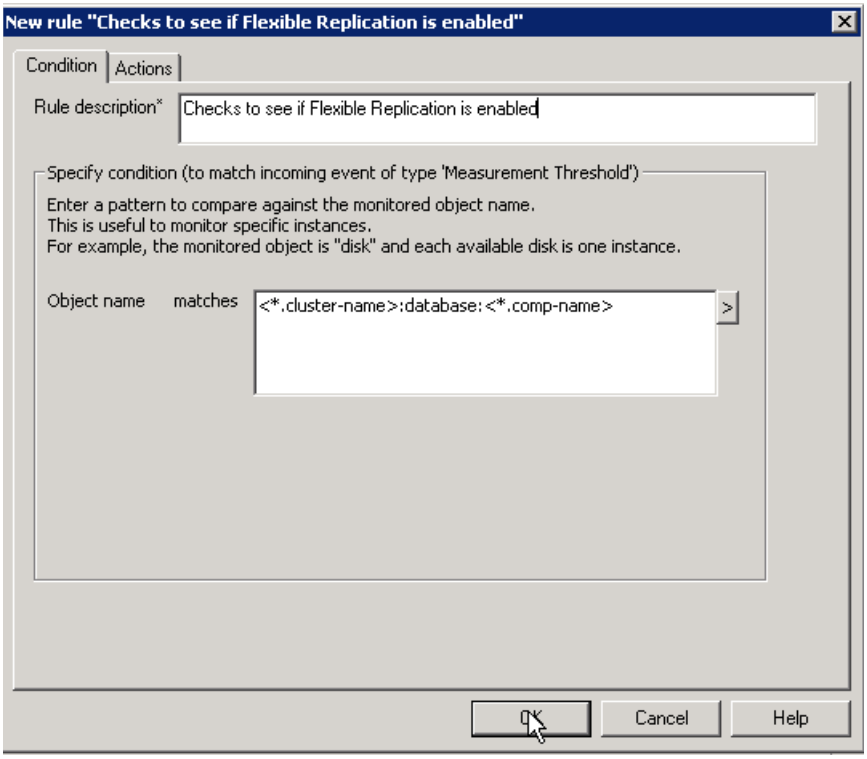
6. To create a copy of the State policy, select File > Save As. In the Save As window, enter the `norm.name` of the metric, `FLEXReplEnabled`, just as it appears in the output file from the 04 Get Current Metric Values tool. Change the text in the Description field and click OK.



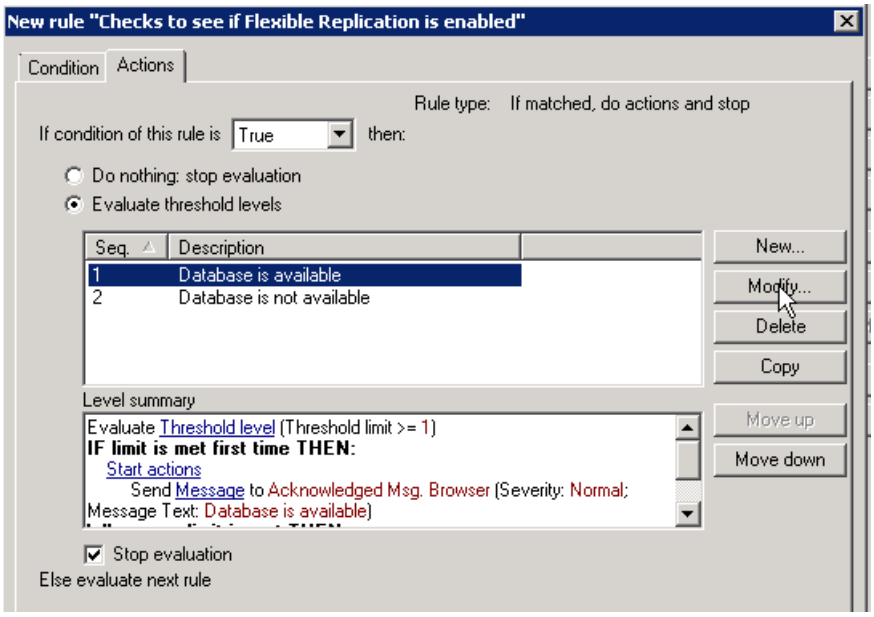
7. Select the Rules tab in the new policy, select the text in the Description field, and click Modify. This opens a New Rule window.



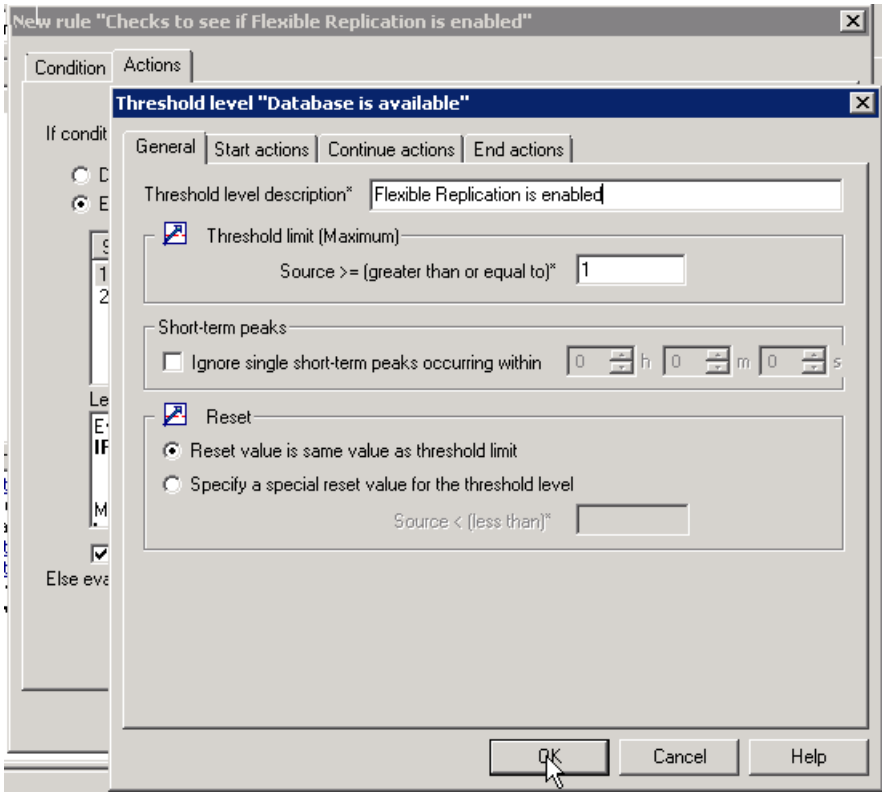
8. In the New Rule window, enter a new description for the policy in Rule Description field. Click OK.



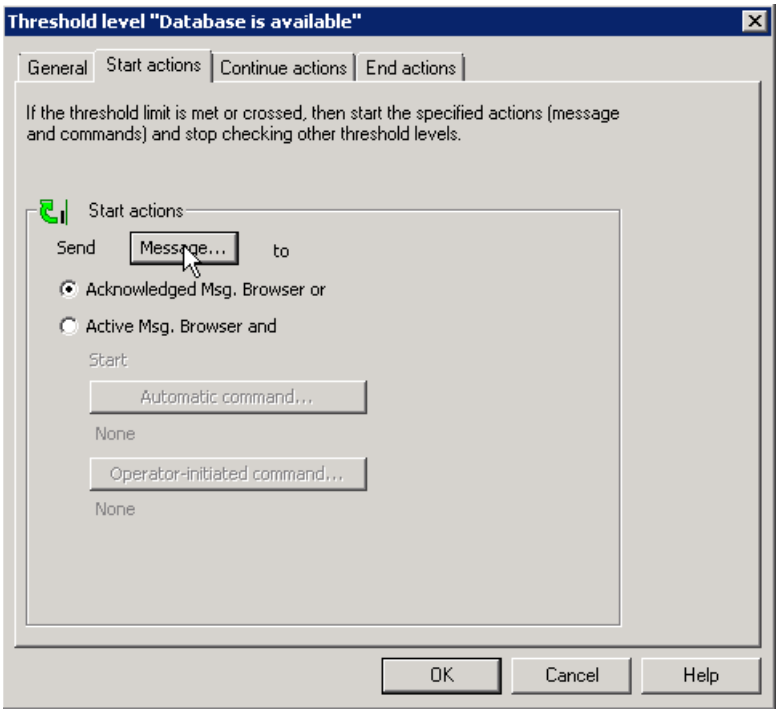
9. Click the Actions tab, and modify the text for each action.



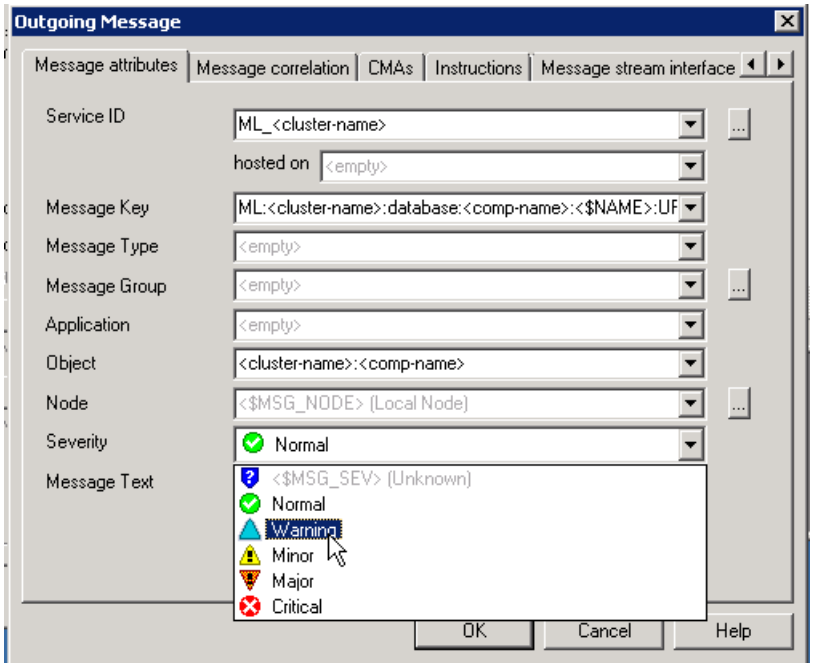
For example, replace the “Database is available” Threshold Level Description with “Flexible Replication is enabled.”



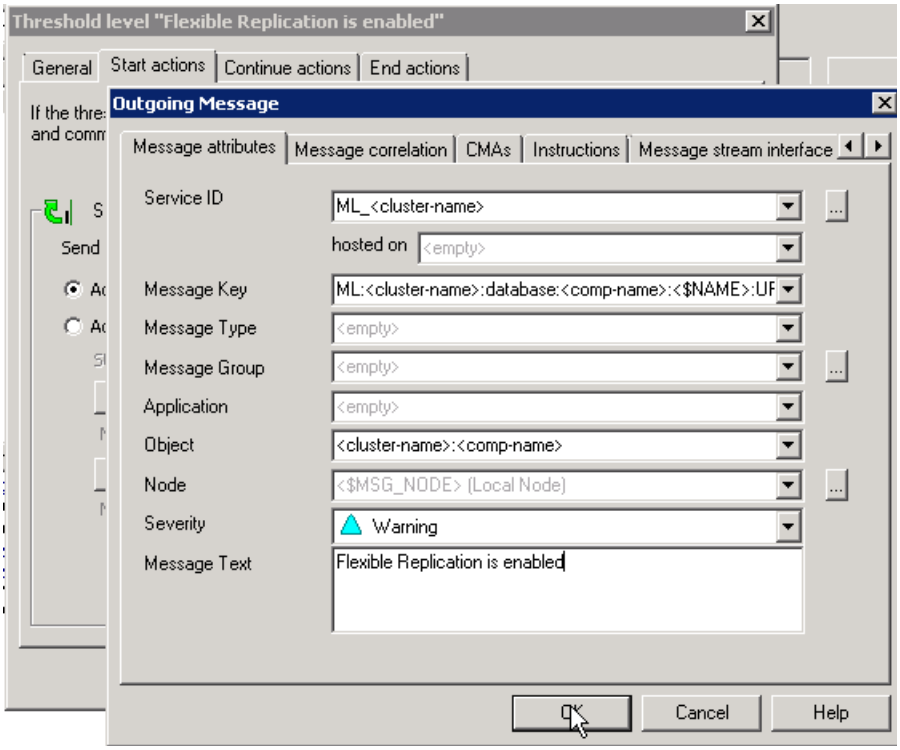
10. To change the contents of the message, select the Start Actions tab and click Messages.



Change the Severity level to “Warning.”

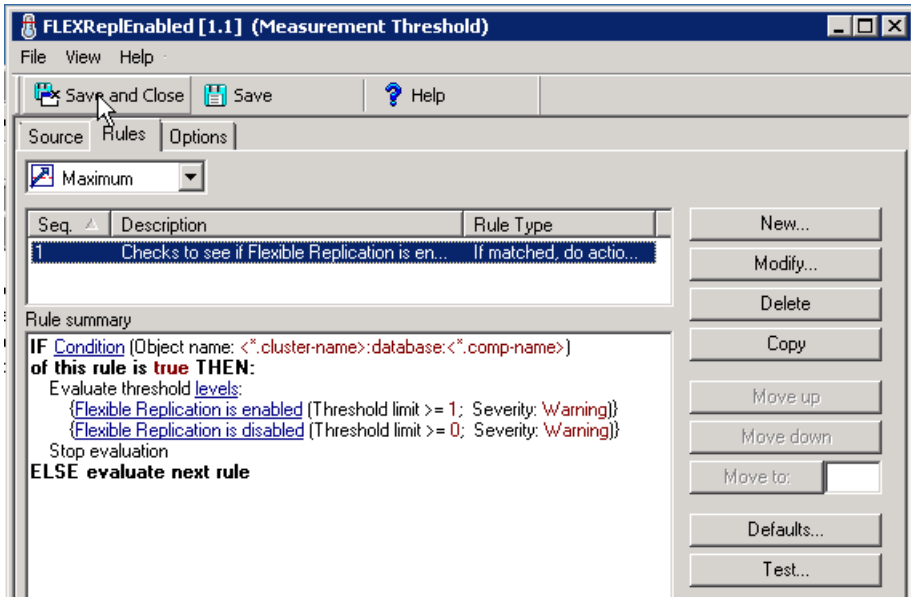


Change the Message Text. Click OK.

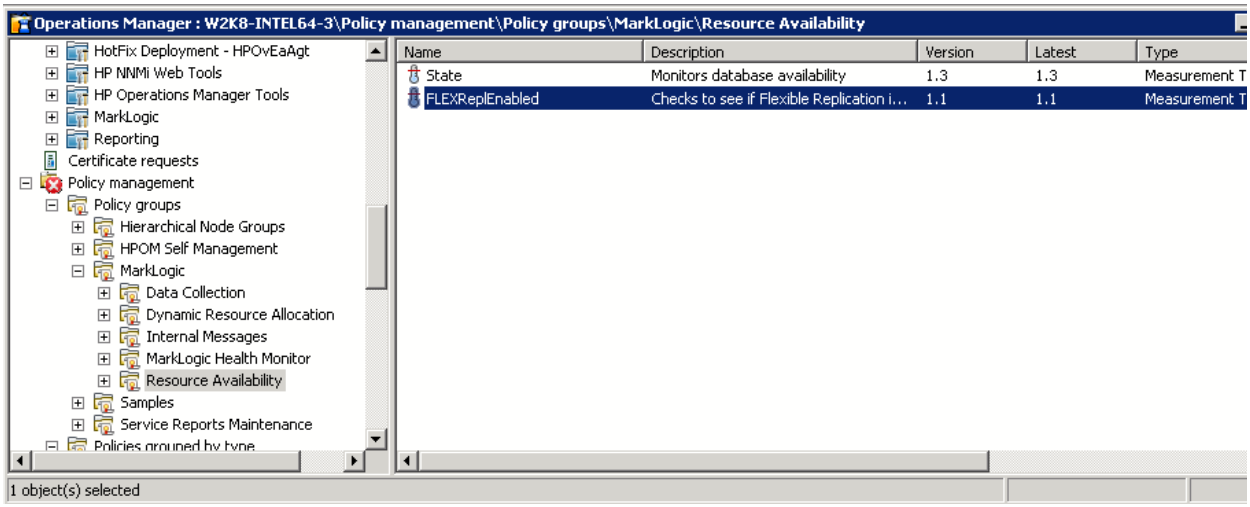


11. Repeat Steps 9 and 10 and replace the "Database is not available" Threshold Level Description and Message Text with "Flexible Replication is disabled." Change the Severity level to "Warning."

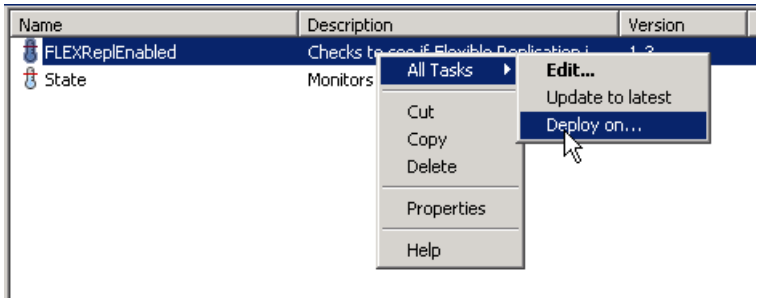
The final policy should look like the following:



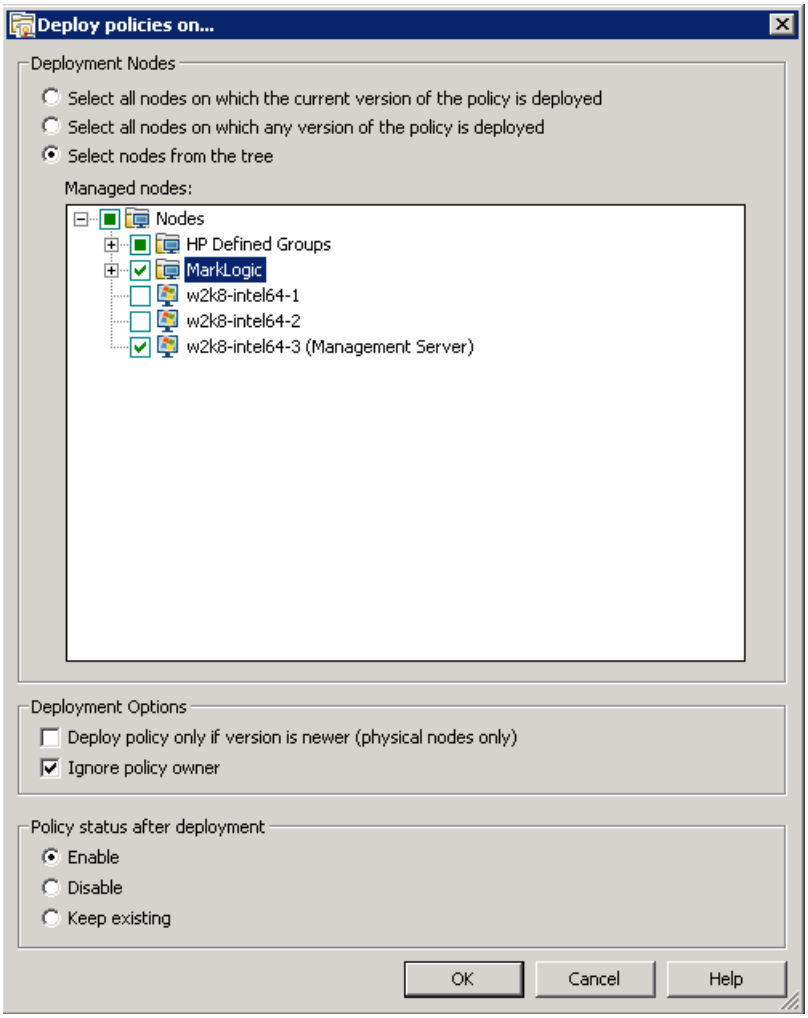
12. The new policy will appear in the Resource Availability Policy Group.



13. To deploy the new policy, right click on the policy and select All Tasks > Deploy on.



14. In the Deploy Policies On window, select MarkLogic. Under Deployment Options, check Ignore Policy Owner (no other options should be checked). Under Policy Status After Deployment, select Enable. Click OK.



4.10 Reconfiguring HP Operations Manager After Changes to MarkLogic Server

This section describes the procedures for updating HP Operations Manager after changes are made to MarkLogic Server.

If you change the username or password for the MarkLogic user account used by HP Operations Manager or add a new monitoring host, you need to:

1. Update the username and password in the mlspi_config.xml file, as described in “Edit the Configuration File” on page 64

2. Deploy the SPI, as described in “Deploy MarkLogic Smart Plug-in Instrumentation” on page 69

3. Run the 02 Discover MarkLogic Environment tool described in “Launch the MarkLogic SPI Setup Tools” on page 71

If you do any of the following, then you need to run the 02 Discover MarkLogic Environment tool described in “Launch the MarkLogic SPI Setup Tools” on page 71:

- Add or delete resource, such as a database, App Server, or host.
- Enable a feature, such as Database Replication.

4.11 Performance Data Source

Graphing instructions exceed the scope of this chapter. If you are interesting in graphing and you need more details then you should consult the HP's official documentation for your version of HP Operations Manager.

MarkLogicSPI creates the performance data stores, which can be used as a data source for other HP products. All MarkLogic metrics can be graphed in HP Performance Manager.

This section provides you the explanation how can you map MarkLogic metric names and HP performance agent objects and metric names.

Below is the fragment of a MarkLogic SPI Configuration Report generated by the 03 List Current SPI Configuration tool described in “Launch the MarkLogic SPI Setup Tools” on page 71. This report that maps MarkLogic metric names to the HP Operations Manager object names required for graphing in HP Performance Manager.

```
-------------------------------------------------------------------------
     Servers
-------------------------------------------------------------------------

    [Manage]
                object: uniqClusterName:server:Manage
          perf. object: UNIQCLUSTERNAME_SERVER_MANAGE
                   uri: /manage/v1/servers/Manage?group-id=Default

    [TaskServer]
                object: uniqClusterName:server:TaskServer
          perf. object: UNIQCLUSTERNAME_SERVER_TASKSERVER
                   uri: /manage/v1/servers/TaskServer?group-id=Default

           Metrics
          ------------------------------------------

             [concurrent-request-limit]
                  norm. name : CncrrntRqsLimit
                  perf. name : CNCRRNTRQSLIMIT
                 metric type : numeric (INT32)
                  comp. type : server
```

5.0 Using the Management API

The Management API is a REST-based API that allows you to access MarkLogic Server instrumentation with no provisioning or set-up. The API provides the ability to easily capture detailed information on MarkLogic Server objects and processes, such as hosts, databases, forests, App Servers, groups, transactions, and requests from a wide variety of tools. The Monitoring Dashboard, Nagios plugin, and HP Operations Manager Smart Plug-in (SPI) described in this guide are all implemented on top of the Management API.

This chapter describes how to use the Management API to obtain monitoring data from MarkLogic Server. This chapter includes the following sections:

- Terms used in this Chapter
- Overview of the Management API
- Security
- Resource Addresses
- Obtaining the Options Node for a Resource Address
- Specifying the Management API Version
- Specifying Parameters in a Resource Address
- Interpreting the Output

5.1 Terms used in this Chapter

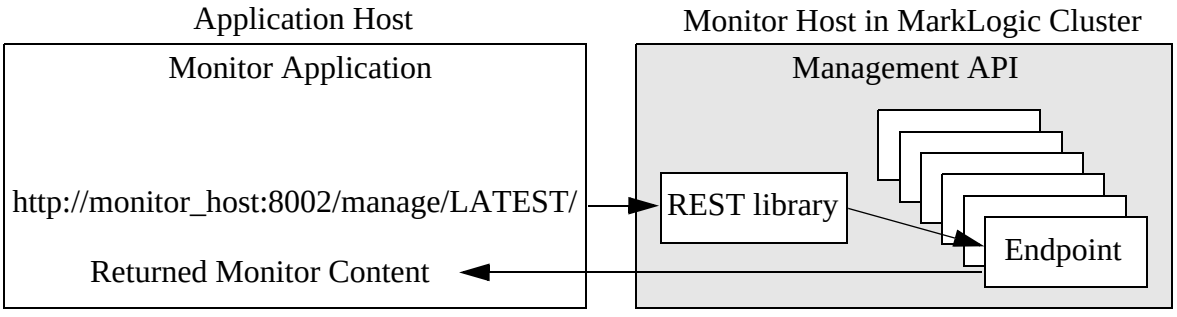
- An *Object* is a component of interest in MarkLogic Server, such as a cluster, host, App Server, or database.
- A *Process* is a request or transaction in MarkLogic Server.
- *Monitor Content* is the XML, HTML, or JSON structure that represents the data returned by the Management API.
- A *Monitor Application* can be any application that requests and makes use of monitoring data, such as a Web browser, a plugin for an existing monitoring tool, or the Monitoring Dashboad described in “Using the MarkLogic Server Monitoring Dashboard” on page 16.
- The *Monitor Host* is the host in the MarkLogic Server cluster that uses the Monitor API to respond to requests for monitoring content from the monitor application.
- The *Manage App Server* is the App Server on the Monitor Host that is configured to handle monitor requests. The Manage App Server is bound to port 8002 and is the App Server used by the Nagios and HP Operations Manager plugins and the Monitoring Dashboard.
- *REST* stands for *Representational State Transfer*, which is an architecture style that, in the context of monitoring MarkLogic Server, describes the use of HTTP to make calls between a monitoring application and monitor host.

- A *Resource* is an abstraction of a MarkLogic Server object, as presented by the REST architecture.
- A *Resource Address* is a URL that identifies a MarkLogic Server resource. The resource addresses are described in "Resource Addresses" on page 114.
- A *View* is the returned monitoring information about a resource. You can have different views of the same resource. A view can be for a single resource (known as an *item view*) or a number of resources (known as a *list view*).
- A *Representation* is a view of a resource in a particular format, such as XML, HTML, or JSON.
- A *Parameter* is an addition to the end of a resource address to filter and/or format the view returned from MarkLogic Server. Parameters are expressed as query strings in the URL and are described in "Specifying Parameters in a Resource Address" on page 126.
- An *Endpoint* is an XQuery module on MarkLogic Server that is invoked by and responds to an HTTP request for monitoring information.
- A *Plugin* is an XQuery module that provides extension capabilities using the Plugin framework described in the System Plugin Framework chapter in the *Application Developer's Guide*.

5.2 Overview of the Management API

The Management API is implemented on top of the REST Library described in Using the REST Library to Develop RESTful Web Services in the *Application Developer's Guide*. Requests to monitor an object in MarkLogic Server are made by means of a resource address that returns a view containing the monitor data for the object. The view can be returned in various formats, such as XML, HTML, or JSON.

Every resource address in the Management API invokes a monitoring endpoint, which is an XQuery module on the target MarkLogic Server host. The monitoring endpoints are invoked by a resource address in an application, such as a browser. The Management API uses the REST library to validate the request, authorize the user, and rewrite the resource address to one understood by the monitoring framework before invoking the endpoint module. The endpoint module returns the monitoring data for the resource to the application.



5.3 Security

As described in "Monitoring Tools and Security" on page 6, client access to a Management API and endpoints requires that they authenticate as a user with the `manage-user` role. If custom Plugin code requires additional privileges, you can create and assign a custom role to users of the Management API to enable that functionality.

If you have enabled SSL on the `Manage` App Server, your resource address must start with HTTPS, rather than HTTP, and you must have a MarkLogic certificate authority on your browser, as described in Accessing an SSL-Enabled Server from a Browser or WebDAV Client in the *Administrator's Guide*.

5.4 Resource Addresses

This section provides an overview of the structure and capabilities of the resource addresses provided by the Management API. For details on each resource address, see the *XQuery and XSLT Reference Guide*.

Resource addresses fall into the categories listed in the table below. The output from each type of resource address is described in "Interpreting the Output" on page 130.

Type of Resource Address	Returns
List	A list of resources. For example, as list of the forests in the cluster: `http://localhost:8002/manage/LATEST/forests`
Item	A specific resource. For example, a specific forest in the cluster: `http://localhost:8002/manage/LATEST/forests/Documents`

A resource address takes the form of a URL that includes a host name and a port number. The most basic resource address returns summary information for the entire cluster:

```
http://host:port/manage/LATEST/
```

The Management API version, `LATEST` in this release, is specified in every resource address to maintain compatibility with future revisions of the Management API.

You can optionally include the name of a resource and parameters in a resource address as follows:

```
http://host:port/manage/version/resource?param=value&param=value
```

The following table summarizes all of the resource addresses in the Management API. The Resource Address column shows the resource address. Some resource addresses have required parameters, which are included in the resource address. The Optional Parameter column lists the optional parameters accepted by the resource address. The Returns column describes what is returned by the resource address.

All resource addresses support the `format` parameter, which is described in "Formatting the Monitor Results" on page 127. Some resource addresses include a *custom* parameter that make use of plugins to return customized results. For details on creating plugins for the Management API, see "Extending Management API with Plugins" on page 131.

All of the URLs listed in the Resource Address column are proceeded by:

```
http://hostname:port/manage/LATEST
```

Resource Address	Optional Parameters	Returns
/	format={xml |json |html} view={default |schema |status |query |connect}	Local cluster summary and links to: clusters?cluster-role=foreign databases forests groups hosts requests servers transactions status query

Resource Address	Optional Parameters	Returns
/status	format={xml |json |html}	Local cluster aggregate status and links to: /hosts?view=status /servers?view=status /forests?view=status /requests /transactions /
/query	format={xml |json |html} name-match=*resource* page-start= *num_windows* page-length= *num_resources*	Local cluster query summary of all resources in the cluster and links to: /clusters/*all_foreign_clusters* /databases/*all_databases* /forests/*all_forests* /groups/*all_groups* /hosts/*all_hosts* /servers/*all_servers*?group-id=*group* /
/clusters	format={xml |json |html} view={default |schema} cluster-role={all |local |foreign}	List of clusters coupled to the local cluster and links to: *all_clusters* clusters?view=schema

Resource Address	Optional Parameters	Returns
/clusters/{*id*|*name*} (where *id*|*name* describes a foreign cluster)	format={xml |json |html} view={default |config |schema |status |query}	Foreign cluster summary and links to: databases forests clusters?cluster-role=foreign clusters/*cluster*/config clusters/*cluster*/status
/clusters/{*id*|*name*}/config (where *id*|*name* describes a foreign cluster)	format={xml |json |html}	Foreign cluster configuration and links to: clusters/*cluster*
/clusters/{*id*|*name*}/status (where *id*|*name* describes a foreign cluster)	format={xml |json |html}	Foreign cluster status and links to: clusters/*cluster*
/forests	view={default |storage |schema |status} format={xml |json |html} database-id={*id*|*name*} group-id={*id*|*name*} host-id={*id*|*name*} fullrefs={true|false}	Summary of forests in the cluster and links to: /forests/*all_forests* / /forests?view=status /forests?view=storage

Resource Address	Optional Parameters	Returns
/forests/{*id*|*name*}	view={default |storage |config |counts |edit |status} format={xml |json |html}	Forest summary and links to: /databases/*database* (if attached) /hosts/*host* /forests /forests/*forest*/config /forests/*forest*/counts /forests/*forest*/edit /forests/*forest*/status
/forests/{*id*|*name*}/status	format={xml |json |html}	Forest status and links to: /databases/*database* (if attached) /hosts/*host* /forests/*forest*
/forests/{*id*|*name*}/counts	format={xml |json |html} *property=*property*	Forest counts and link to: /forests/*forest*
/forests/{*id*|*name*}/config	format={xml |json |html}	Forest configuration and links to: /forests/*forest*
/forests/{*id*|*name*}/edit		Links to the forest configuration page in the Admin Interface.
/forests/{*id*|*name*}/{*custom*}	format={xml |json |html} {*custom*}	Provided by Plugin. For details, see “Extending Management API with Plugins” on page 131.

Resource Address	Optional Parameters	Returns
/databases	view={default|schema} format={xml|json|html}	Summary of databases in the cluster and links to: /databases/*all_databases* /
/databases/{*id*|*name*}	view={default|config|counts|edit|status} format={xml|json|html}	Database summary and links to. /databases/*Schemas* /databases/*Security* /databases/*Triggers* /forests/*all_forests* /databases /databases/*database*/config /databases/*database*/counts /databases/*database*/edit /databases/*database*/status
/databases/{*id*|*name*}/status	format={xml|json|html}	Database aggregate status (rollup from attached forests) and links to: /forests?view=status&database-id=*database* /databases/*database*
/databases/{*id*|*name*}/counts	format={xml|json|html}	Database aggregate counts (rollup from attached forests) and link to: /databases/*database*
/databases/{*id*|*name*}/config	format={xml|json|html}	Database configuration and link to: /databases/*database*
/databases/{*id*|*name*}/edit		Links to the database configuration page in the Admin Interface.

Resource Address	Optional Parameters	Returns
/databases/{*id*|*name*}/{*custom*}	format={xml|json|html} {*custom*}	Provided by Plugin. For details, see "Extending Management API with Plugins" on page 131.
/hosts	view={default|schema|status} format={xml|json|html} group-id={*id*|*name*}	Summary of hosts in cluster and links to: /hosts/*all_hosts* / /hosts?view=status
/hosts/{*id*|*name*}	view={default|config|counts|edit|status} format={xml|json|html}	Host summary and links to: /groups/*group* /forests/*all_forests_in_host* /transactions?host-id=*host* /hosts /hosts/*host*/config /hosts/*host*/counts /hosts/*host*/edit /hosts/*host*/status
/hosts/{*id*|*name*}/config	format={xml|json|html}	Host configuration and link to: /hosts/*host*
/host/{*id*|*name*}/counts	format={xml|json|html}	Host aggregate counts and link to: /hosts/*host*
/hosts/{*id*|*name*}/status	format={xml|json|html}	Host status and links to: /forests?view=status&host-id=*host* /groups/*group*/status /hosts/*host*

Resource Address	Optional Parameters	Returns
/hosts/{*id*|*name*}/edit		Links to the host configuration page in the Admin Interface.
/hosts/{*id*|*name*}/{*custom*} ?group-id={*id*|*name*}	format={xml |json |html} {*custom*}	Provided by Plugin. For details, see "Extending Management API with Plugins" on page 131.
/servers	view={default |schema |status} format={xml |json |html} group-id={*id*|*name*} fullrefs={true|false}	Summary of servers in cluster and links to: /groups/*server_group* /servers/*all_servers*?group-id=*server_group* / /servers?view=status
/servers/{*id*|*name*} ?group-id={*id*|*name*}	view={default |config |edit |status} format={xml |json |html}	Server summary and links to: /groups/*server_group* /databases/*server_database* /requests?server-id=*server* &group-id=*server_group* /servers /servers/server/config ?group-id=*server_group* /servers/server/edit ?group-id=*server_group* /servers/server/status ?group-id=*server_group*
/servers/{*id*|*name*}/config ?group-id={*id*|*name*}	format={xml |json |html}	Server configuration and links to: /groups/*server_group* /servers/*server*?group-id=*server_group*

Resource Address	Optional Parameters	Returns
/servers/{*id*|*name*}/status ?group-id={*id*|*name*}	format={xml |json |html} host-id={*id*|*name*} fullrefs={true|false}	Server status and links to: /groups/*server_group*/status /hosts/*all_hosts*/status /databases/*database*/status /requests?server-id=*server* &group-id=*server_group* /servers/*server*?group-id= *server_group*
/servers/{*id*|*name*}/edit ?group-id={*id*|*name*}		Links to the server configuration page in the Admin Interface.
/servers/{*id*|*name*}/{*custom*}	format={xml |json |html} {*custom*}	Provided by Plugin. For details, see "Extending Management API with Plugins" on page 131.
/groups	view={default |schema} format={xml |json |html}	Summary of groups in the cluster and links to: /groups/*all_groups* /
/groups/{*id*|*name*}	view={default |config |counts |edit |status} format={xml |json |html}	Group summary and links to: /hosts/*all_hosts_in_group* /servers/*all_servers_in_group* /groups /groups/*group*/config /groups/*group*/counts /groups/*group*/edit /groups/*group*/status

Resource Address	Optional Parameters	Returns
/groups/{*id*|*name*}/status	format={xml |json |html}	Group aggregate status (rollup from attached hosts) and links to: /hosts?view=status&group-id=*group* /servers?view=status&group-id=*group* /groups/*group*
/groups/{*id*|*name*}/config	format={xml |json |html}	Group configuration and link to: /groups/*group*
/groups/{*id*|*name*}/counts	format={xml |json |html}	Group aggregate counts and link to: /groups/*group*
/groups/{*id*|*name*}/edit		Links to the group configuration page in the Admin Interface.
/groups/{*id*|*name*}/{*custom*}	format={xml |json |html} {*custom*}	Provided by Plugin. For details, see "Extending Management API with Plugins" on page 131.
/requests	format={xml |json |html} view=schema group-id={*id*|*name*} host-id={*id*|*name*} server-id={*id*|*name*} seconds-min=*seconds* fullrefs={true|false}	Summary of requests in the cluster and links to: /hosts/*host* /servers/*server*?group-id=*group* /databases/*database* /transactions/*transaction* ?host-id=*host* /requests/*all_requests* /

Resource Address	Optional Parameters	Returns
/requests/{id|uri}	format={xml |json |html} group-id={*id*|*name*} host-id={*id*|*name*} server-id={*id*|*name*}	Request summary and link to: /requests
/transactions	format={xml |json |html} view=schema host-id={*id*|*name*} seconds-min=*seconds* fullrefs={true|false}	Summary of transactions in the cluster and links to: /hosts/*host* /servers/*server*?group-id=*group* /databases/*database* /transactions/*transaction* ?host-id=*host* /
/transactions/{id|uri} ?host-id={*id*|*name*}	format={xml |json |html}	Transaction summary and link to: /hosts/*host* /transactions?host-id=*host*

* indicates one or more.

5.5 Obtaining the Options Node for a Resource Address

As described in Using the REST Library to Develop RESTful Web Services in the *Application Developer's Guide*, the REST Library uses an `options` node to map incoming requests to endpoints. The `options` node contains information about the communication options available on the request/response chain for the resource address, such as which parameters can be specified with the resource address.

You can use the `xdmp:http-options` function to output the `options` node for any resource address. For example, you can enter the following query in Query Console to display the `options` node for the `/manage/v1/transactions` resource address:

```
xdmp:http-options(
  "http://localhost:8002/manage/v1/transactions",
  <options xmlns="xdmp:http">
    <authentication method="digest">
      <username>admin</username>
      <password>admin</password>
    </authentication>
  </options>)
```

The output will include a `request` element that defines the options associated with the GET and HEAD methods for the resource address. From this, you can determine the supported parameters and values. For example, the above resource address supports the `view`, `seconds-min`, `host-id`, `fullrefs`, and `format` parameters, as shown below.

```
<rest:http method="GET">
  <rest:param name="view" values="default" default="default"/>
  <rest:param name="seconds-min" as="string"/>
  <rest:param name="host-id" as="string"/>
  <rest:param name="fullrefs" as="boolean" required="false"/>
  <rest:param name="format" as="string" values="xml|json|html"/>
  <rest:or>
    <rest:accept>application/xml</rest:accept>
    <rest:accept>application/json</rest:accept>
    <rest:accept>text/html</rest:accept>
    <rest:accept>application/x-javascript</rest:accept>
  </rest:or>
</rest:http>
```

5.6 Specifying the Management API Version

To guarantee stable behavior of the Management API as new versions are released, each resource address in the Management API includes a version number. The examples in this chapter show the version as LATEST, which means to use the latest version of the API. However, you can also specify the version number to use a specific version of the API, using the format:

```
v#
```

Where # is the version number. For example, in the initial version of the API:

```
http://localhost:8002/manage/LATEST/databases
```

is the same as:

```
http://localhost:8002/manage/v1/databases
```

If you want to update your clients when you choose, use the explicit version number. If you want to update your clients to the most recent version of the Management API, use LATEST.

Note: The version number is only updated when resource addresses and/or parameters have changed. It is not updated when resource addresses and/or parameters are added or removed.

5.7 Specifying Parameters in a Resource Address

resource addresses can take parameters to do the following:

- Specify the format of the returned view
- Return a filtered view

You can also write plugins to provide support for custom parameters, as described in “Extending Management API with Plugins” on page 131.

To specify multiple parameters, use the ‘?’ sign before the first parameter and the ‘&’ sign before any additional parameters:

```
http://host:port/manage/LATEST/resource?param1=value&param2=value....
```

Some resource addresses support optional parameters that are specific to that resource address. For example, to return monitoring information on the forests used by the Documents database, you can use the database-id parameter with the /forests resource as follows:

```
http://monitor_host:8002/manage/LATEST/forests?database-id=Documents
```

You can use combinations of filtering and `property` parameters. For example, to return the list cache ratio for the forests attached to the `Documents` database, you would use:

```
http://monitor_host:8002/manage/LATEST/forests?property=
list-cache-ratio&database-id=Documents
```

The remainder of this section describes the `format` and `property` parameters in more detail.

5.7.1 Formatting the Monitor Results

The application that issues a request to the Management API specifies the format for the returned view. For example, most Web browsers specify the default return format as HTML. If no return format is specified by the application, the view is formatted as XML. You can explicitly specify the view format by means of the `format` parameter at the end of the resource address:

```
format=value
```

Where *value* is either HTML, JSON, or XML.

The XML and JSON formats provide a rich set of data for your monitoring application. For example, to return an XML view of the entire cluster, you can enter the following in a browser:

```
http://monitor_host:8002/manage/LATEST?format=xml
```

This will return a `cluster` view in XML format. For example:

```
<cluster-default xsi:schemaLocation=
  "http://marklogic.com/manage/clusters manage-clusters.xsd">
  <meta>
    <uri>/manage/LATEST</uri>
    <current-time>2011-06-30T16:00:06.81-07:00</current-time>
    <elapsed-time units="sec">0.012</elapsed-time>
  </meta>
  <relations>
    <relation-group array="true">
      <uriref>/manage/LATEST/databases</uriref>
      <typeref>databases</typeref>
      <relation-count>13</relation-count>
    </relation-group>
    <relation-group array="true">
      <uriref>/manage/LATEST/forests</uriref>
      <typeref>forests</typeref>
      <relation-count>13</relation-count>
    </relation-group>
    <relation-group array="true">
      <uriref>/manage/LATEST/groups</uriref>
      <typeref>groups</typeref>
      <relation-count>1</relation-count>
    </relation-group>
```

```
      <relation-group array="true">
        <uriref>/manage/LATEST/hosts</uriref>
        <typeref>hosts</typeref>
        <relation-count>1</relation-count>
      </relation-group>
      <relation-group array="true">
        <uriref>/manage/LATEST/requests</uriref>
        <typeref>requests</typeref>
        <relation-count>1</relation-count>
      </relation-group>
      <relation-group array="true">
        <uriref>/manage/LATEST/servers</uriref>
        <typeref>servers</typeref>
        <relation-count>8</relation-count>
      </relation-group>
      <relation-group array="true">
        <uriref>/manage/LATEST/transactions</uriref>
        <typeref>transactions</typeref>
      </relation-group>
    </relations>
    <related-views>
      <related-view array="true">
        <view-type>item</view-type>
        <view-name>query</view-name>
        <view-uri>/manage/LATEST/query</view-uri>
      </related-view>
      <related-view array="true">
        <view-type>item</view-type>
        <view-name>status</view-name>
        <view-uri>/manage/LATEST/status</view-uri>
      </related-view>
    </related-views>
  </cluster-default>
```

5.7.2 Using the property Parameter on Forest Counts

The results returned by `/manage/LATEST/forests/{id|name}/counts` can be narrowed down by using the `property` parameter.

Properties are specified in the URL as:

```
property=value
```

The `property` parameter allows you to return the value of a specific value within the forest count content. For example, to return the `stands-counts` of the `Documents` forest, you can enter:

```
http://monitor_host:8002/manage/LATEST/forests/Documents/counts
?property=stands-counts
```

The `property` parameter filters the monitor content returned by the resource address. The results returned from the above resource address include the forest id and name, along with the metadata, `uri`, `current-time`, and `elapsed-time`.

```
forest-counts :
  id : 4514318629528848231
  name : Documents
  meta :
    uri :
/manage/LATEST/forests/Documents/counts?property=stands-counts
    current-time : 2011-06-29T13:24:08.469-07:00
    elapsed-time : 0.007
  count-properties :
    stands-counts :
     stand-counts :
       stand-id : 8558661396116142072
       path:C:\Program Files\MarkLogic\Data\Forests\Documents\00000008
       active-fragment-count : 86
       nascent-fragment-count : 0
       deleted-fragment-count : 0
  related-views :
    related-view : default
```

5.8 Interpreting the Output

The reference documentation for the Management API in the *XQuery and XSLT Reference Guide* describes each element in the XML output for each Management API resource address.

The main elements in the output from each resource address for an item or item list is shown in the table below.

Type of Resource Address	Element	Description
Item and Item View	`id`	The item id number.
Item and Item View	`name`	The item name (not available for requests and transactions).
Server Item	`server-kind`	The type of App Server (http, WebDAV, or XDBC)
Item, Item List, and View	`meta`	Metadata that describes: • The URI of the resource. • The current timestamp. • The number of seconds it took to execute the resource address.
View	*`view`*`-properties`	The properties of the item or item list view.
Item and Item List	`relations`	The items that are related to this item or item list.
Item List	`list-items`	The items in this item list
Item, Item List, and View	`related-views`	The views related to this item or item list. If an item, the item list view is also included.

6.0 Extending Management API with Plugins

You can write plugins to extend the capabilities of the Management API. This chapter describes how to write plugins for the Management API. This chapter includes the following sections:

- Types of Management API Plugins
- Sample Management API Plugins
- Management API Plugin Framework
- Creating Custom Resource Plugins
- Creating Custom View Plugins
- Creating Custom Format Plugins

6.1 Types of Management API Plugins

The Management API allows you to create the following types of plugins:

- Custom resource: Define a new resource in the API by naming a resource and a function to execute when that resource address is called. The function supports additional parameters defined by the plugin writer, as well as an option to add resource-specific formats, if desired. For example, you could write plugin to monitor active Information Studio tickets, accessed at /manage/v1/tickets.
- Custom format: Define a new format converter by defining an output media type, an output name, and a conversion function that maps the default XML to a custom format. For example, a you could write a plugin to convert existing XML to an Atom Publishing Protocol web feed.
- Custom view: Define a new or extended view of an existing resource in the API by defining a view name and a function to execute when the view is requested. The function supports additional parameters defined by the plugin writer, as well as an option to add resource-specific formats, if desired. For example, you could write a plugin for a view of the host resource that prunes host status to a few selected elements and adds some metadata.

6.2 Sample Management API Plugins

You can download sample plugins for the Management API from the MarkLogic Developer Community site:

http://developer.marklogic.com/

Move the plugin files to the `<marklogic-dir>/Assets/plugins/marklogic/manage/v1` directory and they will be ready to use.

The sample plugins for the Management API are:

Filename	Type	Description

`mimetype-list.xqy`	Custom resource	Provides a list of all mimetypes at the URI:
		`/manage/v1/mimetypes`
		This plugin also provides a view sorted by format of all mimetypes at the URI:
		`/manage/v1/mimetypes?view=byformat`

`mimetype-item.xqy`	Custom resource	Provides the default view for a single mimetype item at the URI:
		`/manage/v1/mimetypes/{id|name}`
		For example:
		`/manage/v1/mimetypes/application%2fxml`

`forest-basic.xqy`	Custom view	Provides a new, basic view of a single forest item at the URL: `/manage/v1/forests/{id|name}/basic`
`mimetype-counts.xqy`	Custom view	Provides a new view with counts over a list of mimetypes at the URI: `/manage/v1/mimetypes?view=counts`
`keyvalue-pairs.xqy`	Custom format	Provides a `keyvalue` option for the `format` parameter to convert any view to key-value pairs. For example: `/manage/v1/forests?format=keyvalue`

6.3 Management API Plugin Framework

The Plugin Framework allows you to create your own custom plugins for use by the Management API. The general Plugin Framework is described in the System Plugin Framework chapter in the *Application Developer's Guide*.

After creating a plugin, simply move the file to the `<marklogic-dir>/Assets/plugins/marklogic/manage/v1` directory and it will be ready to use. Place any modules referenced by your plugin in the `<marklogic-dir>/Modules` directory.

Every Management API plugin contains a map of function pointers to capabilities. The plugin module implements the functions defined in the capabilities map and registers them by calling the `plugin:register` function. The available capabilities for the plugins are described in the following sections:

- Custom Resource Capabilities and Function Signatures
- Custom View Capabilities and Function Signatures
- Custom Format Capabilities and Function Signatures

A function pointer takes the form:

```
xdmp:function(xs:QName("prefix:functionName"))
```

A capability takes the form:

```
"http://marklogic.com/appservices/manage/capability"
```

For example, the plugin portion of a custom view, named `myview`, would look like the following:

```
declare namespace myview =
  "http://marklogic.com/manage/plugin/myview";

declare default function namespace
  "http://www.w3.org/2005/xpath-functions";

declare option xdmp:mapping "false";

declare function myview:capabilities()
as map:map
{
  let $map := map:map()

  let $_ := map:put(
    $map,
    "http://marklogic.com/appservices/manage/get-property",
    xdmp:function(xs:QName("myview:get-property")))
```

```
  let $_ := map:put(
    $map,
    "http://marklogic.com/appservices/manage/convert-view",
    xdmp:function(xs:QName("myview:convert-view")))

  return $map
};

(: ... implement the functions in the capability map here ... :)
```

Use the `plugin:register` function to register the capabilities map for use by the Management API:

```
plugin:register(myview:capabilities(),"myview.xqy")
```

6.4 Creating Custom Resource Plugins

You can create plugins that produce a list view of multiple resources or an item view of a single resource. This section describes the capabilities used by a custom resource plugin, the function signatures used to implement each capability, and an example plugin.

6.4.1 Custom Resource Capabilities and Function Signatures

At a minimum, a resource plugin must do the following:

- Get the properties of the resource.
- Produce a view of the resource.

The following table describes all of the available collector capabilities and the function signatures used by plugins to implement the capabilities.

Capability	Description

`get-property`	Identifies the properties of this resource to the plugin framework. The resource properties are defined by specific keys, along with an optional language.
	
	The keys required for a custom resource are:
	
	• `resource-type`: The name used to identify a list of resources.
	• `item-name`: The name used to identify a single resource.
	• `view-type`: The type of view (`item` or `list`).
	• `view-names`: A sequence of the names of one or more views for this resource.
	
	Function signature:
	
	`plugin:get-property(`
	`  $key  as xs:string,`
	`  $lang as xs:string?`
	`) as xs:string*`

Capability	Description

`get-accepts`
 	
(optional)
	An optional capability that defines the formats supported by this resource, in order of preference. The formats are defined by their mimetype names, using the format:
	
	`<rest:accept>`*`mimetype_name`*`</rest:accept>`
	
	For example: `<rest:accept>application/xml</rest:accept>`
	
	Function signature:
	
	`plugin:get-accepts(`
	`) as element(rest:accept)*`

`get-parameters`
 	
(optional)
	An optional capability that defines the parameters supported by this resource.
	
	The parameters are defined using the format:
	
	`<rest:param` *`attributes`* `/>`
	
	Where *attributes* is any combination of attributes accepted for the `param` element, as described in The options Node in the *Application Developer's Guide*.
	
	For example:
	
	`<rest:param`
	`name="scene"`
	`as="integer"`
	`values="1|2|3"`
	`default="1"/>`
	
	Function signature:
	
	`plugin:get-parameters(`
	`) as element(rest:param)*`

Capability	Description
`produce-view`	Produces the views for this resource. If `produce-view` returns an XML structure with a single root element, built-in and plugin formats can be applied to the view. Otherwise `produce-view` is responsible for the format of the view. Any additional formats declared in the `get-accepts` function must also be produced by this function. The `query-params` parameter is a map of the query parameters extracted from the request URL. Function signature: `plugin:produce-view( $query-params as map:map ) as item()*`

6.4.2 Example Custom Resource Plugin

The `mimetype-list.xqy` plugin is an example of a custom resource plugin. This plugin returns a list of all mimetypes configured in MarkLogic Server.

```
xquery version "1.0-ml";

declare namespace mlist =
    "http://marklogic.com/manage/plugin/mimetype-list";

import module namespace plugin =
    "http://marklogic.com/extension/plugin"
       at "/MarkLogic/plugin/plugin.xqy";

(: used by this specific plugin :)
import module namespace admin = "http://marklogic.com/xdmp/admin"
     at "/MarkLogic/admin.xqy";

(: optional - needed for get-accepts() or get-parameters() :)
declare namespace rest = "http://marklogic.com/appservices/rest";

(: used by this specific plugin :)
declare namespace mt = "http://marklogic.com/xdmp/mimetypes";

declare default function namespace
    "http://www.w3.org/2005/xpath-functions";

declare option xdmp:mapping "false";
```

```
declare function mlist:capabilities(
  as map:map
{
  let $map := map:map()
  return (
     map:put(
        $map,
        "http://marklogic.com/appservices/manage/get-property",
        xdmp:function(xs:QName("mlist:get-property"))),
     map:put(
        $map,
        "http://marklogic.com/appservices/manage/get-accepts",
        xdmp:function(xs:QName("mlist:get-accepts"))),
     map:put(
        $map,
        "http://marklogic.com/appservices/manage/get-parameters",
        xdmp:function(xs:QName("mlist:get-parameters"))),
     map:put(
        $map,
        "http://marklogic.com/appservices/manage/produce-view",
        xdmp:function(xs:QName("mlist:produce-view"))),
     $map )
};

declare function mlist:get-property(
    $key  as xs:string,
    $lang as xs:string?
) as xs:string*
{
        if ($key eq "resource-type") then "mimetypes"
   else if ($key eq "item-name"    ) then "mimetype"
   else if ($key eq "view-type"    ) then "list"
   else if ($key eq "view-names"   ) then ("default", "byformat")
   else ()
};

(: Return conversions supported only for this resource in order of
   preference. :)
declare function mlist:get-accepts(
) as element(rest:accept)*
{
    <rest:accept>text/csv</rest:accept>
};

(: Return parameters for this resource. :)
declare function mlist:get-parameters(
) as element(rest:param)*
{
    <rest:param
       name="maintype"
       values="all|application|audio|example|image|message|model
               |multipart|text|video"
       default="all"/>
};
```

```
(: Return the representation for this resource in a format handled
   by converters, the representation must be a single XML element. :)
declare function mlist:produce-view(
    $query-params as map:map
) as item()*
{
  let $format := (map:get($query-params,"format"),"xml")[1]
  (: construct the view of the resource in XML format :)
  let $view   :=
    <mimetypes xmlns="http://marklogic.com/xdmp/mimetypes">{
      let $view-type := (map:get($query-params,"view"),"default")[1]
      let $maintype  := (map:get($query-params,"maintype"),"all")[1]
      for $mimetype in admin:mimetypes-get(admin:get-configuration())
        let $format := $mimetype/mt:format/string(.)
        let $name   := $mimetype/mt:name/string(.)
        where
          if ($maintype eq "all") then true()
          else (substring-before($name,"/") eq $maintype)
        order by
               if ($view-type eq "byformat") then $format
          else if ($view-type eq "default")  then $name
          else error(xs:QName("Error"),
             concat("Unknown mimetype list view: ",$view-type))
          ,
          if ($view-type eq "byformat") then $name
          else ()
      return $mimetype
    }</mimetypes>

return

(: the plugin provides a special conversion only to CSV and only for
   this resource. Generic converters can convert the XML for this
   resource to other formats :)
  if ($format ne "csv")
  then $view
  (: join the list of mimetypes with newlines and the details fields
     with commas :)
  else $view/mt:mimetype/string-join(
    string-join(
      (mt:name/string(.),mt:extensions/string(.),mt:format/string(.)),
      "," ),
    "
" )
};

(: register the plugin :)
plugin:register(mlist:capabilities(), "mimetype-list.xqy")
```

6.5 Creating Custom View Plugins

You can create plugins that produce a list view of resources, an item view of a particular resource, or an extended view. This section describes the capabilities used by a custom view plugin, the function signatures used to implement each capability, and an example plugin.

6.5.1 Custom View Capabilities and Function Signatures

At a minimum, a resource plugin must do the following:

- Get the properties of the resource.
- Extend the view of the resource.

The following table describes all of the available collector capabilities and the function signatures used by plugins to implement the capabilities.

Capability	Description

get-property	Identifies the properties of this view to the plugin framework. The view properties are defined by specific keys, along with an optional language.
	
	The keys required for a custom view are:
	
	• `resource-type`: The name of the resource for which this plugin produces views.
	• `view-type`: The type of view (`item` or `list`).
	• `extended-view`: The view to be extended by this plugin. Or an empty sequence, if the view doesn't exist.
	• `view-names`: A sequence of one or more names for the views produced by this plugin.
	
	Function signature:
	
	`plugin:get-property(`
	`  $key  as xs:string,`
	`  $lang as xs:string?`
	`) as xs:string*`

extend-view	Produces a view of the resource. This function can either take one view as input and convert it to another view or produce a view without input.
	
	Function signature:
	
	`plugin:extend-view(`
	`  $query-params as map:map,`
	`  $extended-view as element()?`
	`) as element()?`

6.5.2 Example Custom View Plugin

The `forest-basic.xqy` plugin is an example of a custom view plugin. This plugin produces a view of `/manage/v1/forests/{id|name}/counts`, excluding the `stands-counts` portion.

```
xquery version "1.0-ml";

(: Copyright 2011 MarkLogic Corporation.  All Rights Reserved. :)

declare namespace fbasic =
    "http://marklogic.com/manage/plugin/forest-basic";

import module namespace plugin =
    "http://marklogic.com/extension/plugin"
       at "/MarkLogic/plugin/plugin.xqy";

(: used by this specific plugin :)
declare default element namespace
    "http://marklogic.com/xdmp/status/forest";

declare default function namespace
    "http://www.w3.org/2005/xpath-functions";
       declare option xdmp:mapping "false";

declare function fbasic:capabilities(
) as map:map
{
  let $map := map:map()
  return (
    map:put($map,
            "http://marklogic.com/appservices/manage/get-property",
            xdmp:function(xs:QName("fbasic:get-property"))),
    map:put($map,
            "http://marklogic.com/appservices/manage/extend-view",
            xdmp:function(xs:QName("fbasic:extend-view"))),
    $map )
};

(: the name of the type of resource :)
declare function fbasic:get-property(
    $key  as xs:string,
    $lang as xs:string?
) as xs:string*
{
        if ($key eq "resource-type") then "forests"
   else if ($key eq "view-type"    ) then "item"
   else if ($key eq "extended-view") then "counts"
   else if ($key eq "view-names"   ) then ("basic")
   else ()
};
```

```
(: return the XML representation for the requested view. The
   $context parameter contains the query parameters and properties
  of the execution context. :)
declare function fbasic:extend-view(
    $query-params as map:map,
    $extended-view as element(forest-counts)?
) as element(forest-basic)?
{
    <forest-basic>
      {$extended-view/(* except stands-counts)}
    </forest-basic>
};

(: register the plugin :)
plugin:register(fbasic:capabilities(), "forest-basic.xqy")
```

6.6 Creating Custom Format Plugins

You can create plugins that produce a list view of resources, an item view of a particular resource, or an extended view. This section describes the capabilities used by a custom format plugin, the function signatures used to implement each capability, and an example plugin.

6.6.1 Custom Format Capabilities and Function Signatures

At a minimum, a format plugin must do the following:

- Get the properties of the format.
- Convert the view to the format.

The following table describes all of the available collector capabilities and the function signatures used by plugins to implement the capabilities.

Capability	Description

get-property	Identifies the properties of this format to the plugin framework. The format properties are defined by specific keys, along with an optional language.
	
	The keys required for a custom format are:
	
	• `output-name`: The name for this format.
	• `output-type`: The mimetype for this format.
	
	Function signature:
	
	`plugin:get-property(`
	`  $key  as xs:string,`
	`  $lang as xs:string?`
	`) as xs:string*`

Capability	Description

convert-view	Converts the XML representation of the resource to the desired format.
	
	Function signature:
	
	`plugin:convert-view(`
	`  $view as element()`
	`) as xs:string`

6.6.2 Example Custom Format Plugin

The `keyvalue-pairs.xqy` plugin is an example of a custom format plugin.

```
xquery version "1.0-ml";

(: Copyright 2011 MarkLogic Corporation.  All Rights Reserved. :)

declare namespace keyval =
    "http://marklogic.com/manage/v1/plugin/keyvalue-pairs";

import module namespace plugin =
    "http://marklogic.com/extension/plugin"
      at "/MarkLogic/plugin/plugin.xqy";

declare default function namespace
    "http://www.w3.org/2005/xpath-functions";

declare option xdmp:mapping "false";

declare function keyval:capabilities(
) as map:map
{
  let $map := map:map()
    return (
      map:put($map,
              "http://marklogic.com/appservices/manage/get-property",
              xdmp:function(xs:QName("keyval:get-property"))),
      map:put($map,
              "http://marklogic.com/appservices/manage/convert-view",
              xdmp:function(xs:QName("keyval:convert-view"))),
      $map )
};
```

```
(: the name of the output format :)
declare function keyval:get-property(
    $key  as xs:string,
    $lang as xs:string?
) as xs:string*
{
        if ($key eq "output-name") then "keyvalue"
   else if ($key eq "output-type") then "text/plain"
   else ()
};

(: convert the XML input to the output format :)
declare function keyval:convert-view(
    $view as element()
) as xs:string
{
  fn:string-join(
    for $elem in $view//*[empty(*)]
    let $key :=
       fn:string-join(
         fn:reverse($elem/ancestor-or-self::*)/fn:local-name(),
         ".")
    return concat($key, "=", string($elem)),
    (: separate with a line break :)
    "
" )
};

(: registration for the plugin :)
plugin:register(keyval:capabilities(), "keyvalue-pairs.xqy")
```

7.0 Technical Support

MarkLogic provides technical support according to the terms detailed in your Software License Agreement or End User License Agreement. For evaluation licenses, MarkLogic may provide support on an “as possible” basis.

For customers with a support contract, we invite you to visit our support website at http://support.marklogic.com to access information on known and fixed issues.

For complete product documentation, the latest product release downloads, and other useful information for developers, visit our developer site at http://developer.marklogic.com.

If you have questions or comments, you may contact MarkLogic Technical Support at the following email address:

support@marklogic.com

If reporting a query evaluation problem, please be sure to include the sample XQuery code.

Combined Product Notices

MarkLogic Server 6.0 and supporting products.
Last updated: August 19, 2013

COPYRIGHT

Copyright © 2013 MarkLogic Corporation. All rights reserved.
This technology is protected by U.S. Patent No. 7,127,469B2, U.S. Patent No. 7,171,404B2, U.S. Patent No. 7,756,858 B2, and U.S. Patent No 7,962,474 B2.

The MarkLogic software is protected by United States and international copyright laws, and incorporates certain third party libraries and components which are subject to the attributions, terms, conditions and disclaimers set forth below.

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

2. Highslide Software from Highslide Software Torstein Honsi. All Highslide Software is protected by local and international copyright laws. All rights reserved.

3. Icons developed by Yusuke Kamiyamane. Copyright © 2011 Yusuke Kamiyamane. All rights reserved. Icons are licensed subject to http://creativecommons.org/licenses/by/3.0/legalcode.

4. Antenna House OfficeHTML Copyright © 2000-2008 Antenna House, Inc. All rights reserved.

5. Argus Copyright ©1999-2008 Iceni Technology Ltd. All rights reserved.

6. Rosette Linguistics Platform 6.5.2/6.5.3 from Basis Technology Corporation, Copyright © 2004-2008 Basis Technology Corporation. All rights reserved.

7. ISYS Search. Copyright © 2011 ISYS™ Search Software, Inc. All rights reserved.

8. Software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/) Copyright © 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved. Copyright © 1998-2011 The OpenSSL Project. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)" 4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org. 5. Products derived from this

software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project. 6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)" THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com). Additional terms may apply to the foregoing software as further set forth at http://www.openssl.org/source/license.html

9. ICU v. 4.2.1 available from http://site.icu-project.org// Copyright © 1995-2011 International Business Machines Corporation and others. All rights reserved. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

10. Tidy available at http://www.w3.org/People/Raggett/tidy/. Copyright © 1998-2008 World Wide Web Consortium (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). All Rights Reserved.

11. 'zlib' general purpose compression library v. 1.2.3. Copyright © 1995-2010 Jean-loup Gailly and Mark Adler. http://www.zlib.net/zlib_license.html

12. RSA Data Security, Inc. MD5 Message-Digest Algorithm, copyright © 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved. License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function. License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work. RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind. These notices must be retained in any copies of any part of this documentation and/or software.

13. TRE software, available at http://laurikari.net/tre/. Copyright © 2001-2009 Ville Laurikari <vl@iki.fi>. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: (1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

14. Eigen Library, version 2.0.10, available at http://eigen.tuxfamily.org/index.php?title=Main_Page. Eigen Library is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. Eigen Library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details. You should have received a copy of the GNU Lesser General Public License along with Eigen Library. If not, see: http://www.gnu.org/copyleft/lesser.html

15. FunctX XQuery Library v.1.0 available at http://www.xqueryfunctions.com/xq/, Copyright © Datypic (Priscilla Walmsley). You can redistribute such library and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. This library is distributed in the

hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License as available at http://www.gnu.org/licenses/lgpl-2.1.html#SEC4 for more details.

16. DeployJava.js v. 1.8, Copyright © 2006, 2011, Oracle and/or its affiliates. All rights reserved. ORACLE PROPRIETARY/CONFIDENTIAL. Use of the foregoing library is subject to the following license terms. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Oracle nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

17. YUI libraries v.2.8 available at http://developer.yahoo.com/yui/. Copyright © 2011 Yahoo! Inc. All rights reserved. Use of the foregoing libraries is subject to the following license terms. Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Yahoo! Inc. nor the names of YUI's contributors may be used to endorse or promote products derived from this software without specific prior written permission of Yahoo! Inc. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING

NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Additional notices related to components included in the foregoing library are also available at the following URL: http://yuilibrary.com/license/

18. XSLTForms libraries available at http://sourceforge.net/projects/xsltforms/ . You can redistribute such libraries and/or modify them under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version. These libraries are distributed in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License as available at http://www.gnu.org/licenses/lgpl-2.1.html#SEC4 for more details.

19. jQuery v.1.7 libraries available at http://jquery.com/ and jQuery UI v.1.8.2 libraries available at http://jqueryui.com/. Copyright © 2011 John Resig. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OFMERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE ANDNONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BELIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTIONOF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTIONWITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

20. JSON v. 2 libraries available at http://www.json.org/json2.js. Copyright © 2002 JSON.org. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. The Software shall be used for Good, not Evil. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

21. CodeMirror libraries available at http://codemirror.net/. Copyright © 2011 by Marijn Haverbeke marijnh@gmail.com. Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

22. History.js available https://github.com/balupton/History.js/ Copyright © 2011, Benjamin Arthur Lupton. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Benjamin Arthur Lupton nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

23. Sample content provided for use with the software is copyrighted by the respective authors as contributed by each of them to Wikipedia pursuant to the Creative Commons Attribution-ShareAlike License found at http://creativecommons.org/licenses/by-sa/3.0/.

24. Visual Studio Tools for Office 3.0 (Microsoft VSTO 3.0) licensed under Microsoft Public License Microsoft Public License (Ms-PL) found at http://msdn.microsoft.com/en-us/library/ff647676.aspx.

25. The MarkLogic Connector for Hadoop contains Apache Jakarta Commons Modeler. Copyright 2001-2007 The Apache Software Foundation. Licensed under the Apache License, Version 2.0 (the "License"). You may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

26. jemalloc http://www.canonware.com/jemalloc/index.html, released under the terms of the following BSD-derived licenses: Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice(s), this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice(s), this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER(S) ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER(S) BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Portions of the software are copyright (C) 2002-2013 Jason Evans, and portions copyright (C) 2009-2013 Facebook, Inc. and copyright (C) 2007-2012 Mozilla Foundation. Neither the name of Facebook, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. All rights reserved.

27. Gziphelper; version: 1.0, Feb, 2003, authored by Gao Dasheng, is distributed subject to the following terms: This software is provided 'as-is', without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software. Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions: (1.) the origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required; (2) altered source versions must be plainly marked as such, and must not be misrepresented as being the original software; (3) this notice may not be removed or altered from any source distribution.

28. Some features of MarkLogic allow you to develop and use Google Maps. Review Google Map's privacy policy at http://www.google.com/policies/privacy/, legal notices at http://maps.google.com/help/legalnotices_maps.html, and terms of use at http://maps.google.com/help/terms_maps.html.

29. commons-codec, HttpComponents, commons-logging, and Google Maps Utility are used under the Licensed under the Apache License, Version 2.0 (the "License"). You may not use these files except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

30. Mapstraction. Copyright (c) 2012, Mapstraction. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

31. logback-classic and logback-core are provided in object code form only under the Eclipse Public License - v 1.0 available at http://www.eclipse.org/legal/epl-v10.html. Copyright (C) 1999-2012, QOS.ch. All rights reserved. Source code for the Program is available http://logback.qos.ch/.

32. Simple Logging Facade for Java (SLF4J). Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

33. cURL v. 7.22.0 available at http://curl.haxx.se/. Copyright © 1996 - 2011, Daniel Stenberg, <daniel@haxx.se>. All rights reserved. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

34. heatmap.js distributed subject to the "THE BEER-WARE LICENSE" (Revision 42): <phk@FreeBSD.ORG> wrote this file. As long as you retain this notice you can do whatever you want with this stuff. If we meet some day, and you think this stuff is worth it, you can buy me a beer in return Poul-Henning Kamp.

35. Except for components that are otherwise subject to other open source licenses, the MarkLogic ODBC Driver is released under the GNU Lesser General Public License as published by the Free Software Foundation, version 2.1 of the License.

36. Solr-commons-csv-3.5.0 Copyright 2005-2006 The Apache Software Foundation. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, this file is distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

37. Avro v. 1.7.1. Copyright © The Apache Software Foundation. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, this file is distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

38. Guava v. 11.0.2. Copyright © Google. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, this file is distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

39. Hadoop-auth v. 2.0.0. Copyright © The Apache Software Foundation. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, this file is distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

40. hadoop-hdfs v. 2.0. Copyright © The Apache Software Foundation. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, this file is distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

41. commons-lang v. 2.4, 2.5. Copyright © The Apache Software Foundation. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, this file is distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

42. commons-configuration v. 2.0. Copyright © The Apache Software Foundation. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, this file is distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

43. jackson-core-asl v. 1.8.8. Copyright © FasterXML. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, this file is distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

44. jackson-mapper-asl v. 1.8.8. Copyright © FasterXML. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, this file is distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

45. log4j v. 1.2.16. Copyright © The Apache Software Foundation. Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, this file is distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

46. protobuf-java v. 2.4.0a. Copyright © Google. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Google nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

47. xpp3 v. 1.1.3.3. Copyright © 2002 Extreme! Lab, Indiana University. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Indiana University Extreme! Lab(http://www.extreme.indiana.edu/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. The names "Indiana Univeristy" and "Indiana Univeristy Extreme! Lab" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact http://www.extreme.indiana.edu/. Products derived from this software may not use "Indiana Univeristy" name nor may "Indiana Univeristy" appear in their name, without prior written permission of the Indiana University. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS, COPYRIGHT HOLDERS OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IFADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

48. xstream v. 1.4.2. Copyright © 2003-2006, Joe Walnes. Copyright © 2006-2009, 2011 XStream Committers. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of XStream nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR

BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

TRADEMARK NOTICE

The MarkLogic name and logo are registered trademarks of MarkLogic Corporation in the United States and/or other countries. Excel and PowerPoint are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Any other names or logos included in the Software, this notice or the documentation are property of the respective trademark owners.