MarkLogic Server

Flexible Replication Guide

MarkLogic 6
September, 2012

Last Revised: 6.0-1, September, 2012

Copyright © 2012 MarkLogic Corporation. All rights reserved.

Flexible Replication Guide

1.0 Flexible Replication in MarkLogic Server 4
- 1.1 Terms Used in this Guide 4
- 1.2 Understanding Flexible Replication 5

2.0 Flexible Replication Quick Start 9
- 2.1 Creating Master and Replica Databases 9
- 2.2 Configuring a Flexible Replication Pipeline for the Master Database 10
- 2.3 Creating a Replication App Server 13
- 2.4 Configuring Push Replication on the Master Database 15
- 2.5 Scheduling Replication Push Task 17
- 2.6 Loading Documents into the Master Database and Checking Replication 19

3.0 Configuring Replication 20
- 3.1 Replication Security 20
- 3.2 Defining Replicated Domains 22
- 3.3 Configuring Replication App Servers 25
- 3.4 Configuring Push Replication 26
- 3.5 Creating a Scheduled Replication Task 29
  - 3.5.1 Creating a Push-Local-Forests Replication Task 33
  - 3.5.2 Creating a Pull Replication Task 35
- 3.6 Configuring Pull Replication 36
- 3.7 Backing Up, Restoring, and Clearing the Master and Replica Databases 40

4.0 Filtering Replicated Documents 41
- 4.1 Configuring MarkLogic Server to use a Replication Filter Module 41
- 4.2 Creating a Filter Module 42
- 4.3 Example Filters 43
  - 4.3.1 Adding a Collection 44
  - 4.3.2 Changing the Document Quality 47
  - 4.3.3 Adding Document Permissions 48
  - 4.3.4 Adding a Forest Name 50
  - 4.3.5 Changing the Document URI 52
  - 4.3.6 Changing a Document Element 53
  - 4.3.7 Prohibiting Replication on Select Documents 54
- 4.4 Setting Filter Options 55

5.0 Checking Replication Status 57
- 5.1 Special Circumstances that Impact Replication 57
- 5.2 Checking the Replication Status of a Master Database 57
- 5.3 Checking the Replication Status of a Domain 60
- 5.4 Checking the Replication Status of a Target 61

6.0 Technical Support 64

Combined Product Notices 65
COPYRIGHT 65
TRADEMARK NOTICE 77

1.0 Flexible Replication in MarkLogic Server

This chapter describes Flexible Replication in MarkLogic Server in general terms, and includes the following sections:

- Terms Used in this Guide
- Understanding Flexible Replication

Note: To enable Flexible Replication, a license key that includes Flexible Replication is required. For details on purchasing a Flexible Replication license, contact your MarkLogic sales representative.

1.1 Terms Used in this Guide

The following are the definitions for the replication terms used in this guide:

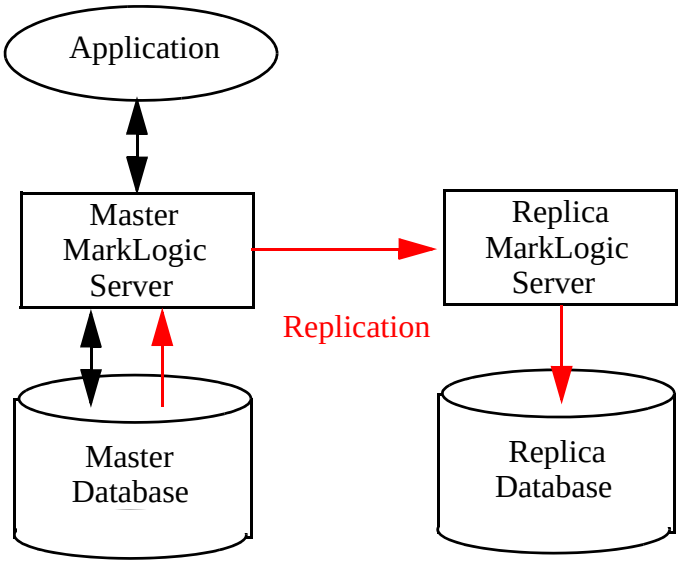
- To *Replicate* is to create a copy of a document in another database and to keep that copy in sync (possibly with some time-lag/latency) with the original.
- The *Master* is the repository that gets updated by the applications. The master, in turn, replicates the updates to other repositories, know as replicas.
- A *Replica* is a repository that receives replicated updates from the master.
- A *Master Copy* is the content being replicated. For any piece of replicated content there is a master and at least one copy.
- A *Master Cluster* is the cluster on which the replicated documents are updated by the applications.
- *Flexible Replication* is an implementation of replication based on the MarkLogic Server Content Processing Framework (CPF). Flexible replication is single-master, asynchronous, and provides a medium level of throughput and latency.
- *Replication Domain* is the specification of the set of documents to be replicated. This may be a collection or some other set definition.
- A *Filter* is an XQuery program that modifies the replicated documents in some manner, determines whether to replicate a change, or selects which parts of a document will be replicated.
- *Asynchronous Replication* refers to a configuration in which the Master does not wait for confirmation that the update has been received by the Replica before sending further updates. Flexible Replication is asynchronous.
- *Transaction-aware* refers to a configuration in which all updates that make up a transaction on the master are applied as a single transaction on the replica.
- *Zero-day Replication* refers to replicating the data in the replicated domains that existed before replication was configured.

1.2 Understanding Flexible Replication

Flexible Replication is the process of maintaining copies of data on multiple MarkLogic Servers. The purpose of replication is to make data continuously available to mission-critical applications and to enhance application performance. Some of the benefits of replication include:

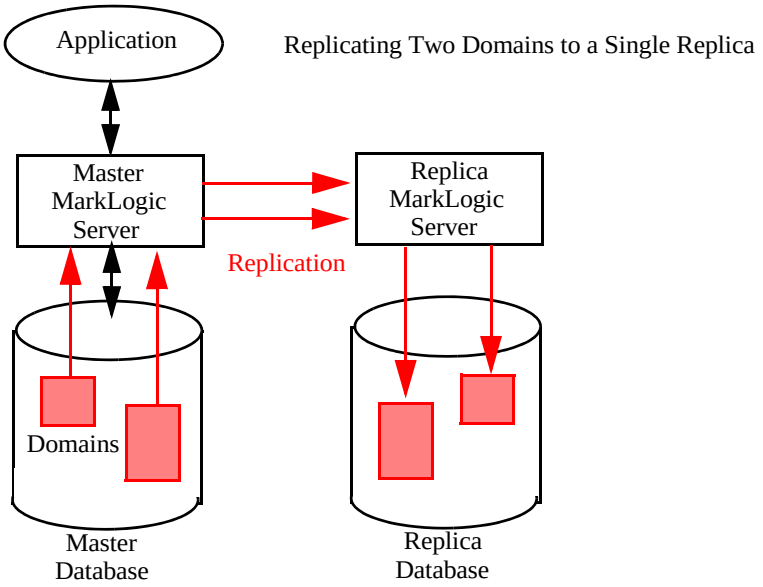
- High Availability: You can maintain duplicate data on two or more MarkLogic Servers. In the event of a software or hardware failure on one server, the data is available from another server.
- Disaster Recovery: In the event of some irreversible disaster on the Master server, duplicate data is preserved on its Replica.
- Performance: Companies with geographically dispersed clusters can use replication to maintain common data on each local cluster. Queries and updates done locally, are faster and the workload can be scaled across clusters, so that each cluster handles less of the query and update load.

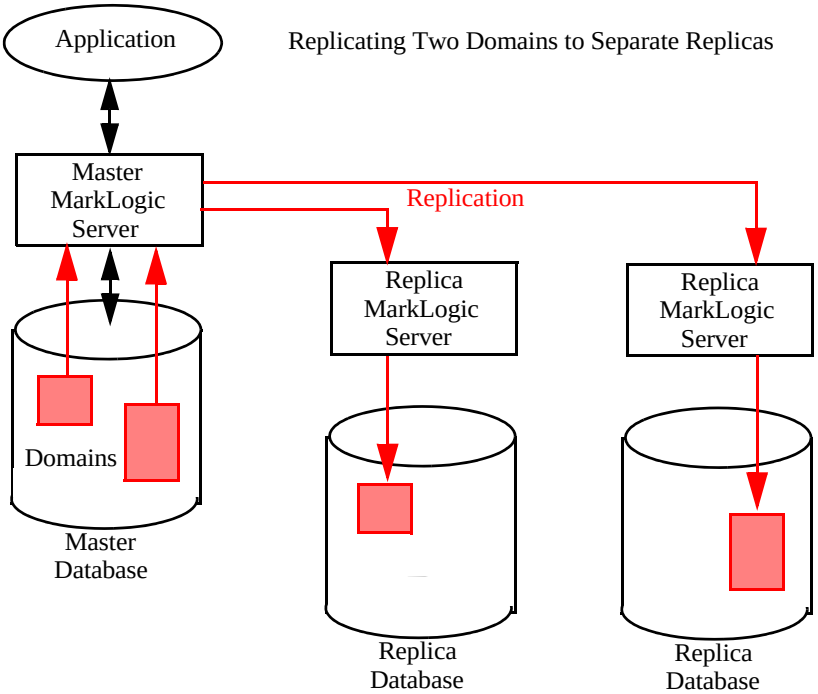
In a replicated environment, the original content is created by an application on the Master MarkLogic Server. Replication then copies the content to one or more Replica MarkLogic Servers. The Master and Replica servers are typically in different clusters, which may be in the same location or in different locations.



Flexible Replication is asynchronous, which means that the Master does not wait for confirmation that the update has been received by the Replica before sending further updates. Replication from the Master to the Replica occurs as soon as possible after the document is added or updated by the application.

MarkLogic Server uses the Content Processing Framework (CPF) as the underlying replication mechanism. The documents to be replicated are defined by a CPF domain. The scope of a domain may be a document, a collection of documents, or a directory. For more details about domains, see Understanding and Using Domains in the *Content Processing Framework Guide* Guide. You can replicate multiple domains either to the same Replica or to different Replicas, as shown is the illustrations below.

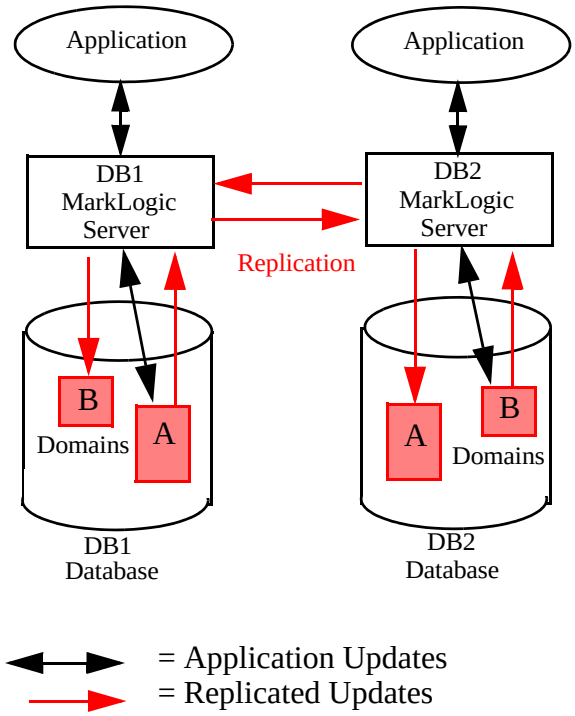




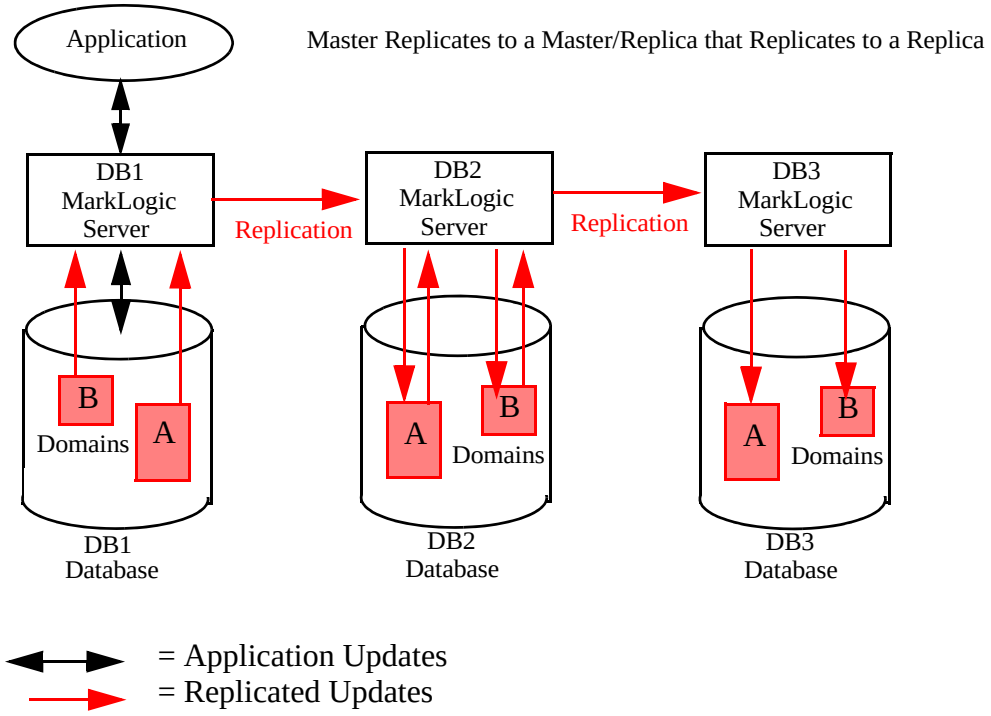
Replicated databases do not necessarily need to be configured as entirely a Master or a Replica in the replication scheme. For example, you may have two databases, DB1 and DB2, where DB1 replicates updates to the documents in Domain A to DB2 and DB2 replicates updates to the documents in Domain B to DB1.

Note: This is not a multi-master replication configuration, as the documents updated by each application must be in different domains. Any overlap between the replicated domains may result in unpredictable behavior.

Replicating Different Domains Between Databases



Another possible Master/Replica configuration is where a Master replicates updates to a Replica/Master that replicates the updates to another Replica. For example, you may have three databases, DB1, DB2, and DB3, where DB1 replicates updates to DB2 and DB2 replicates updates to DB3.



You can set up filters that narrow the scope of what documents and what parts of the documents are replicated within a domain. For example, you can set up a filter to replicate only XML documents, or you can create filters to only replicate inserts and updates (not deletes), or only replicate a particular node or element within each document.

Replication can be configured to either *push* or *pull* updates from the Master to the Replica. Push replication means that the Master pushes updates to the Replica. Pull replication means the Replica pulls updates from the Master. Push replication is triggered whenever an update is made on the Master database. Pull replication can only be configured as a scheduled task, as described in “Configuring Pull Replication” on page 36. Typically, you should use push replication, unless the Master and Replica are separated by a firewall through which a Replica server can only pull content from a Master server outside the firewall.

2.0 Flexible Replication Quick Start

This chapter provides the quick-start procedures for creating a simple flexible replication configuration on a single MarkLogic Server. Typical flexible replication configurations will have the Master and Replica databases on different MarkLogic Servers located on different hosts. Replicated MarkLogic Servers may reside in the same cluster or in different clusters. In order to keep the hardware requirements to a minimum, this quick start describes how to set up replication between two databases on the same MarkLogic Server.

This chapter includes the following sections:

- Creating Master and Replica Databases
- Configuring a Flexible Replication Pipeline for the Master Database
- Creating a Replication App Server
- Configuring Push Replication on the Master Database
- Scheduling Replication Push Task
- Loading Documents into the Master Database and Checking Replication

All of the procedures described in this chapter are done using the Admin Interface described in the Administrative Interface chapter in the *Administrator's Guide*.

2.1 Creating Master and Replica Databases

Before attempting to configure replication, create a Master and a Replica database. In these examples, the databases are named Master and Replica. However, you can use any name you wish for either the master or its replica.

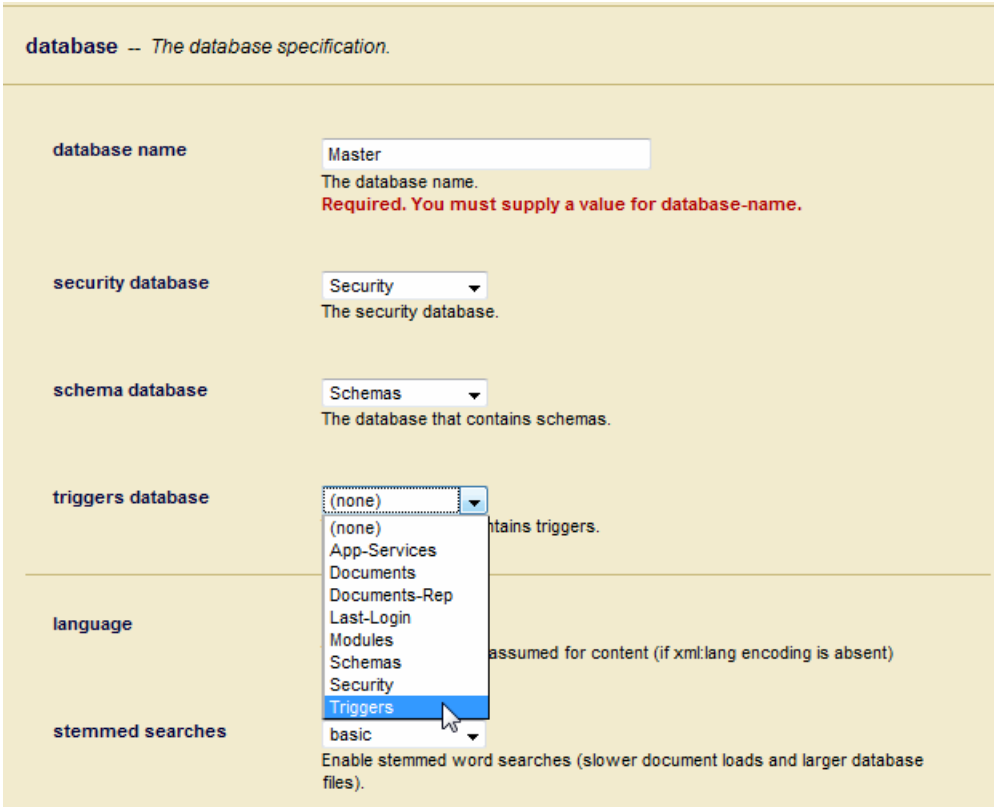
1. Create two forests, one for the Master database and one for the Replica database. For details on creating forests, see Creating a Forest in the *Administrator's Guide*.

2. Create two databases, Master and Replica, and attach the respective forest to each database. For details on creating databases and attaching forests, see Creating a New Database and Attaching and/or Detaching Forests to/from a Database in the *Administrator's Guide*.

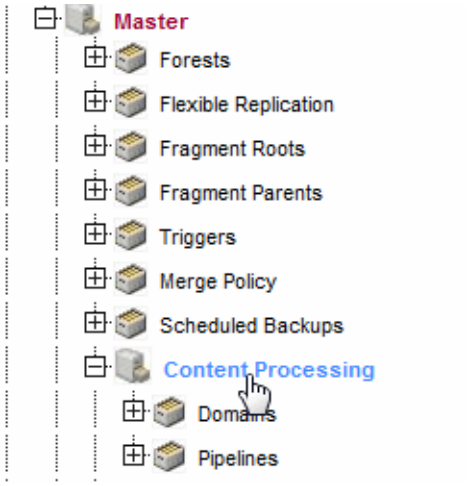
2.2 Configuring a Flexible Replication Pipeline for the Master Database

This section describes how to configure the Content Processing Framework (CPF) for the replicated domains on the Master database. In order to replicate data, CPF on the Master domains must be configured with the Status Change Handling and Flexible Replication pipelines. In this example, we use the existing Triggers database. However, if you are replicating more than one database, you should create a separate CPF database for each master database.

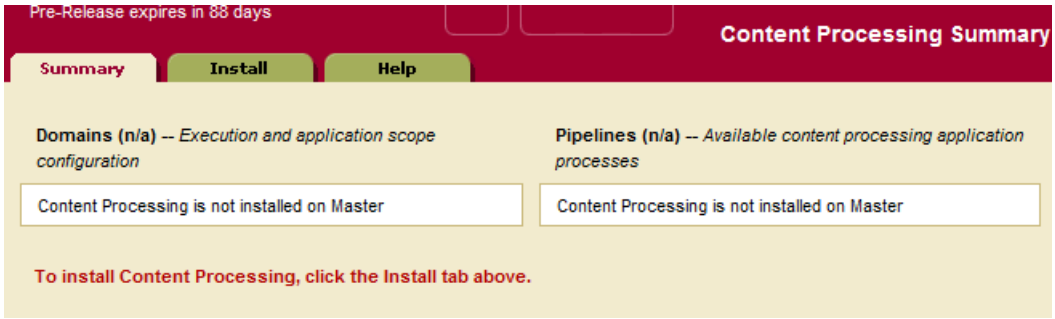
1. Configure the Master database to use a triggers database.



2. Select Content Processing on the Master database:



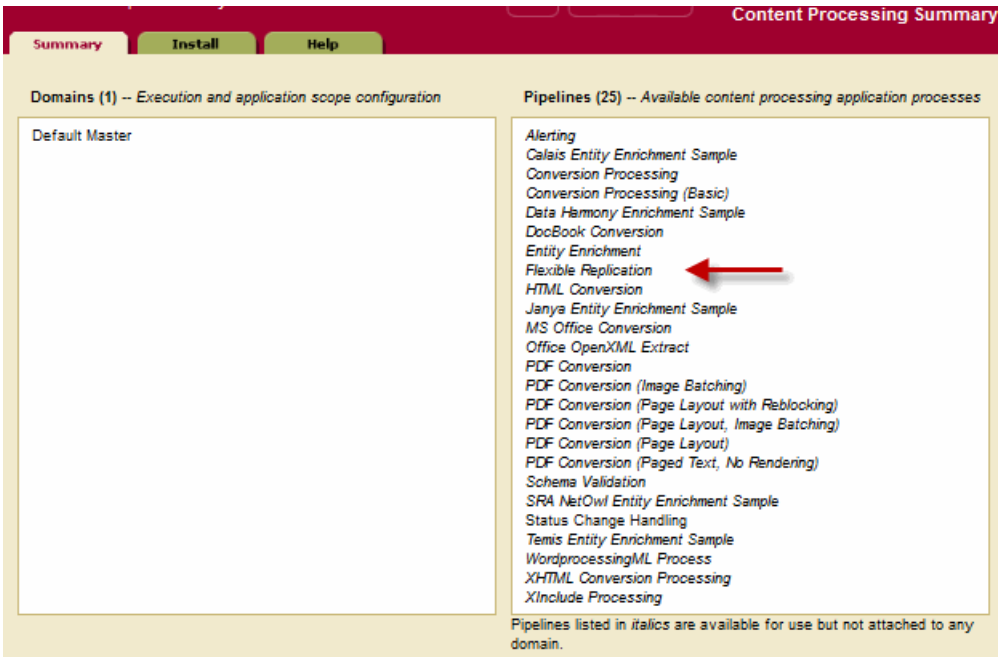
3. In the Content Processing Summary, click on the Install tab:



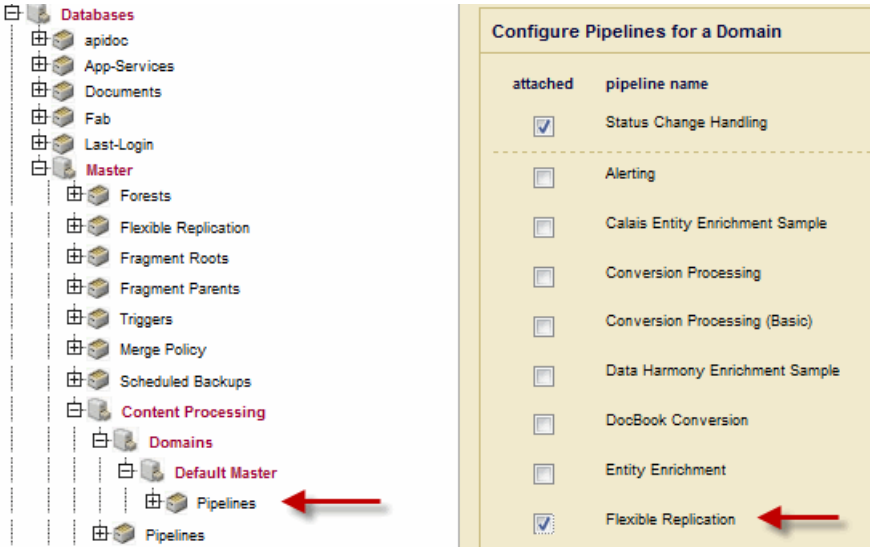
4. On the Content Processing Installation page, click Install:



5. Confirm that Flexible Replication appears in the Pipelines section of the Content Processing Summary page:



6. On the left-hand navigation menu, navigate to the Content Processing > Domains > Default Master > Pipelines page. Select the Status Change Handling and Flexible Replication pipelines and click OK:

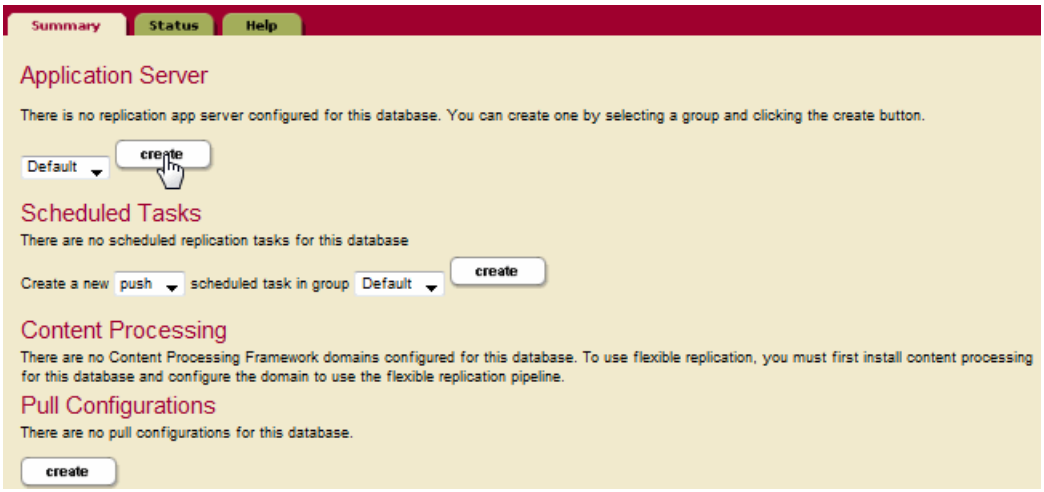


2.3 Creating a Replication App Server

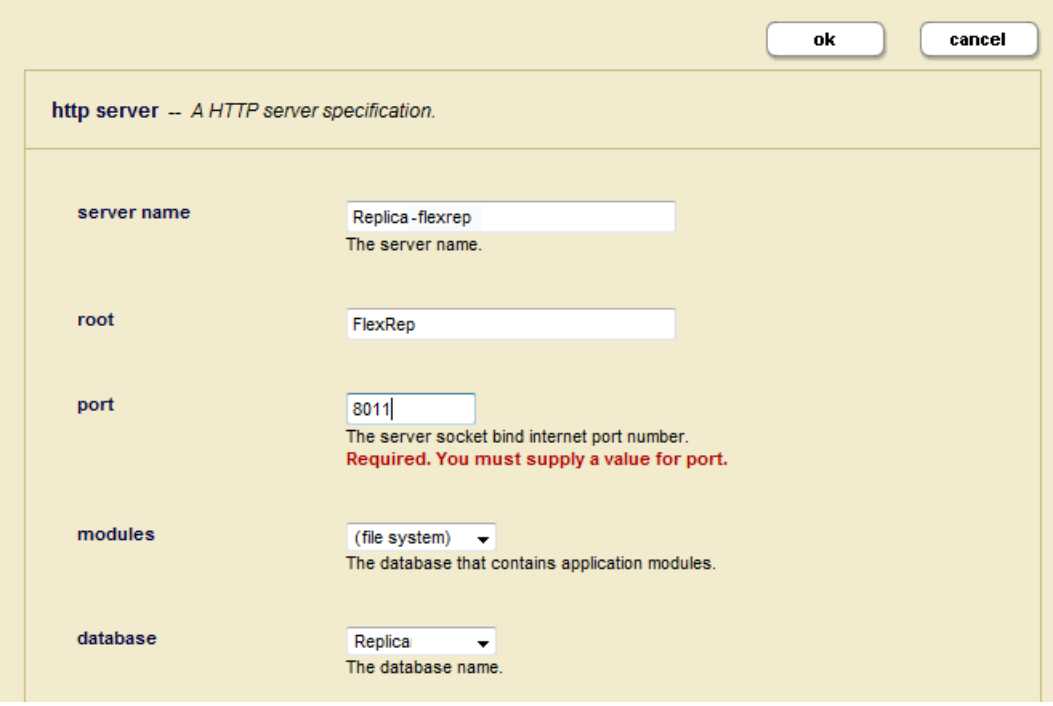
In a push replication configuration, the Replica database must be connected to a Replication App Server. In a pull replication configuration, the Master database must be connected to a Replication App Server.

The following procedure describes how to create an App Server for a Replica in a push replication configuration.

1. Under the Replica database, navigate to Flexible Replication. In the Database Replication page, click Create under Application Server:

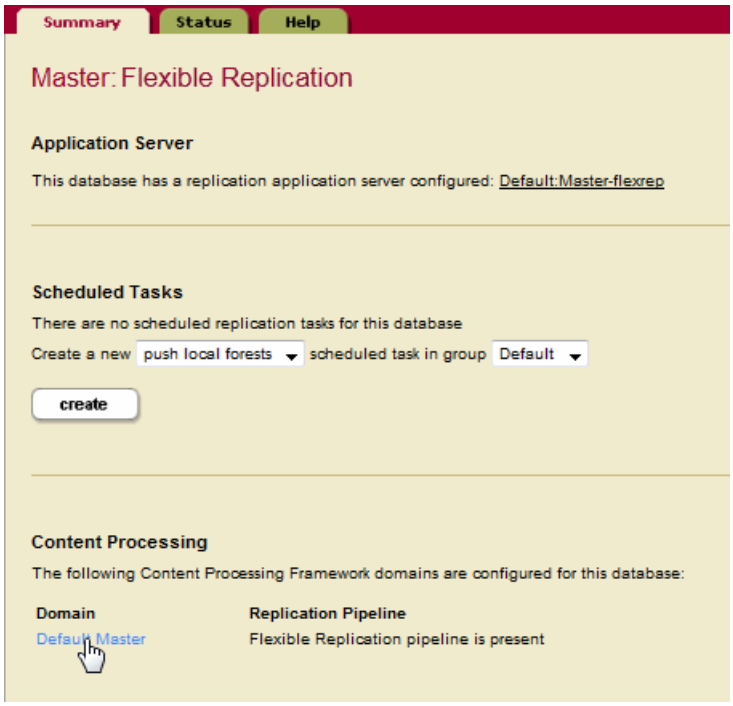


2. In the HTTP Server Configuration page, specify a port number of the App Server. Set the other fields as required, but do not change the server name, root directory, or database settings, which have been prefilled by MarkLogic Server. When finished, click Ok.

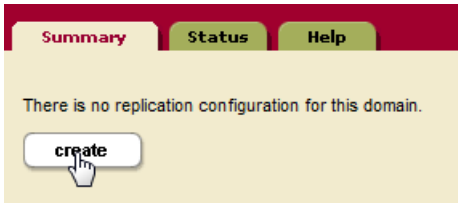


2.4 Configuring Push Replication on the Master Database

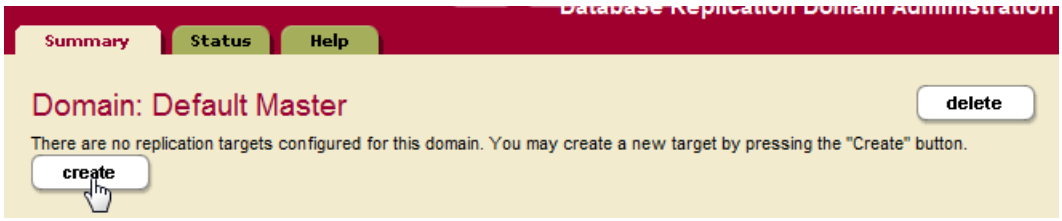
1. For the Master database, navigate to Flexible Replication and click Default Master under Content Processing:



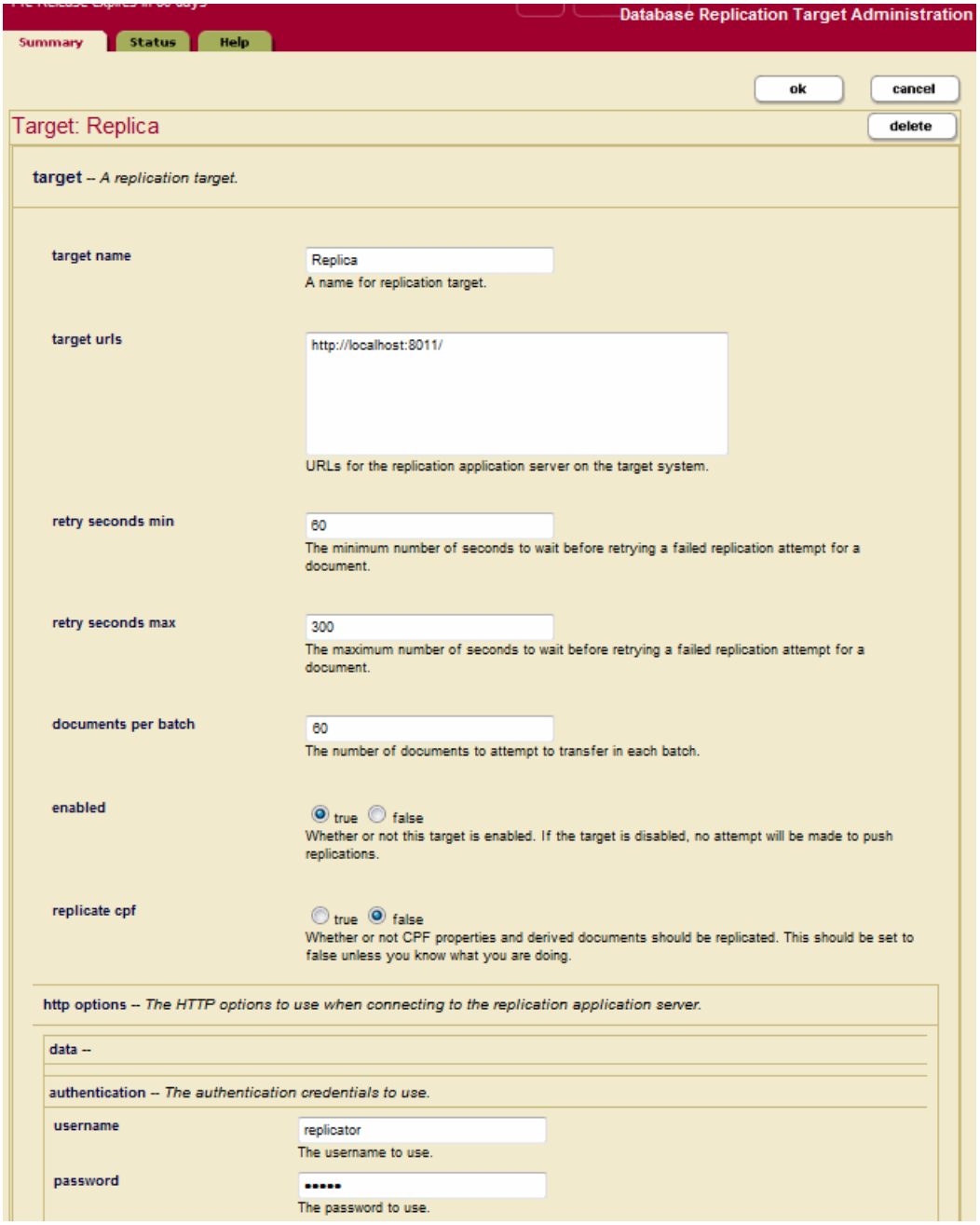
2. Click Create to configure replication for the Default Master domain:



3. Click Create to configure a replication target for the Default Master domain:



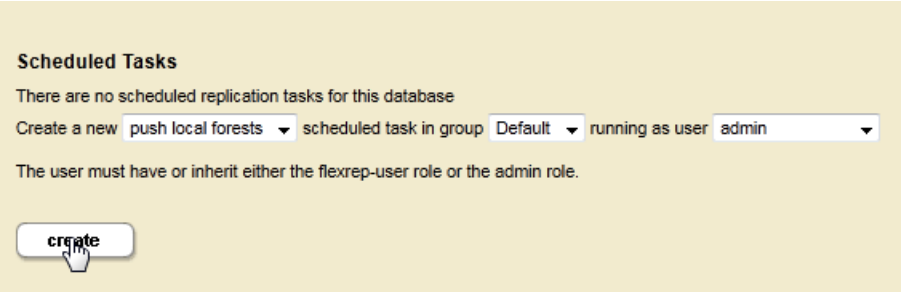
4. In the Database Replication Target page, specify a target name, the target URL that includes the hostname and port number you specified for the Replica's App Server in "Creating a Replication App Server" on page 13. Under Authentication, specify the username and password for a user assigned the `flexrep-user` role on the Replica's App Server:



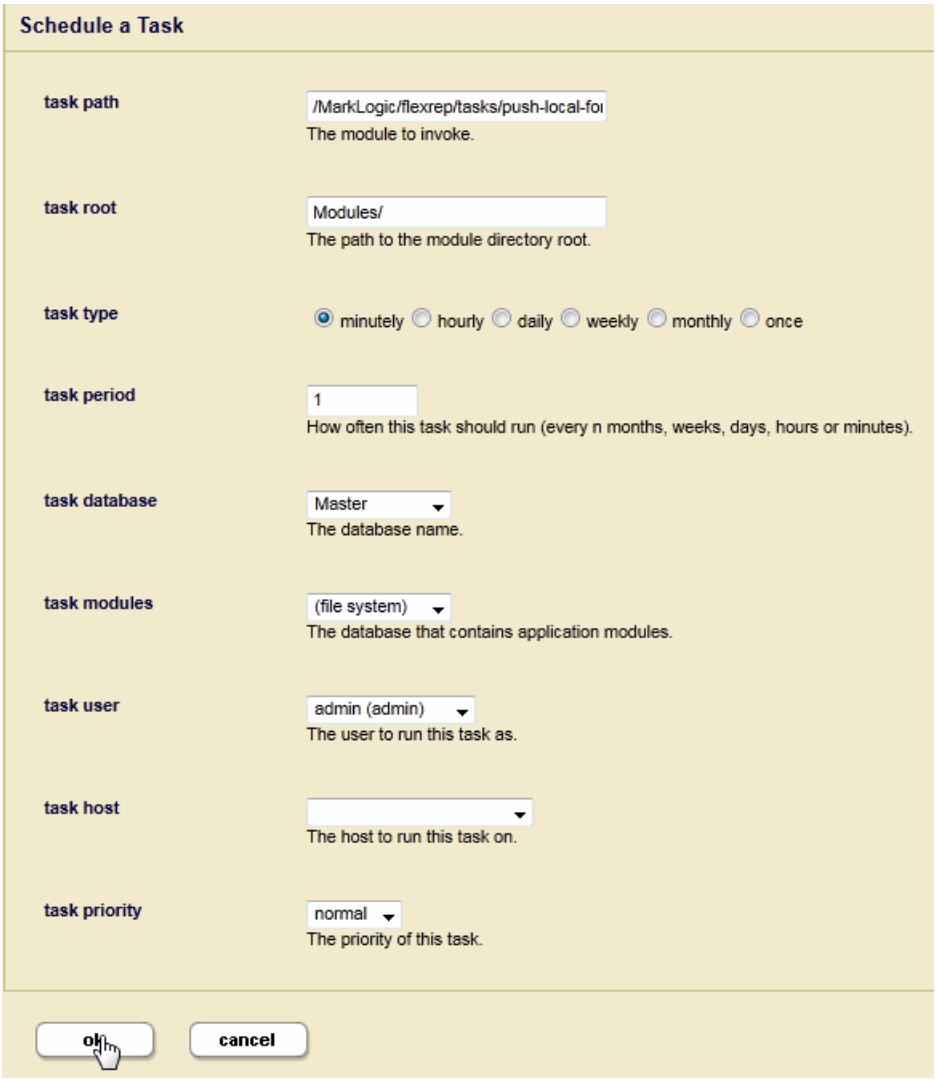
2.5 Scheduling Replication Push Task

You must create a scheduled task to periodically push updated content on the Master to the Replicas. When a document is updated on the Master, it is immediately replicated to the Replica. A scheduled replication task provides a retry mechanism in the event the initial replication fails. A task is also required to replicate deletes and to replicate all of the documents that were on the master before replication was enabled. For more information on scheduled replication tasks, see “Creating a Scheduled Replication Task” on page 29.

1. For the Master database, navigate to Flexible Replication and click Create under Scheduled Tasks:



2. Create a Scheduled Tasks that defines the frequency in which documents are to be replicated from the Master to the Replica. You must also specify the task user. Do not specify a task host. Click OK.



2.6 Loading Documents into the Master Database and Checking Replication

At this point, any documents loaded into the Master database will be replicated to the Replica database. You can use Information Studio to load documents into the Master database. For details on using Information Studio to load documents, see Creating and Configuring Flows in the *Information Studio Developer's Guide*.

> **Note:** Be sure to load your data using the same URI as specified in your replicated domain. For example, if you use the default URI (/), make sure all of your document names start with '/'.

Alternative methods for loading content into a database include:

- Using the XQuery load document functions, as described in Loading Content Using XQuery in the *Loading Content Into MarkLogic Server Guide*.
- Setting up a WebDAV server and client, such as Windows Explorer, to load your documents. See the section Simple Drag-and-Drop Conversion in the *Content Processing Framework Guide* guide for information on how to configure a WebDAV server to work with Windows Explorer.
- Creating an XCC application, as described in Using the Sample Applications in the *XCC Developer's Guide*.

One way to confirm the content has been replicated to the replica is to use the explore feature in Query Console to view the contents of the Replica database. For details on how to use Query Console to explore the contents of a database, see Exploring Your Databases in the *Query Console User's Guide*.

3.0 Configuring Replication

This chapter describes how to configure your MarkLogic Servers for replication. The topics in this chapter assume you are familiar with the replication principles described in "Understanding Flexible Replication" on page 5 and the basic configuration procedures described in the "Flexible Replication Quick Start" on page 9.

This chapter includes the following sections:

- Replication Security
- Defining Replicated Domains
- Configuring Replication App Servers
- Configuring Push Replication
- Creating a Scheduled Replication Task
- Configuring Pull Replication
- Backing Up, Restoring, and Clearing the Master and Replica Databases

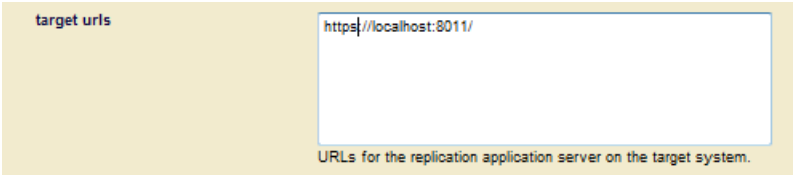
3.1 Replication Security

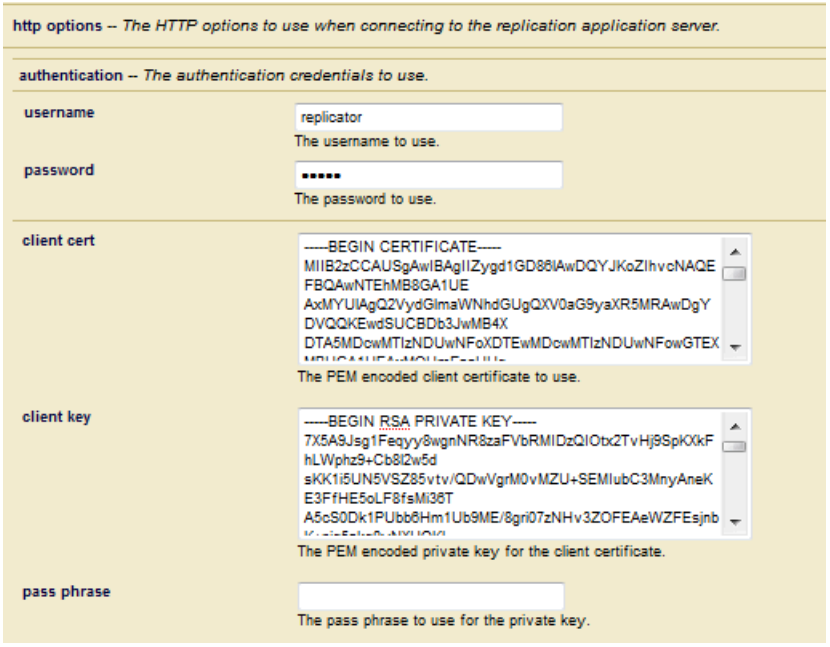
The `flexrep-admin` role is required to configure replication. The user you use to access the Replica App Server when pushing, or into the Master App Server when pulling, requires the `flexrep-user` role. Though you will typically configure replication as the Admin user, you should create a unique replication user to be associated with the replication tasks. The replication user must be given the `flexrep-user` role and have the privileges necessary to update the domain content on both the Master and Replica App Servers.

You can configure SSL on your Master and Replica App Servers to encrypt the replicated data passed between them. For details on configuring SSL on App Server, see the Configuring SSL on App Servers chapter in the *Administrator's Guide*.

If SSL on the Replica App Server is configured to require a client certificate from the Master, the Target URL must start with HTTPS. Paste the PEM-encoded client certificate and client key in the fields located at the bottom of the Database Replication Target Administration page. If the client key is encrypted, you must also specify a pass phase.

The same security configuration applies when configuring the Replica App Server for pull replication, if the Master App Server requires a client certificate from the Replica. For details on how to configure pull replication, see “Creating a Pull Replication Task” on page 35



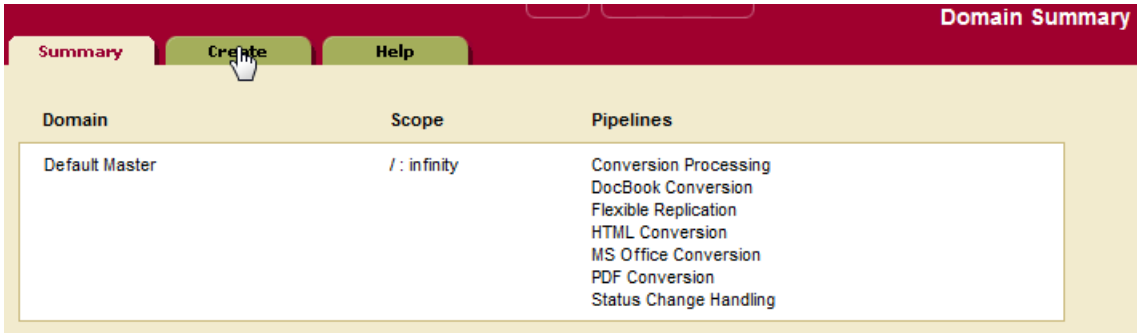


3.2 Defining Replicated Domains

Before you can define replicated domains, you must enable CPF as described in “Configuring a Flexible Replication Pipeline for the Master Database” on page 10.

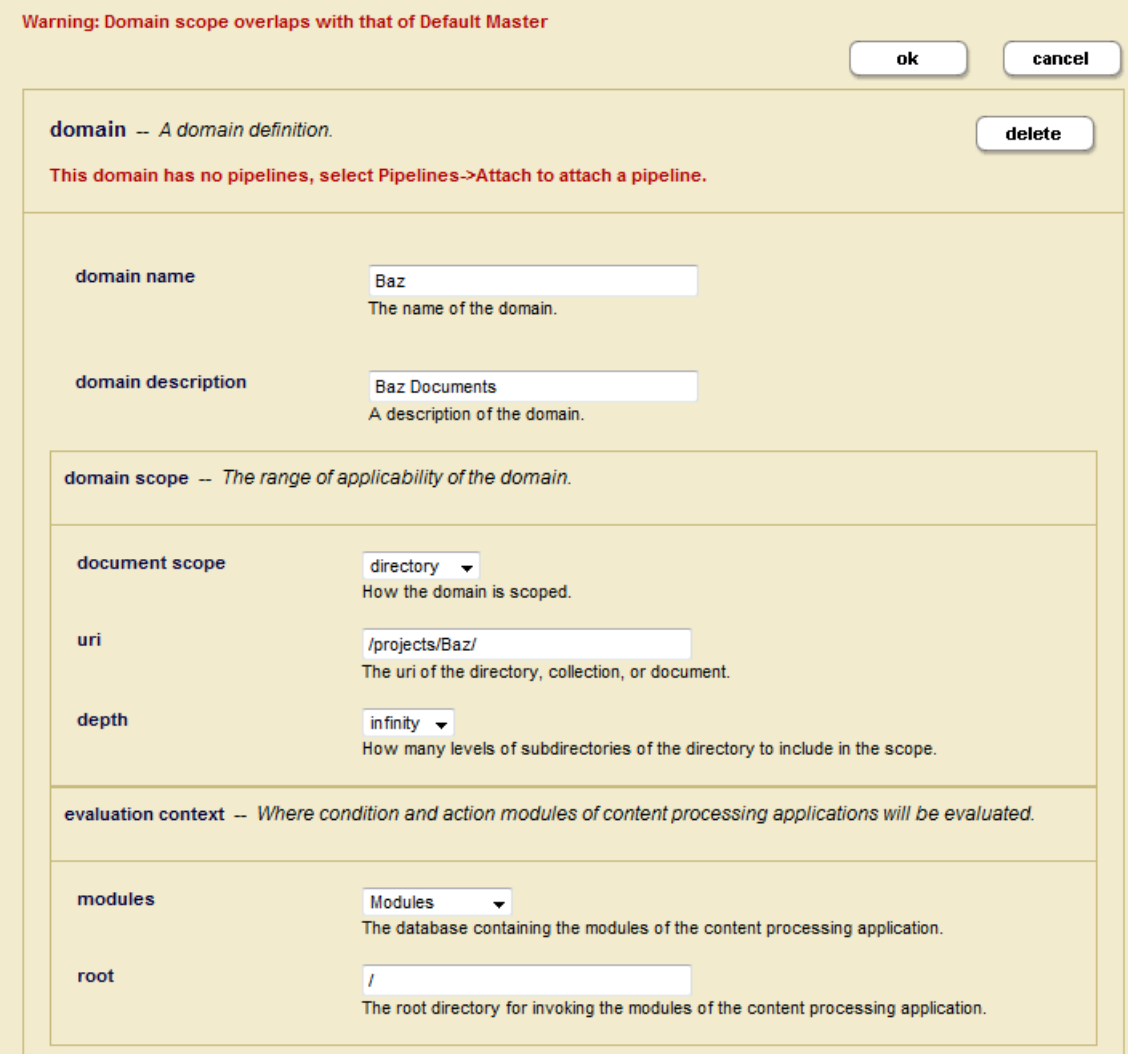
CPF domains are described in detail in the Understanding and Using Domains chapter in the *Content Processing Framework Guide* Guide. The purpose of this section is to describe how to create a domain that defines the scope of the documents in the Master database to be replicated to the Replica database.

Warning Each domain must contain a unique set of documents. No single document can be in more than one domain.



The following procedure describes how to create a new domain for defining the scope of the replicated documents. By default, CPF creates a Default Master domain to replicate from the "/" root directory of the Master database. The "/" root means that the document URIs must be preceded by a "/", such as `/foo.xml` or `/content/foo.xml`. You can also set the Document Scope of the domain to replicate a user collection or an individual document.

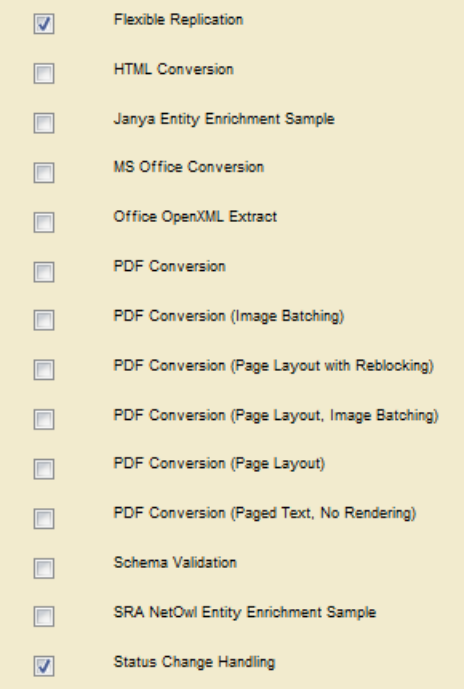
1. To replicate a portion of a database, you can create another domain. For example, you can create a domain that specifies the documents in the `/projects/Baz/` directory on the Master database, as shown below.



You may receive a warning that this domain overlaps with the Default Master domain. This means that the both the Baz domain and the Default Master domain are configured to replicate the documents in the `/projects/Baz/` directory. You cannot have any document that is in more than one domain, so you must either delete or modify the Default Master to resolve this conflict.

Note: If you want to create a domain to replicate the entire database, you can assign a default collection to all of your users and then set the Document Scope to Collections and the URI to the name of the default collection. For details on how to establish a default collection for a user, see Creating a User in the *Administrator's Guide*.

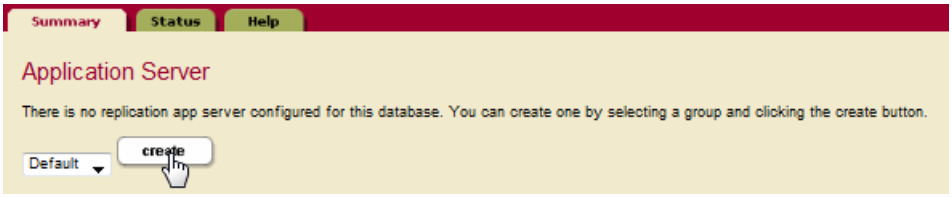
2. Select Pipelines in the navigation frame and select the Flexible Replication and Status Change Handling pipelines. These are the minimum pipelines required for replication. You can select other pipelines, as required for your configuration.



3.3 Configuring Replication App Servers

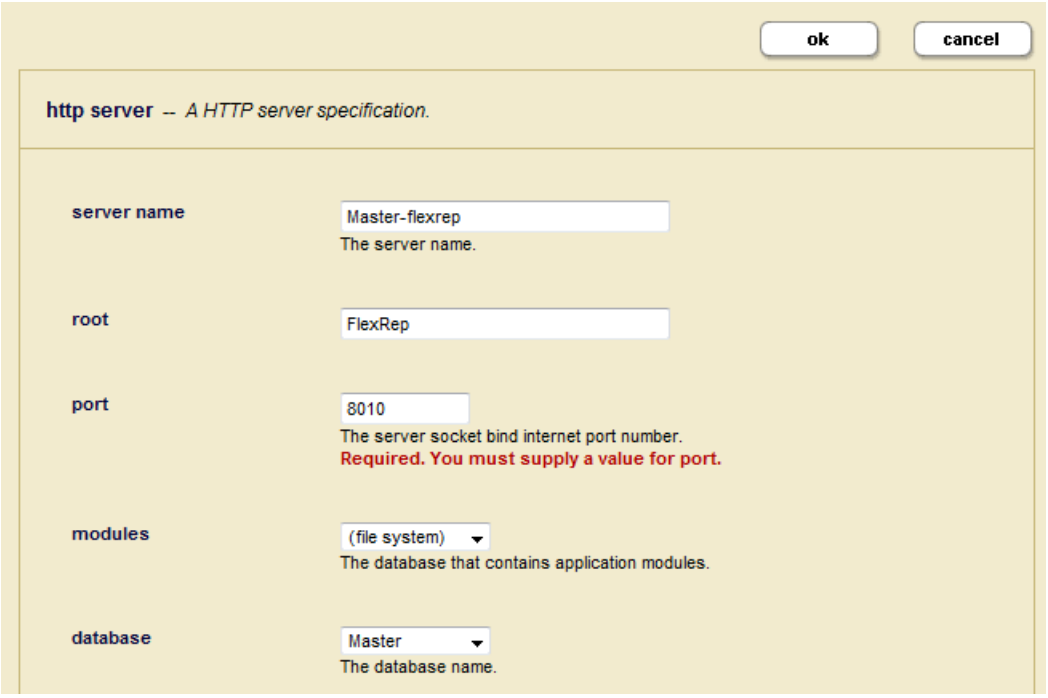
As described in “Creating a Replication App Server” on page 13, in a push replication configuration, the Replica database must be connected to a Replication App Server. In a pull replication configuration, the Master database must be connected to a Replication App Server. This section goes into more detail on creating and configuring Replication App Servers.

1. Create a Replication App Server by clicking Create in the Application Server section of the Database Replication Administration page:



Note: The Server Name, Root, and Database fields are preconfigured and should not be changed.

2. Enter a unique port number for the App Server:



The purpose of the other fields in the HTTP Server Configuration page are described in Creating a New HTTP Server in the *Administrator’s Guide*. If you are going to configure SSL on the App Server to require a client certificate, you must specify the certificate in the

Database Replication Target Administration configuration page, as described in “Replication Security” on page 20.

For example, if you are configuring the Master database to push updates to a Replica App Server that requires a client certificate, you must include a client certificate in the Database Replication Target Administration configuration page for the Master database. If you are configuring a Replica App Server to pull updates from the Master database that requires a client certificate, you must you must include a client certificate in the Replication Pull configuration page for the Replica database. For details on providing a client certificate for either a push or pull replication configuration, see “Replication Security” on page 20.

3.4 Configuring Push Replication

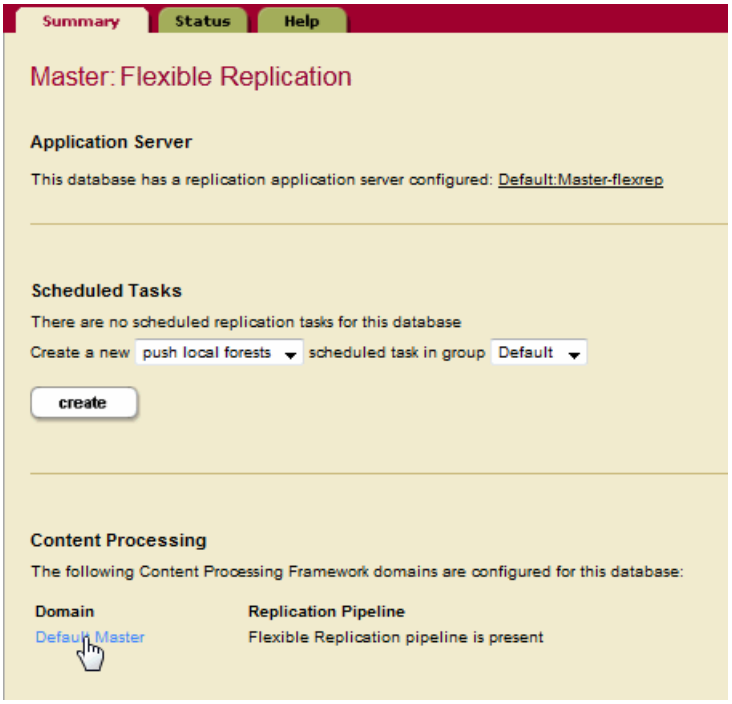
This section describes how to configure your Master database for Push Replication.

Before you can configure a replication target, you must have content processing enabled for the Master database. After configuring push replication, you must create a scheduled task, as described in “Creating a Push-Local-Forests Replication Task” on page 33.

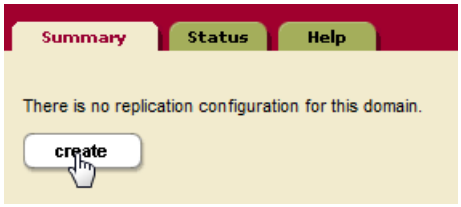
1. Under Databases, click the Master database.

2. Click Flexible Replication under the Master database.

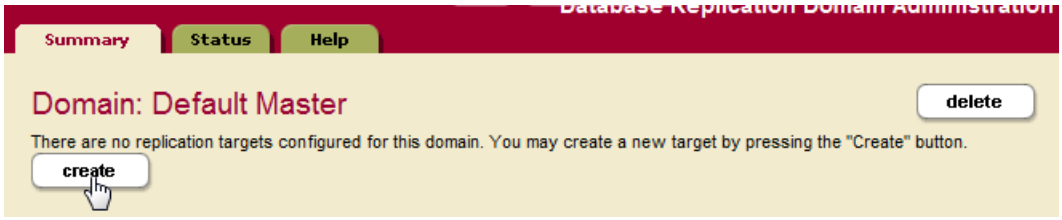
3. Locate the Content Processing section on the Database Replication Administration page and click on the domain name.



4. Click Create to configure replication for the domain:



5. Click Create to configure a replication target for the domain:



6. In the Database Replication Target Administration page, specify a target name and one or more target URLs that include the hostname and port number you specified for your Replica App Servers in “Creating a Replication App Server” on page 13. The Retry Seconds Min/Max and Documents Per Batch setting are for scheduled tasks, as described in “Creating a Scheduled Replication Task” on page 29.

   Under Authentication, specify the username and password for a user assigned the `flexrep-user` role. This user must have the same username/password as the user on the Master App Server who creates and updates the documents. This user needs to have sufficient permissions on the target system to insert and update the replicated documents. Because Flexible Replication passes through the permissions set on the Master App Server when replicating, the user on the target must have permissions to update the document later on. Permissions can either be configured by the administrator on both the Master and Replica App Server, or you can use a filter to adjust the document permissions so the target user can later update the document, as described in “Adding Document Permissions” on page 48.

You can optionally disable pushing replication to the target by setting `Enabled` to `false`. Do not set the `Replicate CPF` option to without the direction from MarkLogic Customer Support.



7. If the Replica App Server requires a client certificate, the Target URL must start with HTTPS. Paste the client certificate, client key and pass phrase (if the client key is encrypted) in the fields located at the bottom of the Database Replication Target Administration page. For details on how to configure the Master to provide a client certificate, see “Replication Security” on page 20.

3.5 Creating a Scheduled Replication Task

Regardless of whether you configure replication as push or pull, you must create a scheduled task to periodically replicate updated content on the Master to the Replicas. A scheduled replication task does the following:

- Moves existing content that was on the Master before replication was configured (zero-day replication).
- Provides a retry mechanism in the event the initial replication fails.
- Replicates deletes on the Master to the Replica.
- Provides the ability to pull data from a Master server located outside a firewall.

There are three types of scheduled tasks, as described in the table below.

Task	Description
push-local-forests	The push-local-forests task setting is the preferred method for retry and zero-day replication. Once the initial scheduled interval is reached, this task pushes each batch of documents, immediately following the previous batch, regardless of the scheduled task frequency. If you had previously used the push task, switching over to the push-local-forests setting will provide better performance. **Note:** The push-local-forests task for each forest can only be respawned a maximum of 500 times per replication period. in order to avoid overflowing the task limit. After 500 respawns, the task will wait for the next replication period to start replicating again. For each respawn, the task will replicate the batch size to the replica. Do not specify a task host for the push-local-forests task. This allows the task to run on all hosts in the cluster.
push	The push task pushes a batch of documents from all of the forests in the database. This task pushes one batch from the entire database per scheduled interval. This form of push replication is available for users who configured replication on earlier versions of MarkLogic Server. If you are currently configured for this form of push replication, you will obtain better performance if you switch to push-local-forests replication. The push task runs on a single host, which you must specify as the task host.

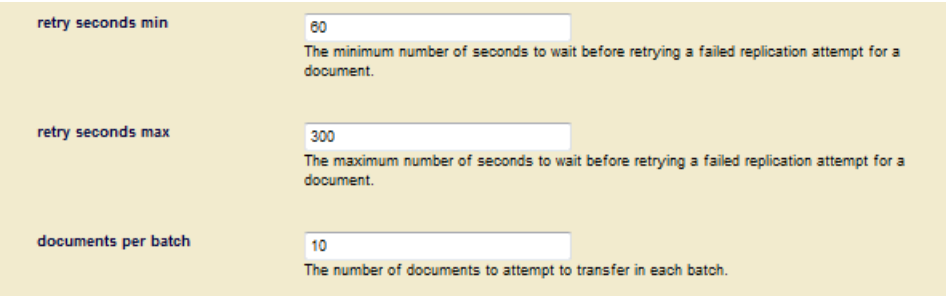
Task	Description
pull	The pull task is configured on a Replica database to pull data from the Master database. This is the only replication option available to a Replica, so the pull scheduled task must serve as the initial replication mechanism in addition to retry and zero-day replication. The pull task runs on a single host, which you must specify as the task host.

For all scheduled replication tasks, the replication retry configuration is a combination of the scheduled task frequency settings and the target settings, Retry Seconds Min, Retry Seconds Max, and Document Per Batch.

The task type in the Scheduled Task Configuration page indicates the interval at which each scheduled replication task is to run. In most configurations, the task type will be minutely. For details on configuring scheduled tasks, see Scheduling Tasks in the *Administrator's Guide*.



The Retry Seconds Min/Max settings in the Database Replication Target Administration page indicate the minimum and maximum number of seconds any documents that failed to replicate are eligible for replication retries. In a push configuration, the Documents Per Batch setting specifies how many documents that have failed to replicate are to be retried during each scheduled task interval. In a pull configuration, the Documents Per Batch setting specifies the total number of replicated documents (retries of failed documents and newly added or updated documents) that are to be pulled from the Master during each scheduled task interval.



For example, the retry minimum is 30 seconds, the retry maximum is 300 seconds (five minutes) and the scheduled replication task period is every one minute. If a document fails replication, it is eligible to be retried in 30 seconds. This means that MarkLogic Server will attempt to replicate the document at the next minute interval. The retry interval is doubled each time the document fails to replicate until the interval reaches the maximum retry setting, at which time the retry interval remains at the maximum. So, if the document in our example fails a second time, it will be eligible for retry in one minute. Should the document fail to replicate the second time, the retry interval is set to two minutes, and so on until the interval reaches the five-minute retry maximum setting, after which MarkLogic Server tries to replicate the document every five minutes.

The Documents Per Batch setting also plays a role with replication retries. For example, if the batch value is one and five documents have failed to replicate, then MarkLogic Server will only attempt to retry one failed document that is eligible for replication retries during each scheduled task interval. The document selected for replication retry is the earliest eligible document. Because of this, zero-day documents take precedence over documents that have failed replication.

Though the Documents Per Batch setting is 1 by default, a more typical value is in the range of 10 - 100. If there is a large number of documents to be replicated in a zero-day fashion, you can maximize the load speed by closely matching the scheduled interval with the Documents Per Batch setting, so that the maximum number of documents are loaded within the time interval. For example, if your scheduled interval is one minute and you have determined that you can replicate a maximum of 250 documents per minute, then the optimum Documents Per Batch setting would be 250.

When a flexible replication domain includes multiple targets, the scheduled task will use the smallest of the configured Documents per Batch settings. So if one target has Documents per Batch set to 1, and another has Documents per Batch set to 100, the scheduled task will use a batch size of 1.

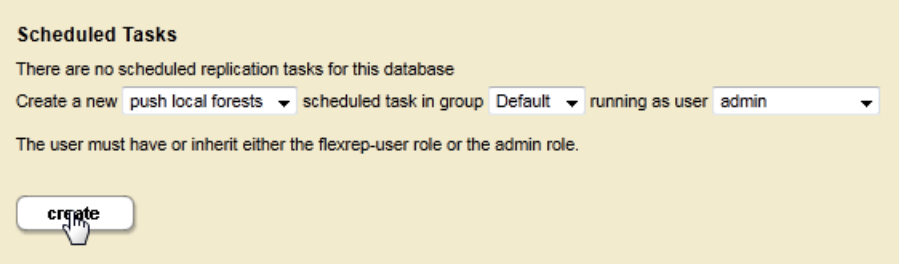
3.5.1 Creating a Push-Local-Forests Replication Task

This section describes how to configure a push-local-forests replication task to provide the Master with the means to replicate deletes to the Replica, retry replication on documents that have failed the initial attempt to replicate, and to replicate documents that were in the replicated domain before replication was configured.

> **Note:** A scheduled push-local-forests task should be run by a user assigned the `flexrep-admin` role or the `admin` role.

Do the following to create a push-local-forests scheduled replication task:

1. For the Master database, navigate to Flexible Replication, select push local forests in the drop-down menu, select the group in which you want to schedule the task, and click Create under Scheduled Tasks:



2. Create a Scheduled Tasks that defines the frequency in which documents are to be replicated from the Master to the Replica (when using push-local-forests, this interval only defines the timeframe to wait before pushing the initial batch). You must also specify the task user.

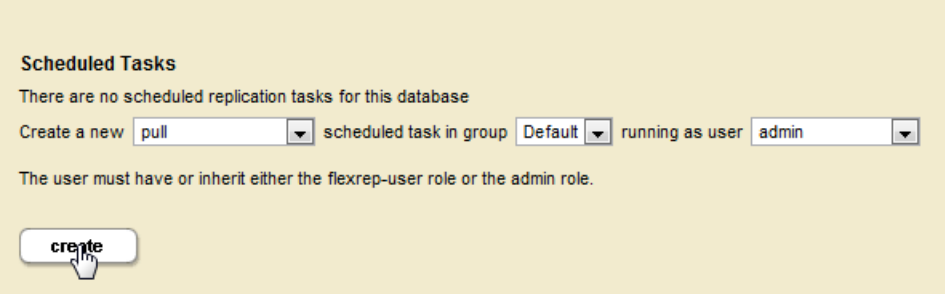
   Note: Do not specify a host if you are creating a push-local-forests scheduled task.



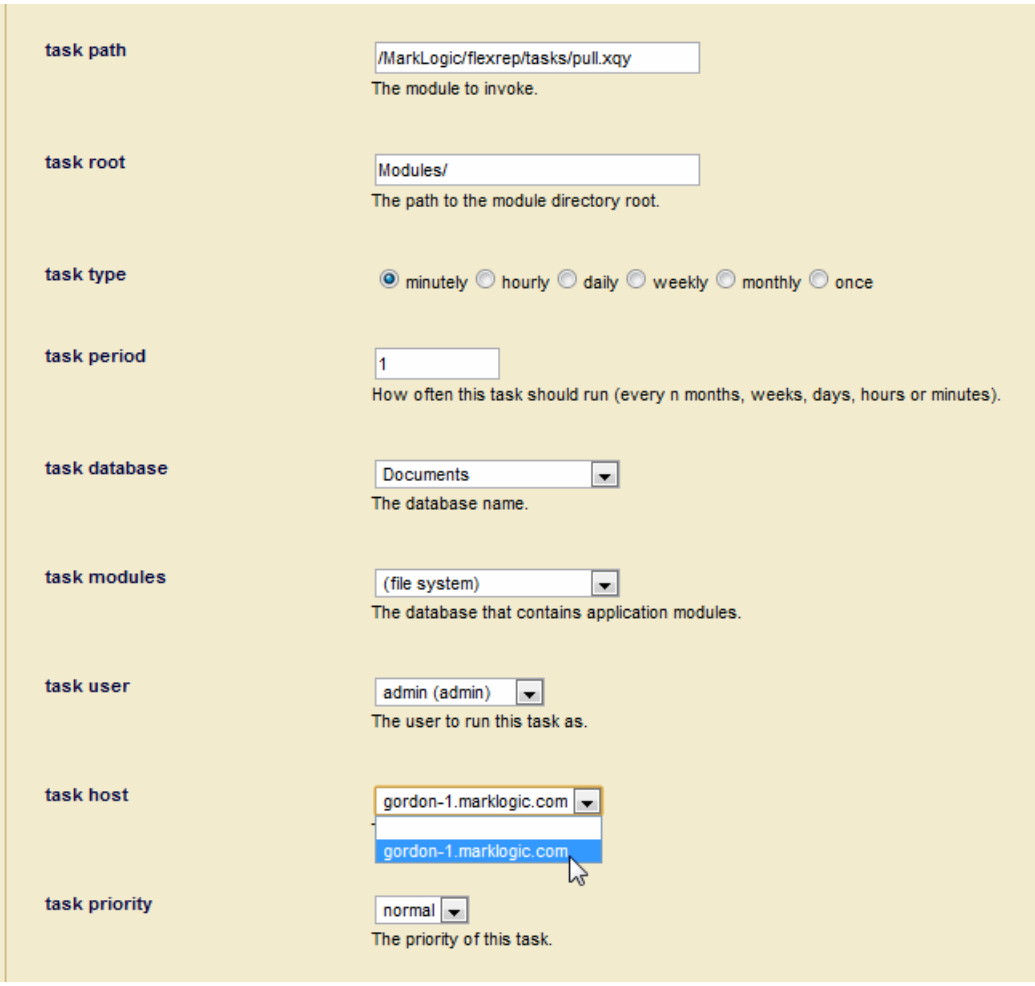
3.5.2 Creating a Pull Replication Task

In a pull replication configuration, the Replicas pull updates from the Master. Unlike push replication, which replicates as soon as the documents on the Master are updated, the only way to configure pull replication is by means of a scheduled task.

1. On the Replica, create a scheduled task to pull the content from the Master:



2. Specify the pull frequency. In the task user pull-down menu, select the user with the `flexrep-user` role on a Replica host. Under task host, select a Replica host or leave the field empty so the task can run on any host in the cluster. Click OK.



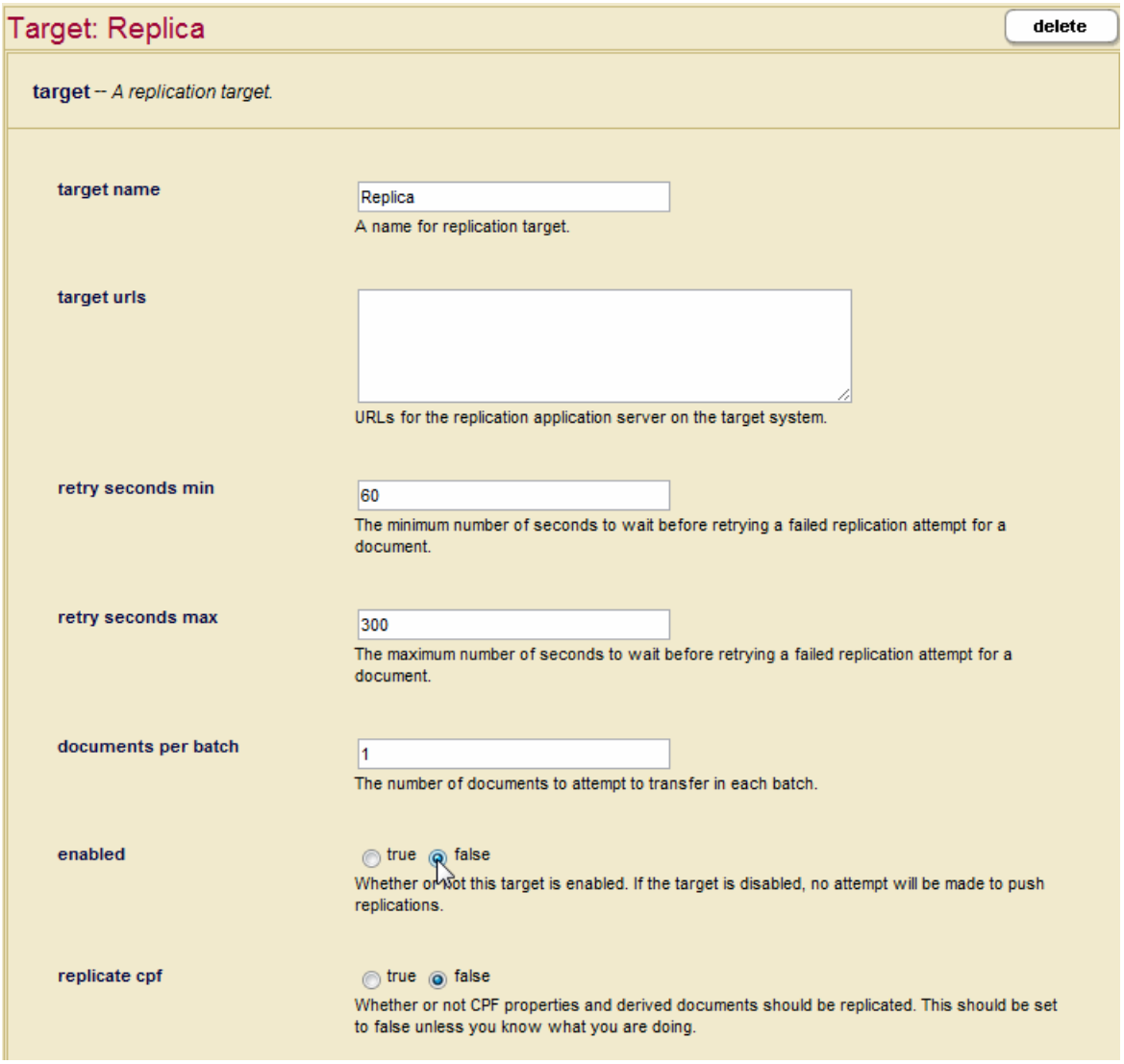
3.6 Configuring Pull Replication

Pull Replication is useful when a Replica is behind a firewall that only allows its internal servers to pull from a Master server outside the firewall. This section describes how to configure your Replica database for Pull Replication.

Note: If the Replica is to act as a Master for another Replica, then you must have CPF enabled on the Replica, as described in "Configuring a Flexible Replication Pipeline for the Master Database" on page 10. However, if the Replica has CPF installed and it is not to act as a Master, then you must disable the Flexible Replication pipeline on the Replica.

The procedure for configuring Pull Replication on the Master and Replica databases is as follows:

1. On a Master host, define a replication target for the Master Content Processing Domain with push replication disabled. You can do this by either not defining a target URL in the target configuration page or by disabling the replication target, or by doing both as shown below.

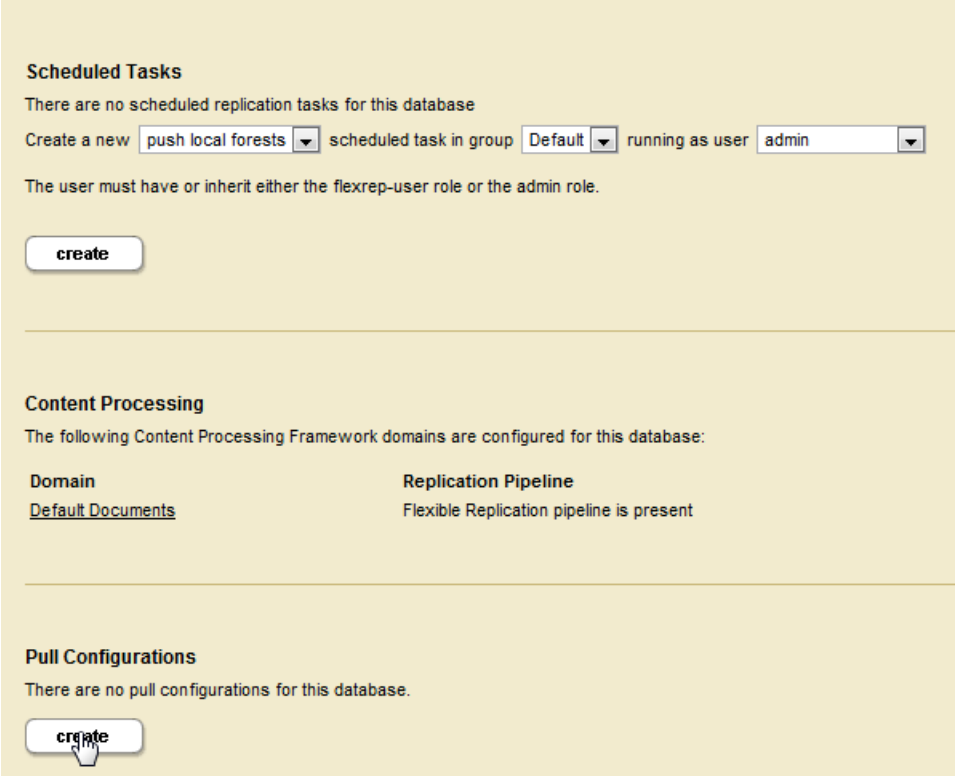


2. Under Databases, click the Replica database.

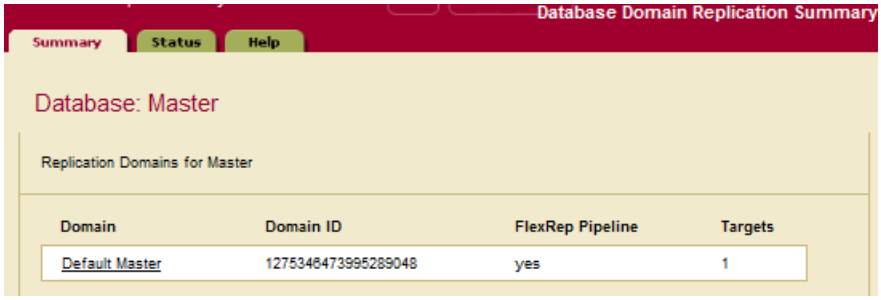
3. Click Flexible Replication under the Replica database.

4. Create a Pull Replication Task, as described in “Creating a Pull Replication Task” on page 35.

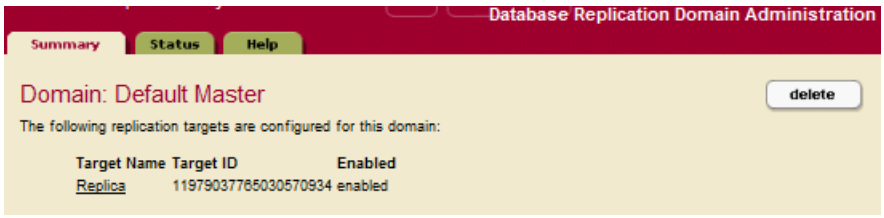
5. Under Pull Configuration, click Create:



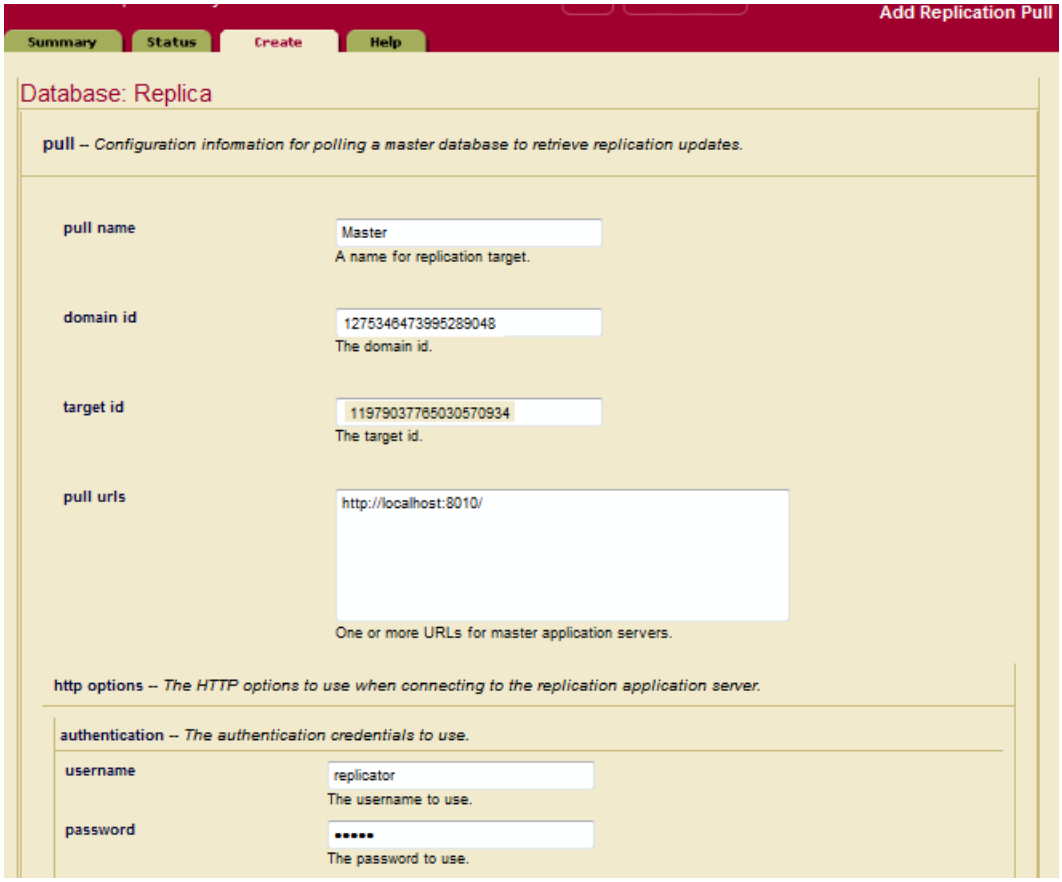
6. On a Master host, obtain the Master domain id from the Database Domain Replication Summary page:



7. On a Master host, obtain the Replica target id from the Database Domain Replication Administration page. If a Replica target does not exist, create one, as described in “Configuring Push Replication” on page 26, Step 5.



8. In the Replication Pull Administration page on the Replica host, enter a pull name, the domain id of the Master, the target id of the Replica, and one or more pull URLs for the Master App Servers. Fill in the username and password fields required to access the Master App Servers.



9. You can optionally specify a client certificate, if one is required by the Master App Server. For details on how to configure the Replica to provide a client certificate, see “Replication Security” on page 20.

3.7 Backing Up, Restoring, and Clearing the Master and Replica Databases

The configuration for flexible replication is stored in the Master and Replica databases. Clearing a Master database has the effect of deleting the target configurations, as well as all document-level information about target replication status. When you backup a Master database, you are also saving the replication configuration information. Should you restore that database to another cluster, the flexible replication configuration will also be present for the database in that cluster. Should you restore a database that is not configured for replication to one that was previously configured for replication, the replication configuration will be lost.

Before clearing or restoring a Master database, you can save the replication configuration by calling the `flexrep:configuration-get` function. Once the Master is cleared or restored, the replication configuration can be restored using the `flexrep:configuration-insert` function. Alternatively you can recreate the replication configuration programmatically, as described in Scripting Flexible Replication Configuration in the *Scripting Administrative Tasks Guide*.

Clearing a Replica database impacts the Master in that it no longer understands the status of the target. The same issues related to backing up and restoring the Master database described above also apply to backing up and restoring the Replica database. If the target status is lost on the Replica, new updates will replicate to the target, but the Master will not know that the older, unchanged documents are missing. Before clearing or restoring a Replica database, disable replication. After clearing or restoring the Replica database, delete the existing target on the Master, create a new target with the same configuration as the old one, and re-enable replication. The scheduled task will gradually populate the new target's database with all the documents in the domain.

If you have pull replication configured on the Replica, then you can save the configuration by calling the `flexrep:pull-get` function before clearing the database. After clearing or restoring the Replica database, you can restore the pull replication configuration by calling the `flexrep:pull-insert` function.

4.0 Filtering Replicated Documents

Documents may be optionally filtered as part of the replication process by configuring a filter module for the target in the master's configuration. You can create a filter module, place it in the modules database for the replicated domain, and specify it at the bottom of the Database Replication Target page.

When a document is replicated, the document node, along with either an update or delete node, are sent by the Master to the Replicas. You can create filters to modify the contents of the document, update or delete node before it is sent to the Replicas.

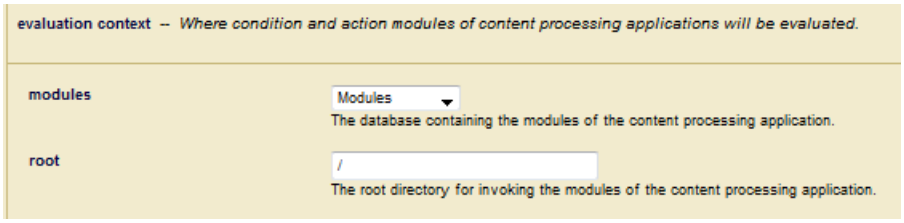
This chapter includes the following sections:

- Configuring MarkLogic Server to use a Replication Filter Module
- Creating a Filter Module
- Example Filters
- Setting Filter Options

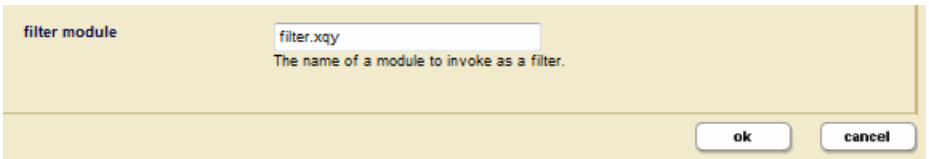
4.1 Configuring MarkLogic Server to use a Replication Filter Module

To configure MarkLogic Server to use a replication filter module, do the following:

1. Navigate to the Domain Definition page for the replicated domain, as described in “Defining Replicated Domains” on page 22. At the bottom of the page, specify the location of your replication filters. The default location for filters is the root directory in the modules database for the replicated domain:



2. Navigate to the Replication Target page, as described in “Configuring Push Replication” on page 26. At the bottom of the page, specify the name of the replication filter module to be used:



4.2 Creating a Filter Module

You can create replication filters to modify the document node, update node, or delete node before it is replicated to the target. If no document node follows the update node in the sequence, the document's root node will be removed on the Replica. If the filter returns an empty sequence, the framework will not replicate the document to the target.

A filter returns both an update and a document node in the case of a document update, or an update node only, in the case of a document delete. If a filter returns multiple update nodes, they will all be applied to the target. This could be used to break a replicated document apart into multiple documents on the target.

A replication filter receives the following external variables as parameters:

```
declare variable $flexrep:uri as xs:string external;
declare variable $flexrep:target as element(flexrep:target) external;
declare variable $flexrep:update as element() external;
declare variable $flexrep:doc as document-node()? external;
```

The `$flexrep:target` variable contains the replication target configuration and the `$flexrep:doc` variable identifies the replicated document node. The `$flexrep:update` variable will be either a `flexrep:update` or a `flexrep:delete` node. Following is an example of each to illustrate the contained information (in the flexrep namespace).

A `flexrep:update` node looks like:

```
<flexrep:update
  xmlns:flexrep="http://marklogic.com/xdmp/flexible-replication">
    <doc:uri xmlns:doc="xdmp:document-load">
       /content/myDoc.xml
    </doc:uri>
    <flexrep:last-updated>
2010-09-29T14:08:28.391-07:00
    </flexrep:last-updated>
    <doc:format xmlns:doc="xdmp:document-load">xml</doc:format>
    <flexrep:permissions>
      <flexrep:permission>
        <sec:role-name xmlns:sec="http://marklogic.com/xdmp/security">
            admin
        </sec:role-name>
        <sec:capability
          xmlns:sec="http://marklogic.com/xdmp/security">
            read
        </sec:capability>
      </flexrep:permission>
      <flexrep:permission>
        <sec:role-name xmlns:sec="http://marklogic.com/xdmp/security">
            admin
        </sec:role-name>
        <sec:capability
          xmlns:sec="http://marklogic.com/xdmp/security">
```

```
                update
            </sec:capability>
          </flexrep:permission>
        </flexrep:permissions>
        <doc:collections xmlns:doc="xdmp:document-load"/>
        <doc:quality xmlns:doc="xdmp:document-load">0</doc:quality>
        <flexrep:forests/>
        <prop:properties xmlns:prop="http://marklogic.com/xdmp/property"/>
    </flexrep:update>
```

A `flexrep:delete` node looks like:

```
    <flexrep:delete xmlns:flexrep=
      "http://marklogic.com/xdmp/flexible-replication">
        <doc:uri xmlns:doc="xdmp:document-load">
           /content/myDoc.xml
        </doc:uri>
        <flexrep:last-updated>
           2010-03-04T14:35:12.714-08:00
        </flexrep:last-updated>
    </flexrep:delete>
```

4.3 Example Filters

This section shows the following filter examples:

- Adding a Collection
- Changing the Document Quality
- Adding Document Permissions
- Adding a Forest Name
- Changing the Document URI
- Changing a Document Element
- Prohibiting Replication on Select Documents

The first example, Adding a Collection, demonstrates how to use either an XQuery function or an XSL stylesheet to produce the same results. The remaining examples make use of XQuery functions, only.

> **Note:** Filter modules must be in the modules database for the replicated domain. You can either use `xdmp:document-insert` to insert the module into the modules database or specify '(file system)' for the modules database and place the module in the `/MarkLogic/Modules` directory.

4.3.1 Adding a Collection

This section shows two filters that add a collection to replicated documents. The first example is a filter that makes use of an XQuery function. The second is a filter that makes use of an XSL stylesheet. Both filters produce the same results.

The following filter defines an XQuery function that iterates through the elements of the update node, locates the `doc:collections` element, and inserts a `sec:uri` element with the value of `http://marklogic.com/flexrep/collection-two`:

```
xquery version "1.0-ml";

declare namespace flexrep =
  "http://marklogic.com/xdmp/flexible-replication";

declare namespace doc = "xdmp:document-load";

declare variable $flexrep:uri as xs:string external;
declare variable $flexrep:target as element(flexrep:target) external;
declare variable $flexrep:update as element() external;
declare variable $flexrep:doc as document-node()? external;

declare function local:add-my-collection(
  $update as element(flexrep:update))
  {
    element flexrep:update {
      $update/@*,
      for $n in $update/node()
      return
        typeswitch($n)
        case element(doc:collections)
          return element doc:collections {
            $n//sec:uri,
            element sec:uri
               { "http://marklogic.com/flexrep/collection-two" }
          }
        default return $n
    }
  };

(
  xdmp:log(fn:concat("Filtering ", $flexrep:uri)),
  typeswitch($flexrep:update)
  case element(flexrep:update)
    return (local:add-my-collection($flexrep:update), $flexrep:doc)
  case element(flexrep:delete)
    return $flexrep:update
  default
    return fn:error((), "FILTER-UNEXPECTED", ())
)
```

The following filter defines an XSLT stylesheet that creates a copy of the update node, locates the `doc:collections` element, and inserts a `sec:uri` element with the value of `http://marklogic.com/flexrep/collection-two`:

```
xquery version "1.0-ml";

declare namespace flexrep =
"http://marklogic.com/xdmp/flexible-replication";
declare namespace doc = "xdmp:document-load";
declare variable $flexrep:uri as xs:string external;
declare variable $flexrep:target as element(flexrep:target) external;
declare variable $flexrep:update as element() external;
declare variable $flexrep:doc as document-node()? external;

let $stylesheet :=
 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
       xmlns:doc = "xdmp:document-load"
       xmlns:sec="http://marklogic.com/xdmp/security"
       version="2.0">
   <!-- Default recursive copy transform -->
   <xsl:template match="@*|node()">
     <xsl:copy>
       <xsl:apply-templates select="@*|node()"/>
     </xsl:copy>
   </xsl:template>

   <!-- Add my collection to the existing collections  -->
   <xsl:template match="doc:collections">
     <xsl:copy>
       <xsl:apply-templates select="node()"/>
       <sec:uri>http://marklogic.com/flexrep/collection-two</sec:uri>
     </xsl:copy>
   </xsl:template>
 </xsl:stylesheet>

return (
  xdmp:log(fn:concat("Filtering ", $flexrep:uri)),
  typeswitch($flexrep:update)
  case element(flexrep:update)
    return (
       xdmp:xslt-eval($stylesheet, $flexrep:update)/flexrep:update,
       $flexrep:doc)
  case element(flexrep:delete)
    return $flexrep:update
  default
    return fn:error((), "FILTER-UNEXPECTED", ())
)
```

Either of the above filters will convert the update node to:

```
<flexrep:update xmlns:flexrep=
  "http://marklogic.com/xdmp/flexible-replication">
    ........
    <doc:collections xmlns:doc="xdmp:document-load">
       <sec:uri xmlns:sec="http://marklogic.com/xdmp/security">
          http://marklogic.com/flexrep/collection-two
       </sec:uri>
    </doc:collections>
    ........
    <prop:properties xmlns:prop="http://marklogic.com/xdmp/property"/>
</flexrep:update>
```

4.3.2 Changing the Document Quality

The following filter changes the quality of documents to 3. This is done by iterating through the elements of the update node, locating the `doc:document-quality` element, and resetting its value to 3.

```
xquery version "1.0-ml";

declare namespace flexrep =
  "http://marklogic.com/xdmp/flexible-replication";

declare namespace doc = "xdmp:document-load";

declare variable $flexrep:uri as xs:string external;
declare variable $flexrep:target as element(flexrep:target) external;
declare variable $flexrep:update as element() external;
declare variable $flexrep:doc as document-node()? external;

declare function local:change-quality($update as
element(flexrep:update))
{
  element flexrep:update {
    $update/@*,
    for $n in $update/node()
    return
      typeswitch($n)
      case element(doc:quality)
        return element doc:quality   { 3 }
      default return $n
  }
};

(
  xdmp:log(fn:concat("Filtering ", $flexrep:uri)),
  typeswitch($flexrep:update)
  case element(flexrep:update)
    return (local:change-quality($flexrep:update), $flexrep:doc)
  case element(flexrep:delete)
    return $flexrep:update
  default
    return fn:error((), "FILTER-UNEXPECTED", ())
)
```

This will convert the update node to:

```
<flexrep:update xmlns:flexrep=
  "http://marklogic.com/xdmp/flexible-replication">
    ........
    <doc:quality xmlns:doc="xdmp:document-load">3</doc:quality>
    ........
</flexrep:update>
```

4.3.3 Adding Document Permissions

The following filter adds read and update permission for users with the “developer” role to documents. This is done by iterating through the elements of the update node, locating the `doc:permissions` element, and inserting `sec:permission` elements containing `sec:capability` and `sec:role-id` elements that establish read and update permissions for “developer” users.

```
xquery version "1.0-ml";

declare namespace flexrep =
  "http://marklogic.com/xdmp/flexible-replication";

declare namespace doc = "xdmp:document-load";

declare variable $flexrep:uri as xs:string external;
declare variable $flexrep:target as element(flexrep:target) external;
declare variable $flexrep:update as element() external;
declare variable $flexrep:doc as document-node()? external;

declare function local:change-permission(
  $update as element(flexrep:update))
{
  element flexrep:update {
    $update/@*,
    for $n in $update/node()
    return
      typeswitch($n)
      case element(flexrep:permissions)
        return element flexrep:permissions {
          $n/flexrep:permission ,
          element flexrep:permission  {
            element sec:role-name { "developer" },
            element sec:capability  { "read" }
          },
          element flexrep:permission  {
            element sec:role-name { "developer" },
            element sec:capability  { "update" }
          }
        }
      default return $n
  }
};

(
  xdmp:log(fn:concat("Filtering ", $flexrep:uri)),
  typeswitch($flexrep:update)
  case element(flexrep:update)
    return (local:change-permission($flexrep:update), $flexrep:doc)
  case element(flexrep:delete)
    return $flexrep:update
  default
    return fn:error((), "FILTER-UNEXPECTED", ())
)
```

This will convert the update node to:

```
<flexrep:update xmlns:flexrep=
  "http://marklogic.com/xdmp/flexible-replication">
    ........
    <flexrep:permissions>
      <flexrep:permission>
        <sec:role-name xmlns:sec="http://marklogic.com/xdmp/security">
             admin
        </sec:role-name>
        <sec:capability
          xmlns:sec="http://marklogic.com/xdmp/security">
             read
        </sec:capability>
      </flexrep:permission>
      <flexrep:permission>
        <sec:role-name xmlns:sec="http://marklogic.com/xdmp/security">
             admin
        </sec:role-name>
        <sec:capability
          xmlns:sec="http://marklogic.com/xdmp/security">
             update
        </sec:capability>
      </flexrep:permission>
    </flexrep:permissions>
    <flexrep:permissions>
      <flexrep:permission>
        <sec:role-name xmlns:sec="http://marklogic.com/xdmp/security">
             developer
        </sec:role-name>
        <sec:capability
          xmlns:sec="http://marklogic.com/xdmp/security">
             read
        </sec:capability>
      </flexrep:permission>
      <flexrep:permission>
        <sec:role-name xmlns:sec="http://marklogic.com/xdmp/security">
             developer
        </sec:role-name>
        <sec:capability
          xmlns:sec="http://marklogic.com/xdmp/security">
             update
        </sec:capability>
      </flexrep:permission>
    </flexrep:permissions>
    ........
</flexrep:update>
```

4.3.4 Adding a Forest Name

The following filter adds the forest name, `myFavoriteForest`, to the update node. MarkLogic Server maps the forest name to its ID and passes it to the `xdmp:document-insert` function to insert the document into the named forest. If you specify multiple forests, MarkLogic Server will insert the document into one of them. See the documentation for the `xdmp:document-insert` function for more information.

```
xquery version "1.0-ml";

declare namespace flexrep =
  "http://marklogic.com/xdmp/flexible-replication";

declare namespace doc = "xdmp:document-load";

declare variable $flexrep:uri as xs:string external;
declare variable $flexrep:target as element(flexrep:target) external;
declare variable $flexrep:update as element() external;
declare variable $flexrep:doc as document-node()? external;

declare function local:add-forest(
  $update as element(flexrep:update))
  {
    element flexrep:update {
      $update/@*,
      for $n in $update/node()
      return
        typeswitch($n)
        case element(flexrep:forests)
          return element flexrep:forests {
            element flexrep:forest { "myFavoriteForest" }
          }
        default return $n
    }
  };

(
  xdmp:log(fn:concat("Filtering ", $flexrep:uri)),
  typeswitch($flexrep:update)
  case element(flexrep:update)
    return (local:add-forest($flexrep:update), $flexrep:doc)
  case element(flexrep:delete)
    return $flexrep:update
  default
    return fn:error((), "FILTER-UNEXPECTED", ())
)
```

This will convert the update node to:

```
<flexrep:update xmlns:flexrep=
  "http://marklogic.com/xdmp/flexible-replication">
    ........
    <flexrep:forests>
        <flexrep:forest>myFavoriteForest</flexrep:forest>
    </flexrep:forests>
    ........
</flexrep:update>
```

4.3.5 Changing the Document URI

The following filter adds `/replicated/` to the front of each document URI. This is done by iterating through the elements of the update node, locating the `doc:uri` element and adding `/replicated/` to its value.

```
xquery version "1.0-ml";

declare namespace flexrep =
  "http://marklogic.com/xdmp/flexible-replication";

declare namespace doc = "xdmp:document-load";

declare variable $flexrep:uri as xs:string external;
declare variable $flexrep:target as element(flexrep:target) external;
declare variable $flexrep:update as element() external;
declare variable $flexrep:doc as document-node()? external;

declare function local:change-uri($update as element(flexrep:update))
{
  element flexrep:update {
    $update/@*,
    for $n in $update/node()
    return
      typeswitch($n)
      case element(doc:uri)
        return element doc:uri {
          fn:concat("/replicated", $flexrep:uri)
        }
      default return $n
  }
};

(
  xdmp:log(fn:concat("Filtering ", $flexrep:uri)),
  typeswitch($flexrep:update)
  case element(flexrep:update)
     return (local:change-uri($flexrep:update), $flexrep:doc)
  case element(flexrep:delete)
    return $flexrep:update
  default
    return fn:error((), "FILTER-UNEXPECTED", ())
)
```

This will convert the update node to:

```
<flexrep:update xmlns:flexrep=
  "http://marklogic.com/xdmp/flexible-replication">
    <doc:uri xmlns:doc="xdmp:document-load">
       <doc:uri>/replicated//content/foo.xml</doc:uri>
    </doc:uri>
    ........
</flexrep:update>
```

4.3.6 Changing a Document Element

The following filter changes all <PARA> elements in replicated documents to <PARAGRAPH> and leaves all of the other elements in the documents unchanged. This is done by iterating through the elements of the document node, locating each PARA element and converting its value to PARAGRAPH.

```
xquery version "1.0-ml";

declare namespace flexrep =
"http://marklogic.com/xdmp/flexible-replication";
declare namespace doc = "xdmp:document-load";

declare variable $flexrep:uri as xs:string external;
declare variable $flexrep:target as element(flexrep:target) external;
declare variable $flexrep:update as element() external;
declare variable $flexrep:doc as document-node()* external;

declare function local:passthru($x as node()) as node()*
{
  for $z in $x/node() return local:change-element($z)
};

declare function local:change-element($x as node()) as node()*
{
  typeswitch ($x)
    case document-node() return document {local:passthru($x)}
    case text() return $x
    case element (PARA)
       return <PARAGRAPH>{local:passthru($x)}</PARAGRAPH>
    default return element {
       fn:node-name($x)} {$x/@*, local:passthru($x)}
};

(
  xdmp:log(fn:concat("Filtering ", $flexrep:uri)),
  typeswitch($flexrep:update)
  case element(flexrep:update)
    return ($flexrep:update, local:change-element($flexrep:doc) )
  case element(flexrep:delete)
    return $flexrep:update
  default
    return fn:error((), "FILTER-UNEXPECTED", ())
)
```

4.3.7 Prohibiting Replication on Select Documents

Should you want to prohibit replication on certain documents in a replicated domain, you can add a property to the document that flags it as a no-replicate document. You can then write a filter that checks for the property and determines whether or not to replicate the document, depending on the presence or value of the property.

For example, if the `/content` directory is in a replicated domain, but you don't want to replicate the document, `/content/foo.xml`, you can assign the document a `replicate` property with a value of `no`.

```
xquery version "1.0-ml";

declare namespace prop = "http://marklogic.com/xdmp/property";

xdmp:document-add-properties(
    "/content/foo.xml",
    (<prop:replicate>no</prop:replicate>) )
```

You can write a filter that looks for the `replicate` property on each document. If the property is missing or it is some value other than `no`, then the document is replicated. If the property is set on the document and its value is `no`, then the document will not be replicated.

```
xquery version "1.0-ml";

declare namespace flexrep =
  "http://marklogic.com/xdmp/flexible-replication";

declare namespace doc = "xdmp:document-load";

declare variable $flexrep:uri as xs:string external;
declare variable $flexrep:target as element(flexrep:target) external;
declare variable $flexrep:update as element() external;
declare variable $flexrep:doc as document-node()? external;

(
  xdmp:log(fn:concat("Filtering ", $flexrep:uri)),
  typeswitch($flexrep:update)
  case element(flexrep:update)
    return
    if (xdmp:document-properties($flexrep:uri)//prop:replicate = "no")
    then ()
    else ($flexrep:update, $flexrep:doc)
  case element(flexrep:delete)
    return $flexrep:update
  default
    return fn:error((), "FILTER-UNEXPECTED", ())
)
```

4.4 Setting Filter Options

You can use the `flexrep:configuration-target-set-filter-options` function to change the evaluation parameters used to invoke your replication filter. For example, you can specify filter options that determine which user can invoke the filter or on what database the filter is to be invoked. The options specified by the `flexrep:configuration-target-set-filter-options` function are passed to the `xdmp:invoke` function of the filter module, so any of the options you would specify in the `xdmp:eval` function are recognized.

Filter options cannot be set in by the Admin Interface. You must set filter options programmatically using the flexrep API. The flexrep API is described in the Scripting Flexible Replication Configuration chapter in the *Scripting Administrative Tasks Guide* and the reference documentation for each function is in the *MarkLogic XQuery and XSLT Function Reference*.

For example, you can write a module that specifies that the filter can only be invoked by the user John:

```
xquery version "1.0-ml";

import module namespace flexrep =
   "http://marklogic.com/xdmp/flexible-replication"
      at "/MarkLogic/flexrep.xqy";

import module namespace trgr="http://marklogic.com/xdmp/triggers"
   at "/MarkLogic/triggers.xqy";

let $trigger := trgr:get-trigger("cpf:update Replicated Content")

(: Obtain the id of the replicated CPF domain from the
   Triggers database. :)

let $domain := xdmp:eval(
    'xquery version "1.0-ml";
    import module namespace dom = "http://marklogic.com/cpf/domains"
      at "/MarkLogic/cpf/domains.xqy";
    fn:data(dom:get( "Replicated Content" )//dom:domain-id)',
    (),
    <options xmlns="xdmp:eval">
      <database>{xdmp:database("MyTriggers")}</database>
    </options>)

(: Obtain the replication configuration. :)
let $cfg := flexrep:configuration-get($domain, fn:true())

(: Obtain the ID of the replication target. :)
let $target-id := flexrep:configuration-target-get-id($cfg, "Replica")
```

```
(: Define a flexrep:filter-options element. :)
let $filter-opts :=
  <flexrep:filter-options>
    <user-id xmlns="xdmp:eval">{xdmp:user("John")}</user-id>
  </flexrep:filter-options>

(: Set the flexrep:filter-options element. :)
let $cfg :=
  flexrep:configuration-target-set-filter-options(
    $cfg,
    $target-id,
    $filter-opts)

(: Save the new replication configuration. :)
return flexrep:configuration-insert($cfg)
```

5.0 Checking Replication Status

This chapter describes how to check replication status for a Master database, a domain, and a target. This chapter includes the following sections:

- Special Circumstances that Impact Replication
- Checking the Replication Status of a Master Database
- Checking the Replication Status of a Domain
- Checking the Replication Status of a Target

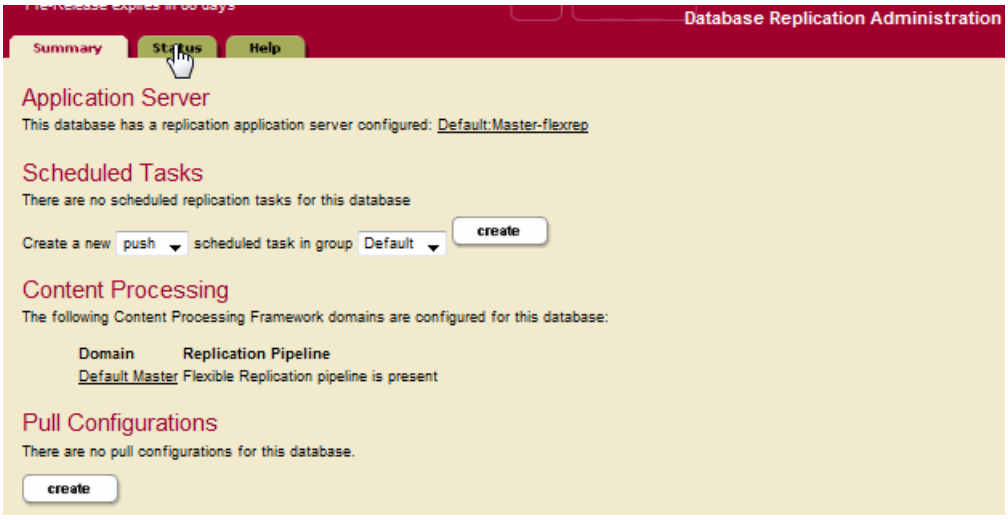
5.1 Special Circumstances that Impact Replication

If the task server queue is more than half full, the Master Server will not push documents to the Replica and will instead leave it for the scheduled push task. This is to help avoid overflowing the task queue when ingesting at a high rate.

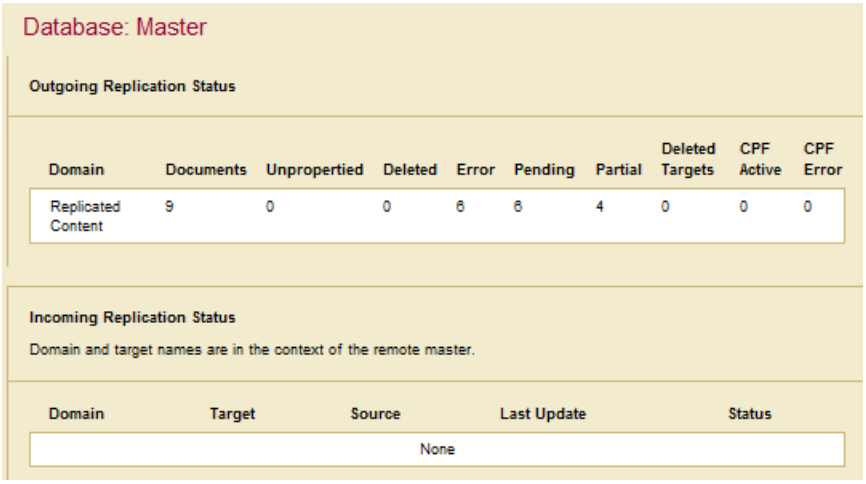
See Examining the Host and Task Server Status Pages For Tasks in the Queue in the *Content Processing Framework Guide* for information on checking the status of the task server.

5.2 Checking the Replication Status of a Master Database

You can check the replication status of a Master database by selecting the Flexible Replication icon under the Master database in the left tree menu and clicking on the Status tab on the Database Replication Administration page.



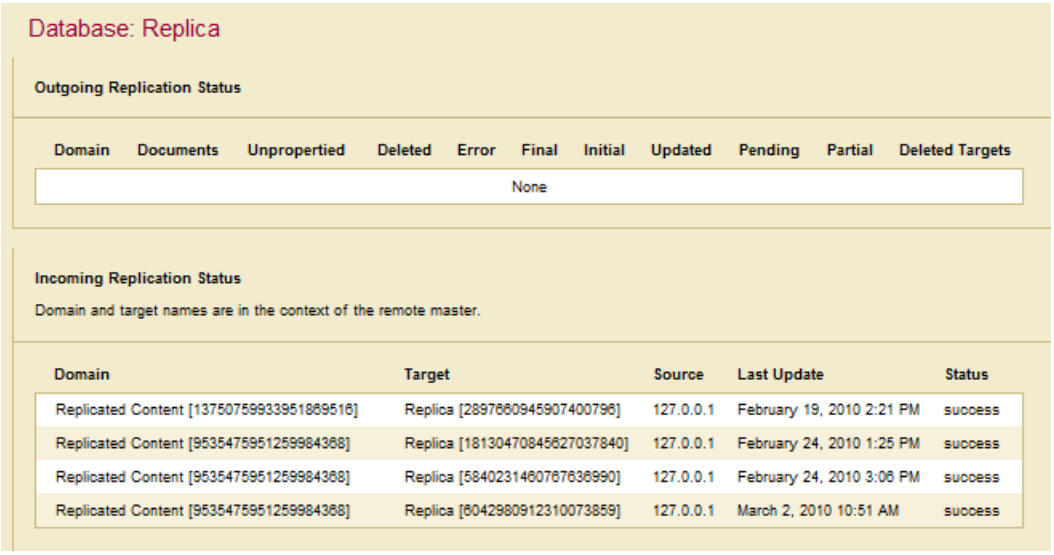
The Outgoing Replication Status indicates the replication status for each replicated domain from the perspective of the Master database.



Field	Description
Domain	The name of the replicated domain.
Documents	The number of documents in the domain on the Master.
Unpropertied	The number of documents in the replicated domain without a replication status. This generally means that these documents existed before the replication domain was configured and have not yet been replicated.
Deleted	The number of replicated documents that were deleted, but the delete has not yet been replicated.
Error	The number of replication errors.
Pending	The number of documents in the replicated domain that have not yet been replicated to the Replicas.
Partial	The number of documents in the replicated domain that have been sent to one or more Replicas, but are pending or have errors for at least one Replica.
Deleted Targets	The number of targets that were deleted from the replication configuration for this domain.

Field	Description
CPF Active	The number of documents that are currently in the CPF pipeline.
CPF Error	The number of documents that encountered an error while in the CPF pipeline.

The Incoming Replication Status indicates the replication status for each replicated domain from the perspective of the Replica database.

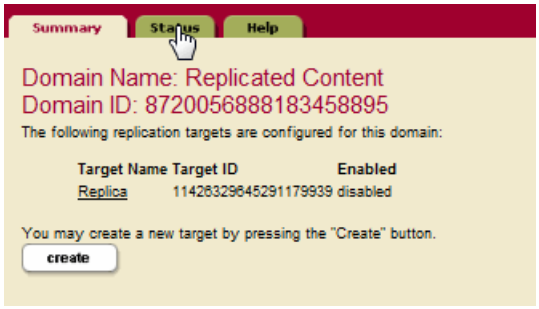


Field	Description
Domain	The name of the replicated domain.
Target	The name and id of the replication target.
Source	The IP address of the Master host server.
Last Update	The date and time of the last replicated update.
Status	The status of the last replicated update

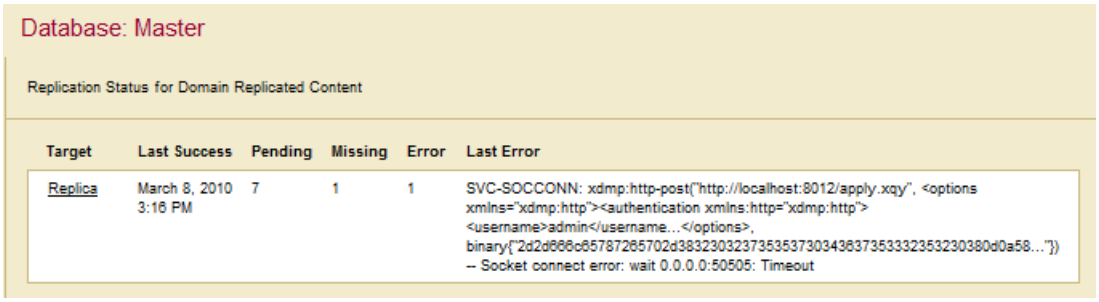
5.3 Checking the Replication Status of a Domain

You can check the replication status of a replicated domain by the following procedure:

1. Select the domain icon under Flexible Replication in the left tree menu and click on the Status tab on the Database Replication Domain Administration page:



2. The page containing the replication status of the domain appears:

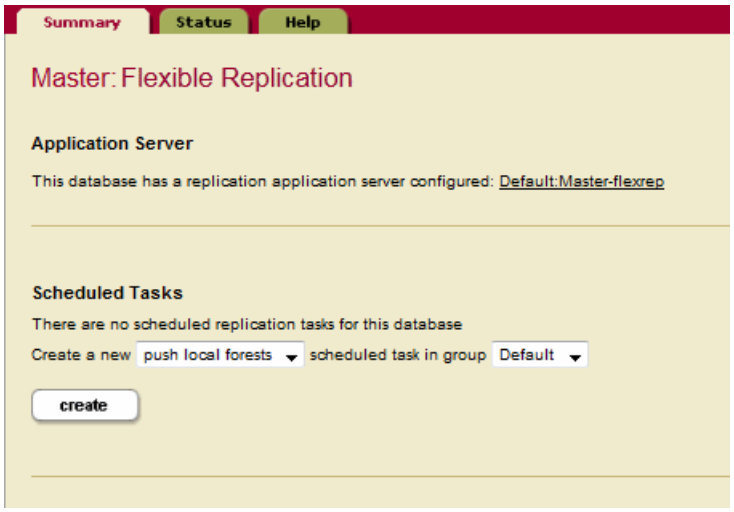


Field	Description
Target	The name of the replication target.
Last Success	The date and time of the last successful update to the target.
Pending	The number of documents in the replicated domain that have not yet been replicated to the target.
Missing	The number of documents missing replication properties. For example, documents that were already in the replicated domain before replication was configured.
Error	The number of replication errors.
Last Error	The last replication error.

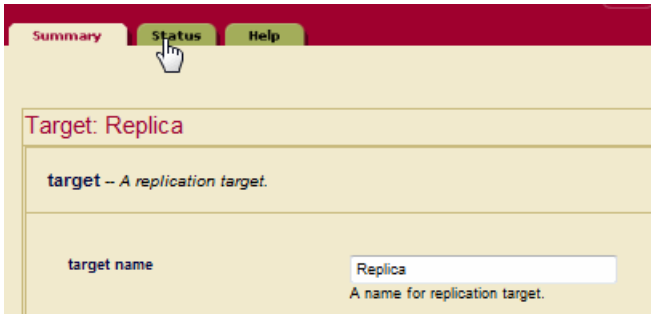
5.4 Checking the Replication Status of a Target

You can check the replication status of a replication target by the following procedure:

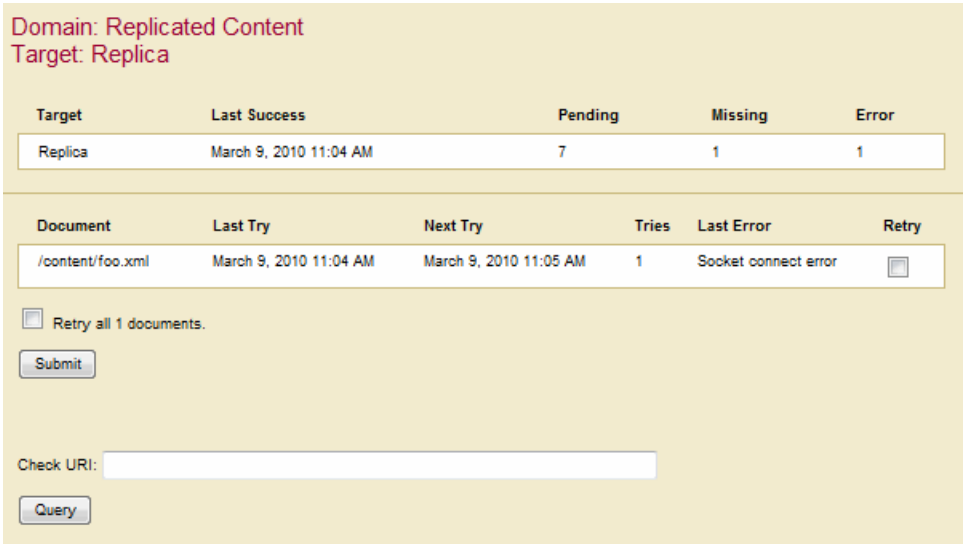
1. Select the domain icon under Flexible Replication in the left tree menu, selecting the target from the and clicking on the Status tab on the Database Replication Target Administration page.



2. Click the Status tab on the Database Replication Target Administration page:



3. The page containing the replication status of the target appears. The overall status of replicated documents to the target appears in the top field. The documents that failed to replicate are listed in the second field, along with last and next replication retry times. You can click Retry to retry replication on selected documents or Retry All Documents to retry replication on all of the documents in the list.



Field	Description
Target	The name of the replication target.
Last Success	The date and time of the last successful update to the target.
Pending	The number of documents in the replicated domain that have not yet been replicated to the target.
Missing	The number of documents missing replication properties. For example, documents that were already in the replicated domain before replication was configured.
Error	The number of replication errors.
Document	The document that failed to replicate to the Replica.
Last Try	The last attempt to replicate the failed document.
Next Try	The next attempt to replicate the failed document.
Tries	The number of times MarkLogic Server attempted to replicate the failed document.

Field	Description
Last Error	The error generated on the last attempt to replicate the failed document.
Retry	Check box to retry replicating the failed document.

You can query the status of a specific document by entering its URI in the Check URI field and clicking Query. The status of the document, whether it replicated successfully or returned an error is displayed and highlighted in the above field.



6.0 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.