MarkLogic Server

Database Replication Guide

MarkLogic 9
May, 2017

Last Revised: 9.0-5, May 2018

Copyright © 2019 MarkLogic Corporation. All rights reserved.

Table of Contents

Database Replication Guide

1.0 Database Replication in MarkLogic Server 5
1.1 Terms Used in this Guide 5
1.2 Understanding Database Replication 6
1.2.1 Overview of Database Replication 7
1.2.2 Bulk Replication 8
1.2.3 Bootstrap Hosts 9
1.2.4 Inter-cluster Communication 10
1.2.5 Replication Lag 10
1.2.6 Master and Replica Database Index Settings 11
1.3 Example Database Replication Configurations 12
1.4 Database Replication for Disaster Recovery 15
1.5 Rolling Back to the Non-Blocking Timestamp 18
1.6 Upgrading Clusters Configured with Database Replication 20
1.7 Comparison of Database Replication with Flexible Replication 21

2.0 Database Replication Quick Start 23
2.1 Identify the Local Cluster to Foreign Clusters 24
2.2 Couple the Local Cluster with the Foreign Cluster 25
2.3 Configure Database Replication 27
2.4 Load Documents into the Master Database and Check Replication 30

3.0 Configuring Database Replication 31
3.1 Database Replication Security 31
3.2 Avoid Replicating the App-Services Database 31
3.3 Configuring Database Replication 32
3.4 Connecting Master and Replica Forests with Different Names 38
3.5 Enabling, Disabling, Suspending, and Resuming Database Replication 42
3.6 Deleting a Database Replication Configuration 43
3.7 Decoupling the Local and Foreign Clusters 44
3.8 Configuring App Servers on the Replica Cluster 46
3.9 Changing the Foreign Bind Port 46
3.10 TCP Tuning For High-Latency Environments 47

4.0 Checking Database Replication Status 49
4.1 Checking the State of the Replica Forests 49
4.2 Checking the Journal Frames Received from the Master 50
4.3 Checking the Update Lag on the Master 52

5.0 Technical Support 55

6.0 Copyright 57
6.0 COPYRIGHT 57

1.0 Database Replication in MarkLogic Server

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

- Terms Used in this Guide
- Understanding Database Replication
- Example Database Replication Configurations
- Database Replication for Disaster Recovery
- Rolling Back to the Non-Blocking Timestamp
- Upgrading Clusters Configured with Database Replication
- Comparison of Database Replication with Flexible Replication

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

1.1 Terms Used in this Guide

The following are the definitions for the Database 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.
- A *Local Cluster* is the cluster of MarkLogic Server hosts at your current location.
- A *Foreign Cluster* is a remote cluster of MarkLogic Server hosts.
- A *Master Database* is the database being replicated. In any Database Replication scheme there is a Master database and at least one Replica database.
- A *Master Forest* is the forest being replicated. In any Database Replication scheme there is a Master forest and at least one Replica forest.
- A *Master Cluster* is shorthand for the cluster that hosts a Master Database.
- A *Replica Database* is the database that receives replicated data from the Master database.
- A *Replica Forest* is the forest that receives replicated data from the Master forest.
- A *Replica Cluster* is shorthand for the cluster that hosts a Replica Database.
- A *Bootstrap Host* is the MarkLogic Server host machine used by a foreign cluster to initiate communication with the local cluster.
- A *Foreign Bind Port* is the port used on each host to handle XDQP communication with foreign clusters.

- *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 committing the transaction and proceeding with additional transactions. Database 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 from the Master database that existed before replication was configured.

1.2 Understanding Database Replication

Database Replication is the process of maintaining copies of forests on databases in multiple MarkLogic Server clusters. At a minimum, there will be a Master database and one Replica database. A Master database can replicate to multiple databases. A Replica database cannot serve as a Master database to another Replica.

Note: All hosts participating in Database Replication must:

- Run the same maintenance release of MarkLogic Server.
- Use the same operating system.

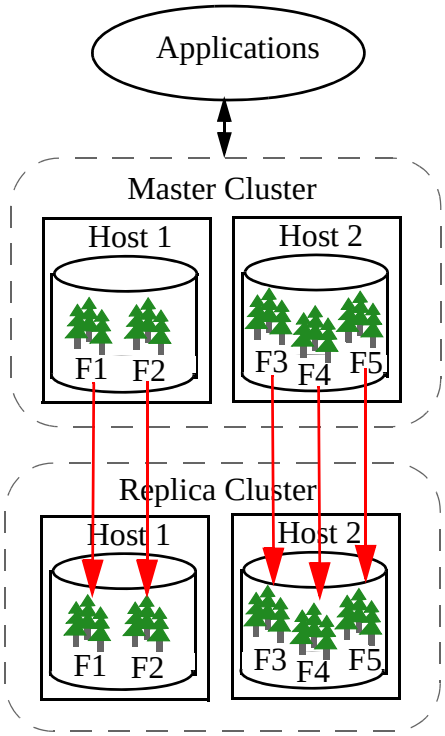
This section contains the following topics:

- Overview of Database Replication
- Bulk Replication
- Bootstrap Hosts
- Inter-cluster Communication
- Replication Lag
- Master and Replica Database Index Settings

1.2.1 Overview of Database Replication

Database replication operates at the forest level by copying journal frames from a forest in the Master database and replaying them on a corresponding forest in the foreign Replica database.

As shown in the illustration below, each host in the Master cluster connects to the remote hosts that are necessary to manage the corresponding Replica forests. Replica databases can be queried but cannot be updated by applications.



1.2.2 Bulk Replication

Any content existing in the Master databases before Database Replication is configured is bulk replicated into the Replica databases. Bulk replication is also used after the Master and foreign Replica have been detached for a sufficiently long period of time that journal replay is no longer possible. Once bulk replication has completed, journal replication will proceed.

The bulk replication process is as follows:

1. The indexing operation on the Master database maintains a catalog of the current state of each fragment. The Master sends this catalog to the Replica database.

2. The Replica compares the Master's catalog to its own and updates its fragments using the following logic:

    - If the Replica has the fragment, it updates the nascent/deleted timestamps, if they are wrong.
    - If the Replica has a fragment the Master doesn't have, it marks that fragment as deleted (it likely existed on the Master at some point in the past, but has been merged out of existence).
    - If the Replica does not have a fragment, it adds it to a list of missing fragments to be returned by the Master.

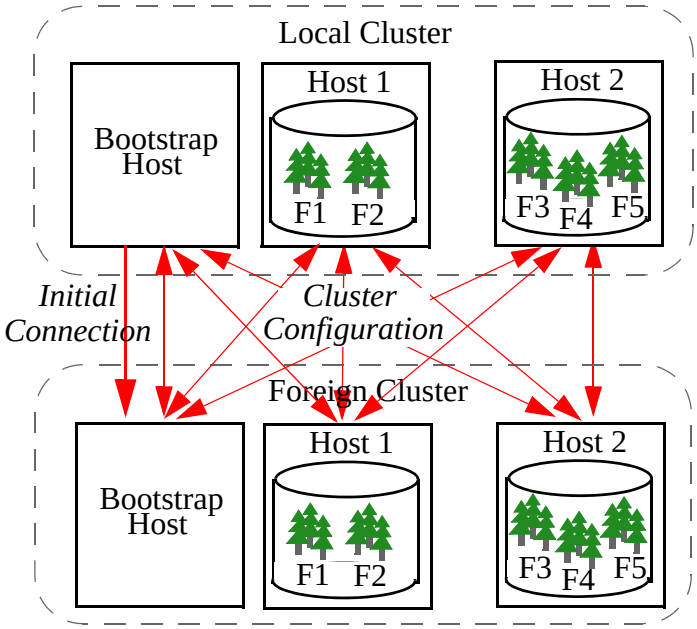
3. The Master iterates over the list of missing fragments returned from the Replica and sends each of them, along with their nascent/deleted timestamps, to the Replica where they are inserted.

For more information on fragments, see the Fragments chapter in the *Administrator's Guide*.

1.2.3 Bootstrap Hosts

Each cluster in a Database Replication scheme contains one or more bootstrap hosts that are used to establish an initial connection to foreign clusters it replicates to/from and to retrieve more complete configuration information once a connection has been established. When a host initially starts up and needs to communicate with a foreign cluster, it will bootstrap communications by establishing a connection to one or more of the bootstrap hosts on the foreign cluster. Once a connection to the foreign cluster is established, cluster configuration information is exchanged between all of the local hosts and foreign hosts.

For details on selecting the bootstrap hosts for your cluster, see Coupling Clusters in the *Administrator's Guide*.



1.2.4 Inter-cluster Communication

Communication between clusters is done using the intra-cluster XDQP protocol on the *foreign bind port*. A host will only listen on the foreign bind port if it is a bootstrap host or if it hosts a forest that is involved in inter-cluster replication. By default, the foreign bind port is port 7998, but it can be configured for each host, as described in “Changing the Foreign Bind Port” on page 46. When secure XDQP is desired, a single certificate / private-key pair is shared by all hosts in the cluster when communicating with foreign hosts.

XDQP connections to foreign hosts are opened when needed and closed when no longer in use. While the connections are open, foreign heartbeat packets are sent once per second. The foreign heartbeat contains information used to determine when the foreign cluster's configuration has changed so updated information can be retrieved by the local bootstrap host from the foreign bootstrap host.

1.2.5 Replication Lag

Queries on a Replica database must run at a timestamp that lags the current cluster commit timestamp due to replication lag. Each forest in a Replica database maintains a special timestamp, called a Non-blocking Timestamp, that indicates the most current time at which it has complete state to answer a query. As the Replica forest receives journal frames from its Master, it acknowledges receipt of each frame and advances its nonblocking timestamp to ensure that queries on the local Replica run at an appropriate timestamp. Replication lag is the difference between the current time on the Master and the time at which the oldest unacknowledged journal frame was queued to be sent to the Replica.

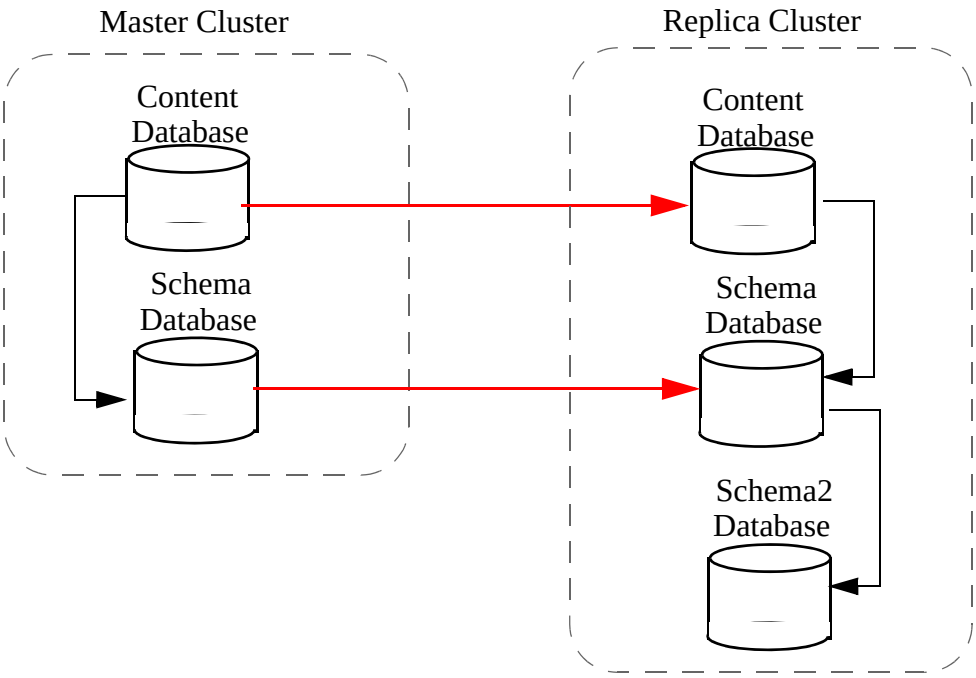
You can set a lag limit in your configuration that specifies, if the Master does not receive an acknowledgement from the Replica within the time frame specified by the lag limit, transactions on the Master are stalled. The default lag limit is 15 seconds, and should work well for most installations. For the procedure to set the lag limit, see “Configuring Database Replication” on page 32.

1.2.6 Master and Replica Database Index Settings

Indexing information is not replicated by the Master database and is instead regenerated on the Replica system. If you want the option to switch over to the Replica database after a disaster, the index settings must be identical on the Master and Replica clusters.

If you need to update index settings after configuring Database Replication, update the settings on the Replica database before updating those on the Master database. This is because changes to the index settings on the Replica database only affect newly replicated documents and will not trigger reindexing on existing documents. Changes to the index settings on the Master database will trigger reindexing, after which the reindexed documents will be replicated to the Replica. When a Database Replication configuration is removed for the Replica database (such as after a disaster), the Replica database will reindex, if necessary.

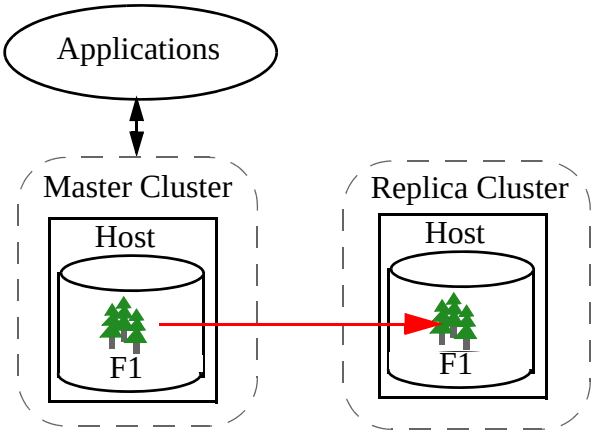
The process of recreating index information on the Replica database requires information located in the schema database. You cannot replicate a Master database that acts as its own schema database. When replicating a Master schema database, create a second empty schema database for the Replica schema database on the Replica cluster. The diagram below shows the Schema2 database as the schema database from the Replica schema database.



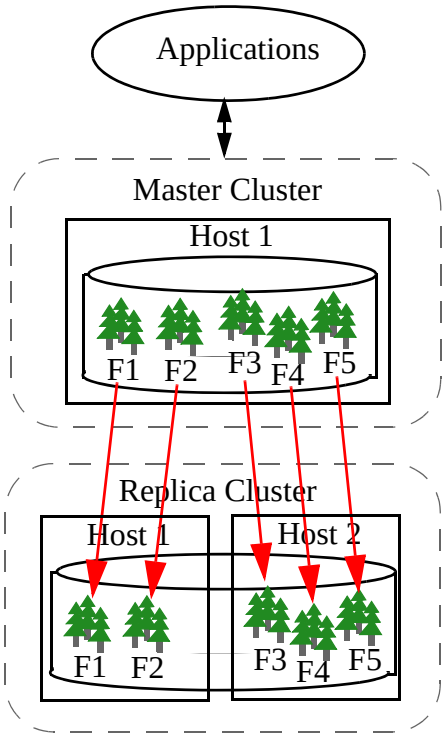
1.3 Example Database Replication Configurations

This section describes some of the possible Database Replication configurations.

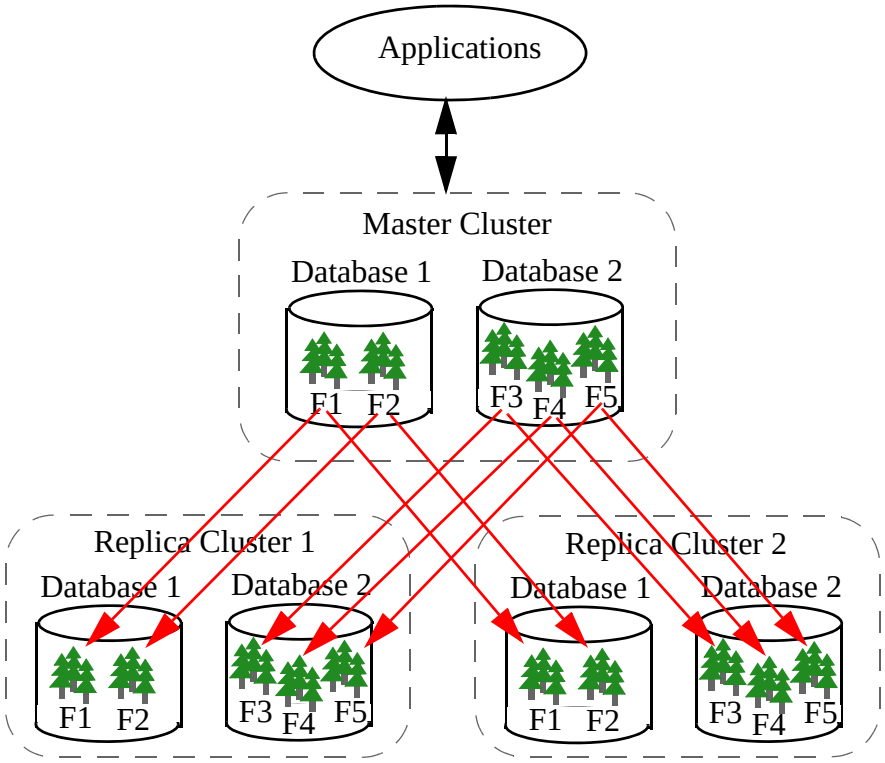
The most basic form of Database Replication consists of a two clusters, each containing a single database with a single forest, as shown below.



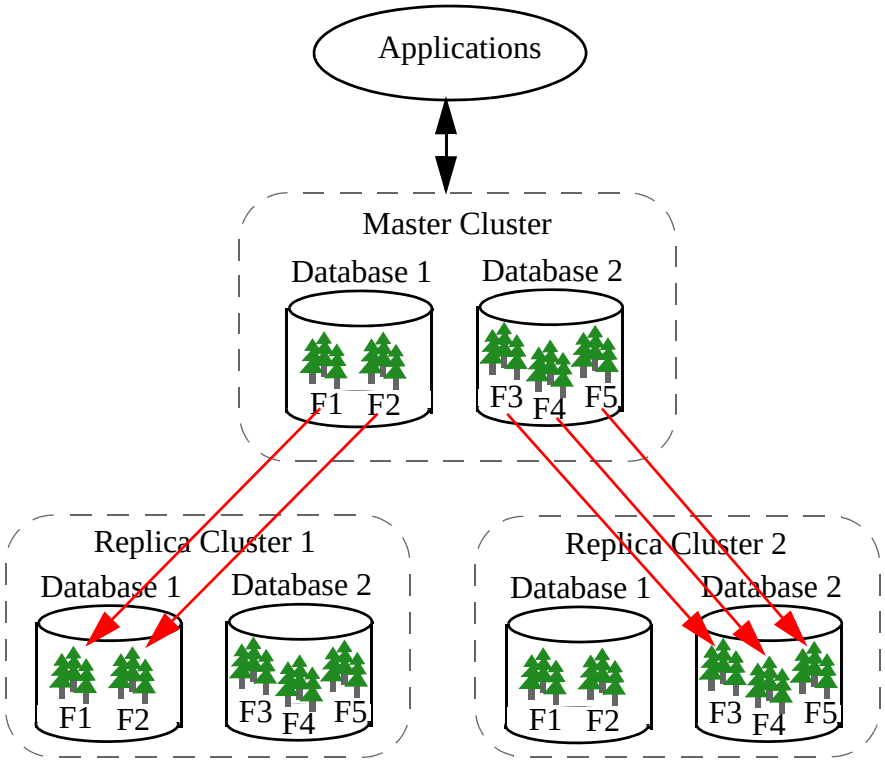
Master and Replica forests can be distributed differently across hosts in each cluster. For example, as shown below, all of the Master forests may reside on a single host and the Replica forests on two hosts.



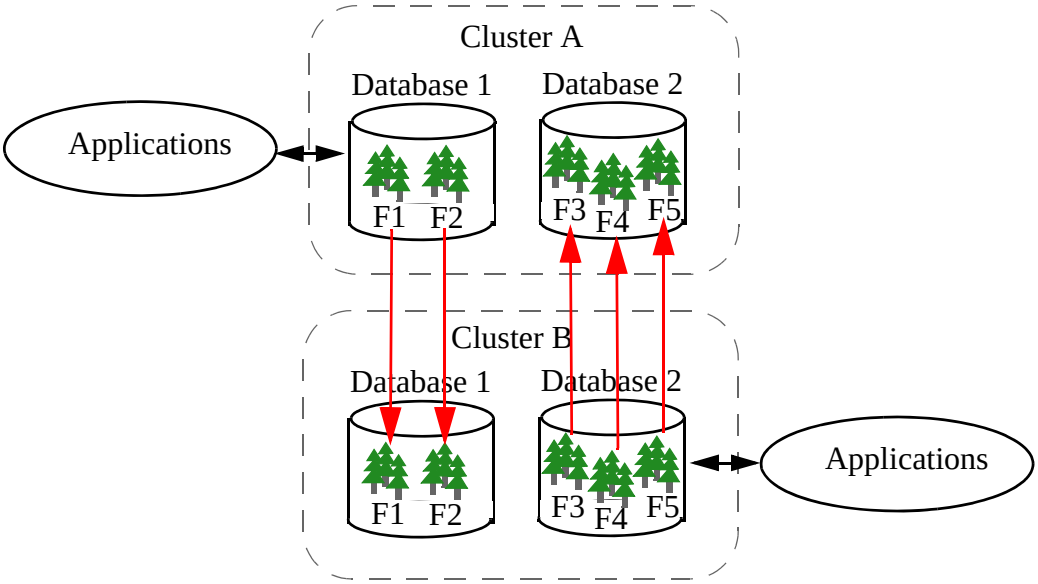
A Master cluster can replicate forests to multiple Replica clusters. In the example shown below, the Master maintains a copy of each forest on two clusters:



In the example shown below, the Master maintains a single copy of Database 1 on Replica Cluster 1 and a single copy of Database 2 on Replica Cluster 2:



A cluster can contain both Master and Replica databases. On Cluster A in the example shown below, Database 1 is the Master database and Database 2 is the Replica database.



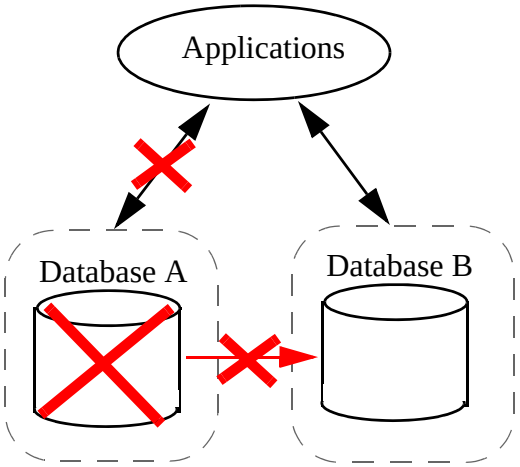
1.4 Database Replication for Disaster Recovery

The purpose of Database Replication is to make data continuously available to mission-critical applications with minimal impact to application performance. In a typical disaster recovery scheme, clusters are located in data centers at different geographical locations. In the event of a disaster in the location of the Master cluster, the data can be made available by the Replica cluster

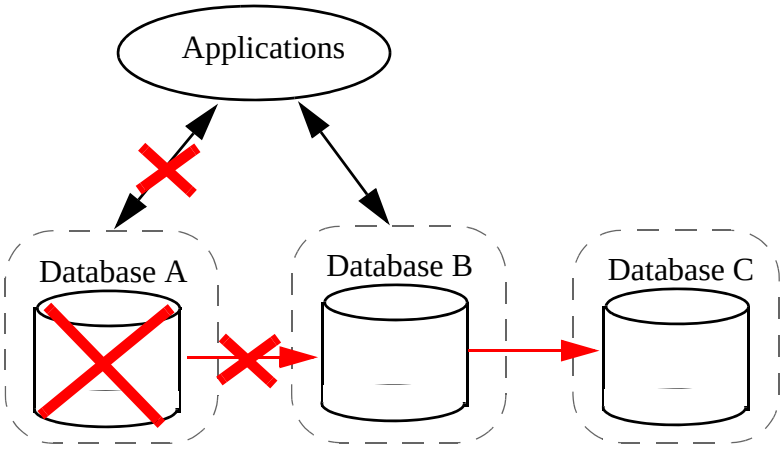
> **Note:** In this section, the original Master and Replica databases are referred to as Database A and Database B, respectively.

There are two basic approaches to a disaster recovery operation, each of which involves redirecting the applications to Database B. Before executing your applications on Database B, the database must be rolled back, as described in “Rolling Back to the Non-Blocking Timestamp” on page 18.

- Simply disable Database Replication on Database B so that it is no longer a Replica and can accept updates.

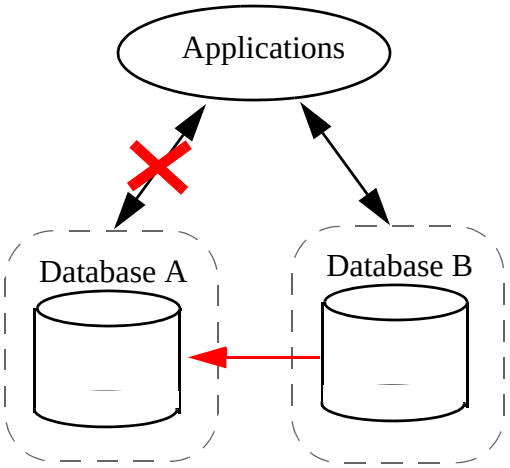


- Reconfigure Database B as the Master database and replicate updates to a third foreign database (Database C).

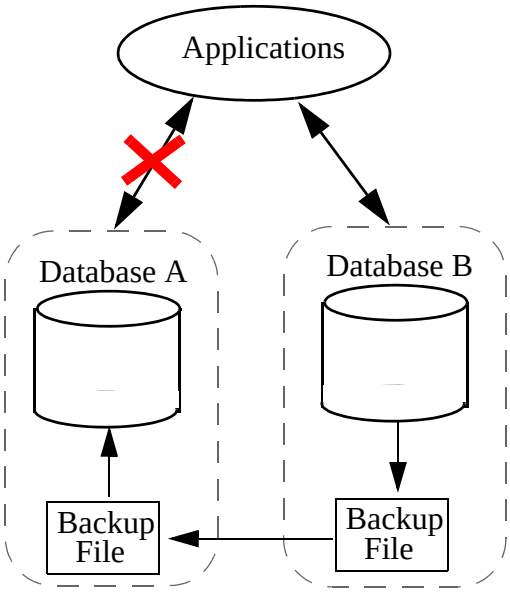


When Database A is returned to service, you can resynchronize it with Database B by either:

- Configuring Database A as a Replica of Database B. In this case, Database B will automatically bulk replicate to Database A.

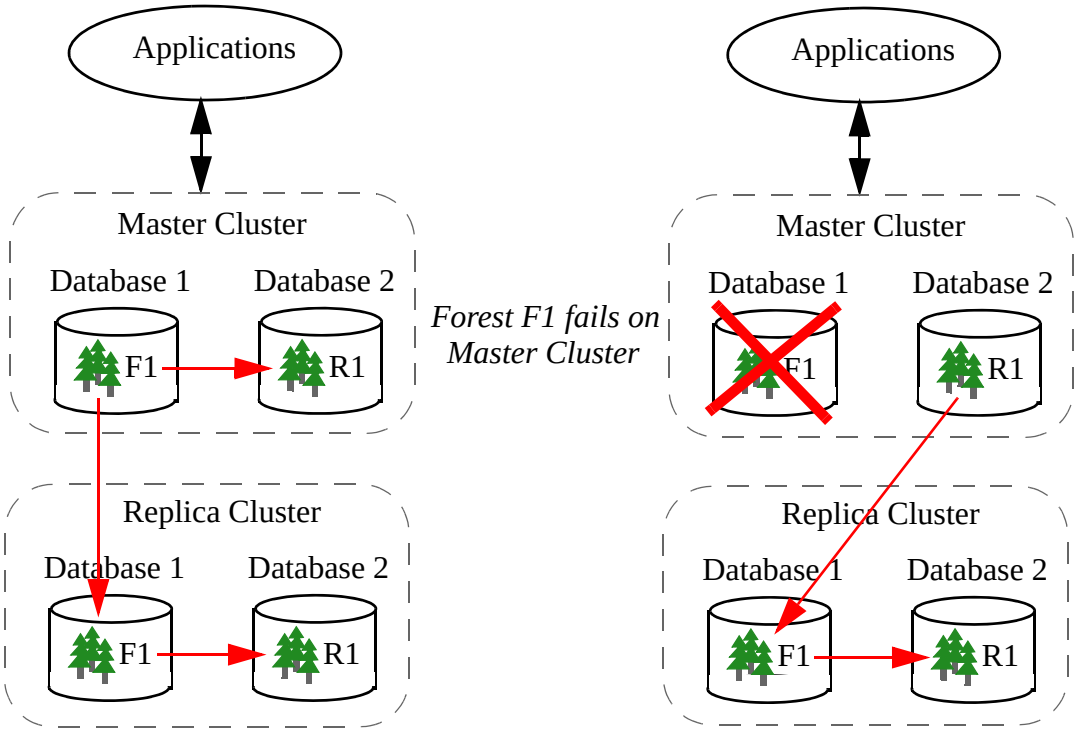


- Running a backup operation on Database B and restoring Database A from the backup file, as described in Backing Up and Restoring a Database in the *Administrator's Guide*.



Once the data has been restored on the Database A, you can redirect the applications from Database B back to Database A.

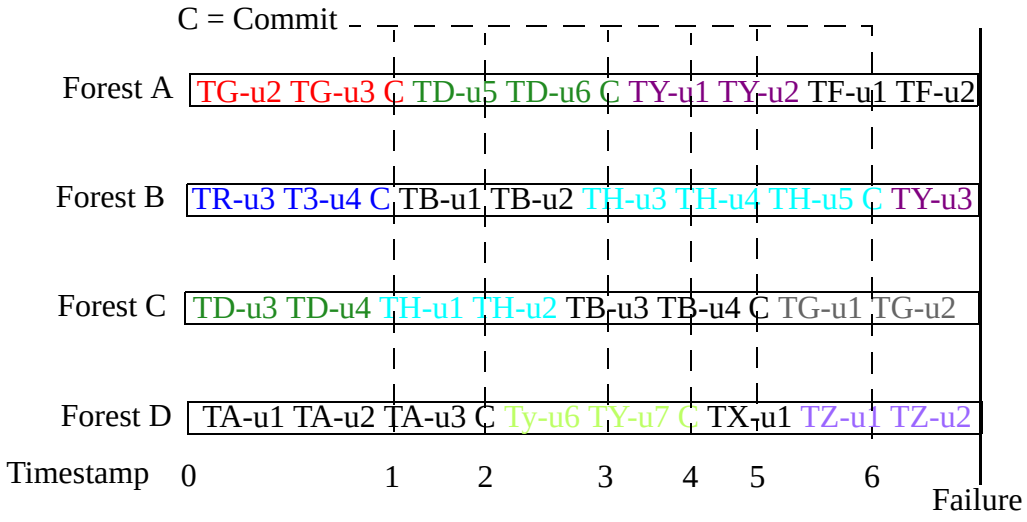
Database Replication can be used in combination with the local-disk failover feature described in High Availability of Data Nodes With Failover and Configuring Local-Disk Failover for a Forest in the *Scalability, Availability, and Failover Guide*. For example, the Master and Replica clusters shown below are configured for both local-disk failover and Database Replication. Should, for example, the F1 forest in the Master cluster become unavailable, then the R1 forest resumes replication to the F1 forest in the Replica cluster.



1.5 Rolling Back to the Non-Blocking Timestamp

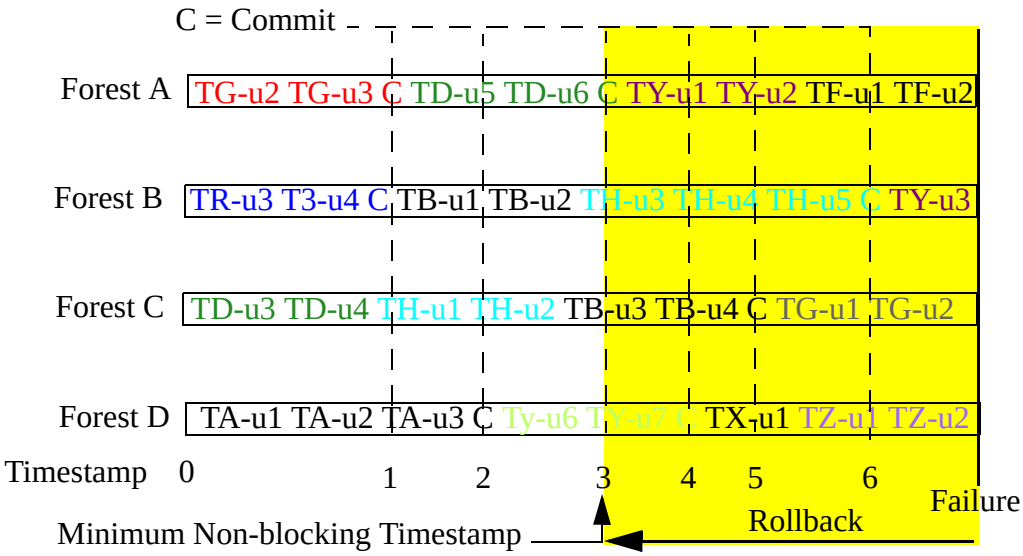
After you fail over your applications to a Replica database, each Replica forest will likely have committed its last transaction at different timestamps. If the Replica database has multiple forests and relationships exist between those forests, this inconsistency may cause problems. In order to return the database to a transactionally consistent state, all forests must be rolled back to the minimum non-blocking timestamp.

For example, the illustration below shows four forests and their committed transactions. Updates for each transaction are identified by the convention 'T#-u#' and commits are identified by a 'C'. Each forest completed its last commit at a different point in time when the failure of the Master database occurred. In this example, Forest A has only committed transactions up to timestamp 3 while Forest B has committed transactions up to timestamp 6. This means that, in order to return the database to a transactionally consistent state, all forests must be rolled back to *at least* timestamp 3.



Note: The procedure described in this section can also be applied to the Master database, should you want to roll it back to some point after the last merge timestamp. When the Master database is rolled back, the rollback will be replicated to the Replica databases.

For example, the database you want to restore has four forests, as shown below. You use the `xdmp:forest-status` function to locate the `nonblocking-timestamp` value for each forest and roll back all of your forests to the minimum `nonblocking-timestamp` value. In this example, the minimum `nonblocking-timestamp` is the timestamp of the last committed transaction in Forest A.



The following procedure describes how to restore to the minimum non-blocking timestamp using the XQuery API.

1. Use the `admin:database-get-merge-timestamp` function to get the current merge timestamp. Save this value so it can be reset after you have completed the rollback operation.

2. Use the `admin:database-set-merge-timestamp` function to set the merge timestamp to any time before the failure of the master database. This will preserve fragments in merge after this timestamp until you have rolled back your forest data.

3. Use the `xdmp:forest-rollback` function to roll back the forests to the minimum `nonblocking-timestamp` returned by the `xdmp:forest-status` function for each forest that is in the `open` or `open replica` state.

For example, you can use the following query to rollback your forest data for the Documents database:

```
xquery version "1.0-ml";

let $db := xdmp:database("Documents")
let $timestamp := xdmp:database-nonblocking-timestamp($db)
let $rollback := xdmp:forest-rollback(
    xdmp:forest-open-replica(
    xdmp:database-forests($db)),
    $timestamp)

return
    "Roll back done"
```

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

   Note: When the Security database is replicated, amps and triggers refer to Modules database IDs in the Master cluster. When failing over to the Security database on the Replica cluster, use the `sec:amps-change-modules-database` and `trgr:triggers-change-modules-database` functions to map Modules database IDs from the Master cluster IDs to the Replica cluster IDs.

1.6 Upgrading Clusters Configured with Database Replication

The XDQP protocol version must be the same on all hosts in all of the clusters being replicated. If the XDQP protocol version is different, database replication will pause until the clusters are both upgraded to the same version. XDQP protocol versions generally change between major versions of MarkLogic, but they can sometimes change between maintenance releases.

If the Security database isn't replicated, then there shouldn't be anything special you need to do other than upgrade the two clusters.

If the security database is replicated, do the following:

1. Upgrade the Replica cluster and run the upgrade scripts. This will update the Replica's Security database to indicate that it is current. It will also do any necessary configuration upgrades.

2. Upgrade the Master cluster and run the upgrade scripts. This will update the Master's Security database to indicate that it is current. It will also do any necessary configuration upgrades.

1.7 Comparison of Database Replication with Flexible Replication

MarkLogic Server provides two types of replication: Database Replication, as described in this guide, and Flexible Replication, as described in the *Flexible Replication Guide*. You should choose a replication type based on your application and high-availability requirements

Depending on your requirements, one type of replication might have an advantage over the other.

The key differences between the these two types of replication are:

- Replicate all or part of the Master database — Database Replication replicates the entire Master database to the Replica database. Flexible Replication replicates Content Processing Framework (CPF) domains and allows you to write XQuery filters that enable you to modify the replicated documents in some manner, determine whether to replicate a change, or select which parts of a document will be replicated. As a consequence, Flexible Replication enables you to replicate select content from the Master database.
- Performance — Flexible Replication uses CPF and generates information in the properties of each replicated document. Database Replication simply copies and replays journal frames. If you replicate a complete database, you can expect less overhead with Database replication than with Flexible Replication.
- Transactional — Database Replication replicates as soon as a document is inserted or updated and ensures that a single multi-document transaction in the Master database replicates as a single multi-document transaction to the Replica database. Flexible Replication replicates documents after each transaction without consideration of transactional grouping in the Master database. For example, if you have a transaction that commits 10 documents to the Master database and some sort of failure occurs, with Flexible Replication it is possible for the Replica to end up with 5 documents from that transaction on the Replica. With Database Replication, the Replica will either have no documents or all 10 of them.

2.0 Database Replication Quick Start

This chapter provides the quick-start procedures for creating a simple Database Replication configuration to replicate the Documents database from one MarkLogic Server host to another. The replicated MarkLogic Server hosts must be able to communicate with each other and they must not be on the same cluster.

> **Note:** The purpose of this chapter is to walk you through a specific Database Replication configuration procedure. For more general details on how to configure Database Replication, see "Configuring Database Replication" on page 31.

> **Note:** A cluster can consist of a single MarkLogic Server host.

This chapter includes the following sections:

- Identify the Local Cluster to Foreign Clusters
- Couple the Local Cluster with the Foreign Cluster
- Configure Database Replication
- Load Documents into the Master Database and Check Replication

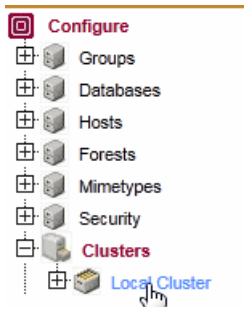
All of the procedures described in this chapter are done using the Admin Interface described in the Administrative Interface chapter in the *Administrator's Guide*.

2.1 Identify the Local Cluster to Foreign Clusters

Each cluster may be configured to have a descriptive name. It is a good practice to assign each cluster a name before coupling them with each other.

This section describes how to name the local cluster so it can be identified by foreign clusters. Repeat this procedure on a host for each cluster that is to participate in Database Replication.

1. On any host in the local cluster, select Local Cluster under Clusters at the bottom of the left-hand menu:



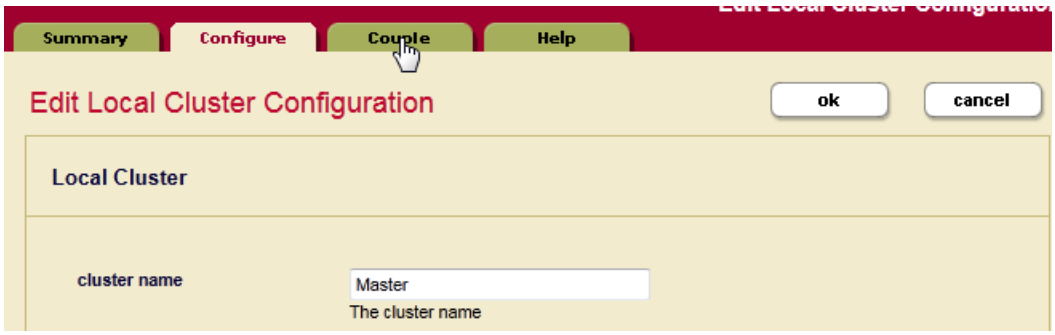
2. In the Edit Local Cluster Configuration page, enter the cluster name and select a host as the bootstrap host:



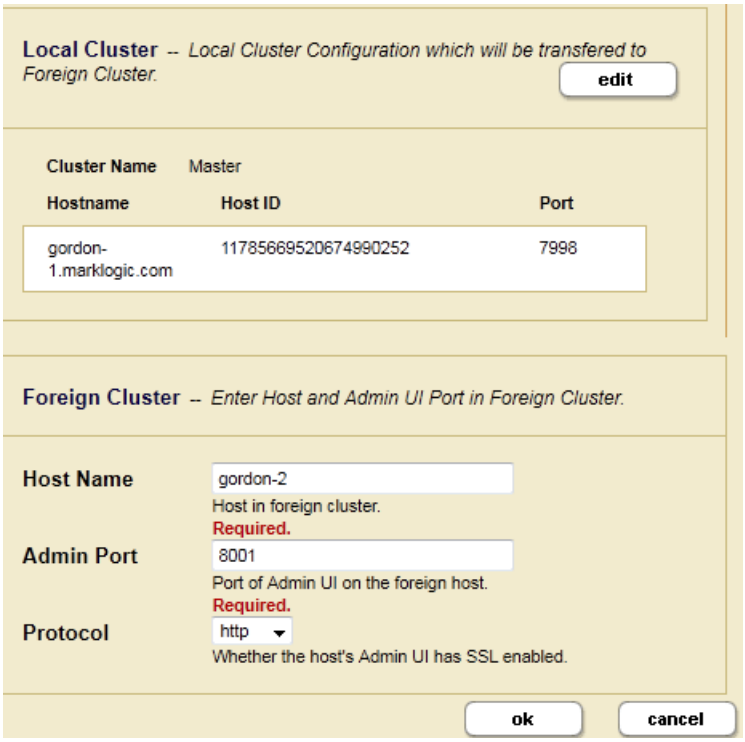
2.2 Couple the Local Cluster with the Foreign Cluster

Before you can configure Database Replication, each cluster in the replication scheme must be aware of the configuration of the other clusters. This can be accomplished by "coupling" the local cluster to the foreign cluster.

1. In the Local Cluster Configuration page, select the Couple tab.



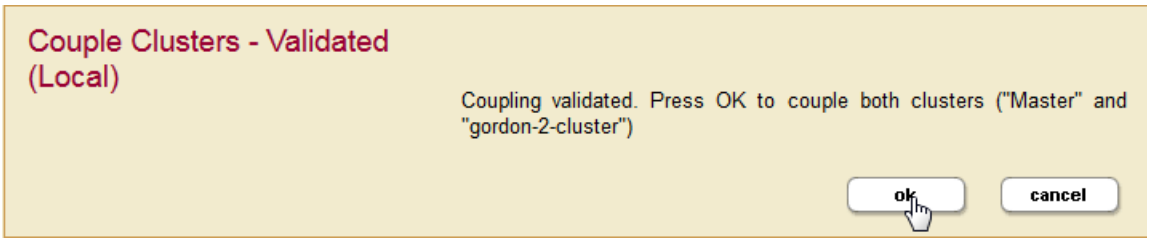
2. In the Foreign Cluster portion of the Local Cluster Configuration page, enter the Host Name for the bootstrap host in the foreign cluster. Leave the default values for the other settings and click OK.



3. In the Specify SSL settings for XDQP communication & Timeouts page, leave all of the default settings and click OK.



4. In the Confirm to Couple Clusters page, click OK. Click OK for any subsequent validate screens.

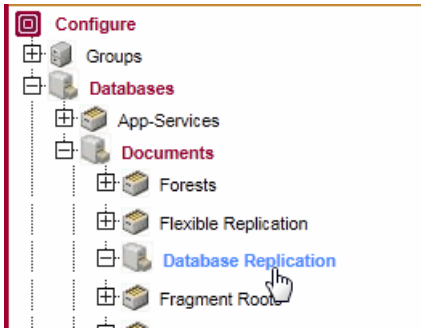


5. The Summary window appears and displays the summary for the Foreign Cluster configuration.

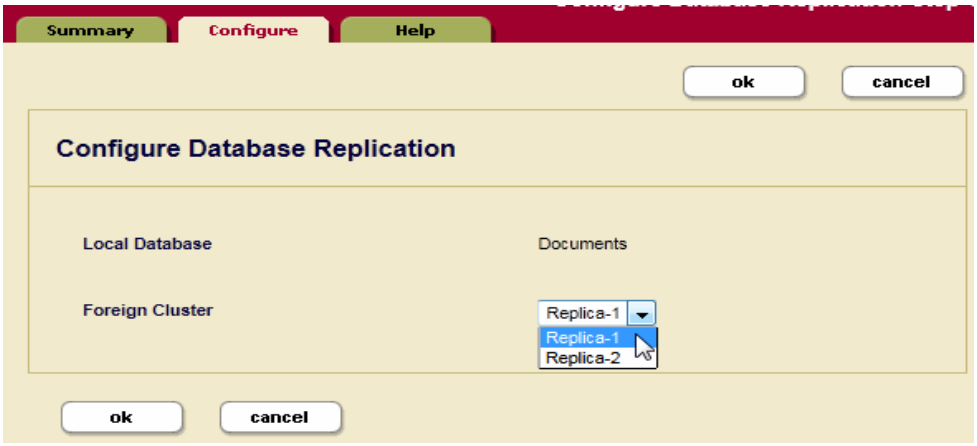
2.3 Configure Database Replication

The procedure described in this section configures Database Replication to replicate the Documents database on the Master to the Documents database on the Replica.

1. On the Master host, navigate to the Documents database in the left-hand menu and select Database Replication:



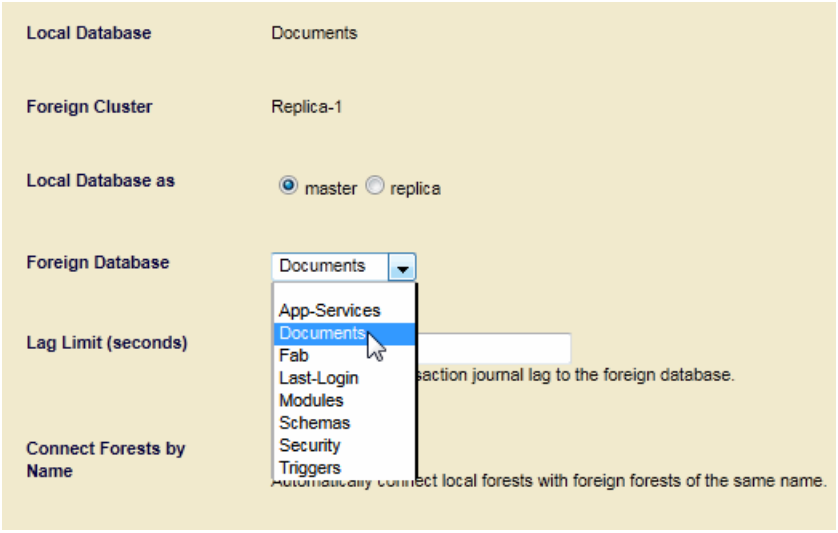
2. Select the Configure tab and select the foreign cluster from the pull-down menu. Click OK.



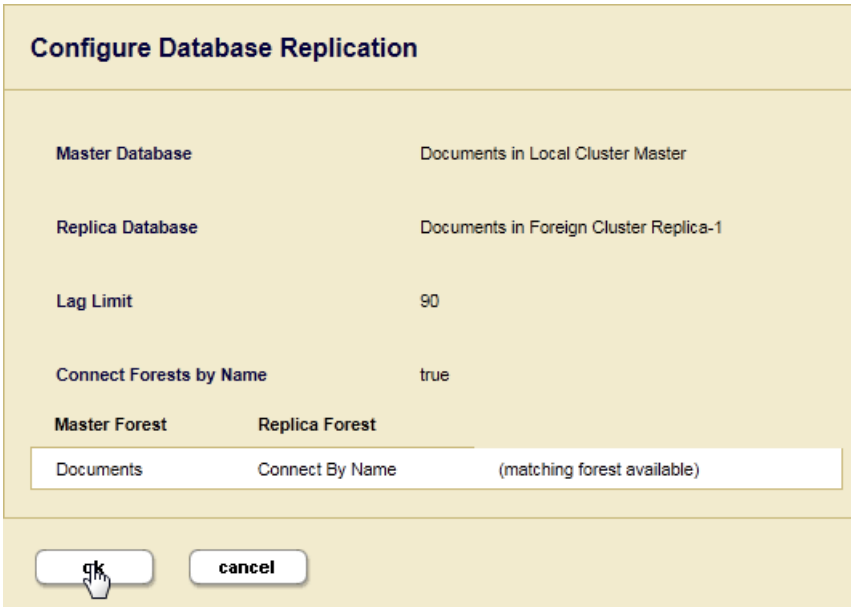
3. In the Configure Database Replication page, set Master for Local Database as.



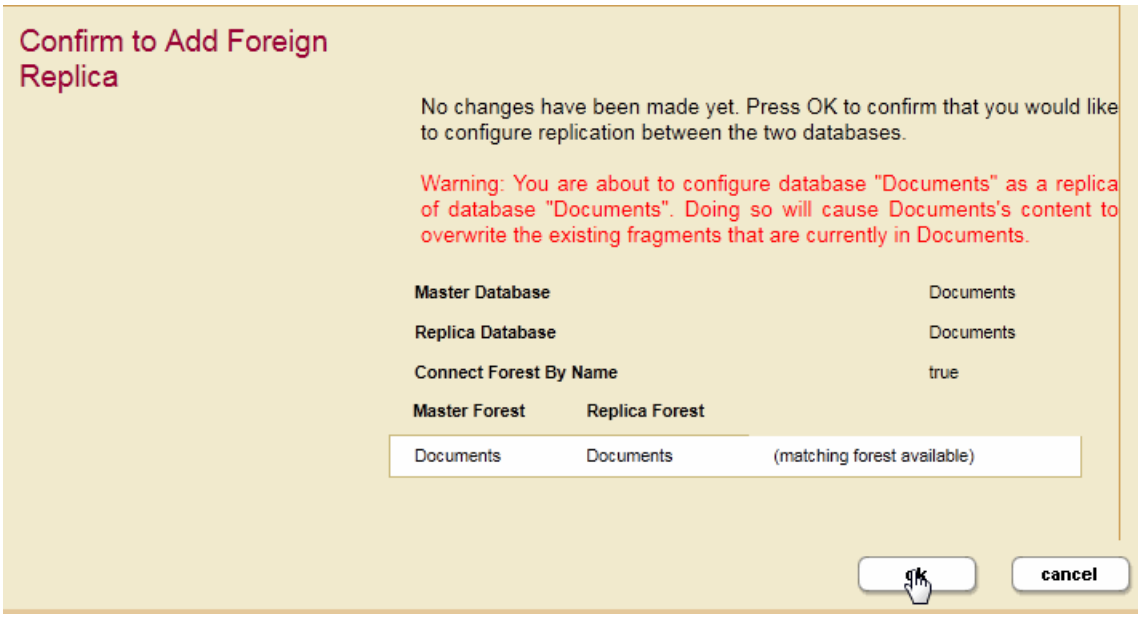
4. Select the Documents database in the Foreign Database pull-down menu. Leave all other settings as default. Click OK.



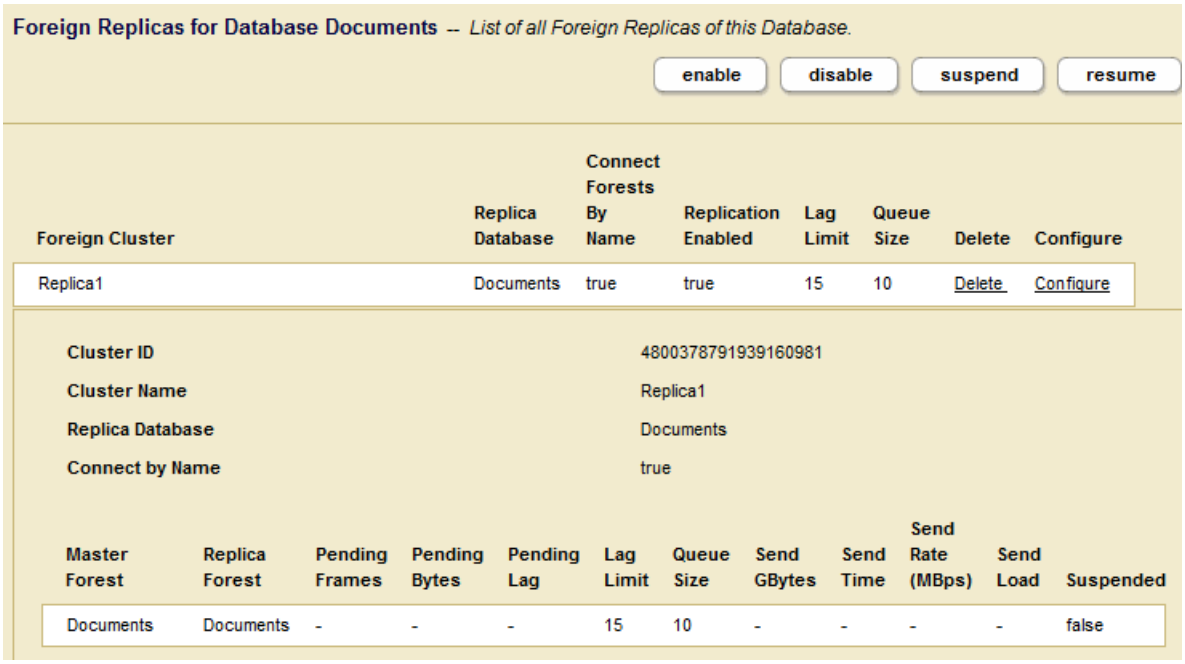
5. A page appears to confirm the Database Replication configuration on the Master. Click OK.



6. The Confirm to Add Foreign Replica page appears. Click OK.



7. The Summary page appears with the Database Replication settings.



2.4 Load Documents into the Master Database and Check Replication

Once you have completed the Database Replication configuration procedures described in this chapter, documents loaded into the Documents database on the Master host will be replicated to the Documents database on the Replica host.

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 a Database in the *Query Console User Guide*.

3.0 Configuring Database Replication

This chapter describes how to configure your MarkLogic Server clusters for Database Replication. The topics in this chapter assume you are familiar with the replication principles described in "Understanding Database Replication" on page 6 and the basic configuration procedures described in the "Database Replication Quick Start" on page 23.

For details on using the REST API to script database replication, see Scripting Database Replication Configuration in the *Scripting Administrative Tasks Guide*.

This chapter includes the following sections:

- Database Replication Security
- Avoid Replicating the App-Services Database
- Configuring Database Replication
- Connecting Master and Replica Forests with Different Names
- Enabling, Disabling, Suspending, and Resuming Database Replication
- Deleting a Database Replication Configuration
- Decoupling the Local and Foreign Clusters
- Configuring App Servers on the Replica Cluster
- Changing the Foreign Bind Port
- TCP Tuning For High-Latency Environments

3.1 Database Replication Security

The `admin` role is required to configure and couple local and foreign clusters and to configure Database Replication.

When coupling clusters, you can configure SSL to encrypt the XDQP data passed between them. For details on configuring SSL, see Coupling Clusters in the *Administrator's Guide*. If SSL is enabled for XDQP, you must set the protocol to HTTPS in the procedures described in Coupling Clusters in the *Administrator's Guide*.

3.2 Avoid Replicating the App-Services Database

The App-Services database is used to store information used in Query Console and for other internal applications. Therefore, it needs to be writable on all hosts, otherwise those applications might not work correctly. Therefore, MarkLogic recommends that you do not set up database replication on the App-Services database.

3.3 Configuring Database Replication

Before you configure clusters for Database Replication, you must couple the clusters as described in Coupling Clusters in the *Administrator's Guide*.

This section describes how to configure Database Replication on the cluster that hosts the Master database. Before starting this procedure, you must have identified all of the foreign clusters you plan to replicate to by following the procedure described in Coupling Clusters in the *Administrator's Guide*.

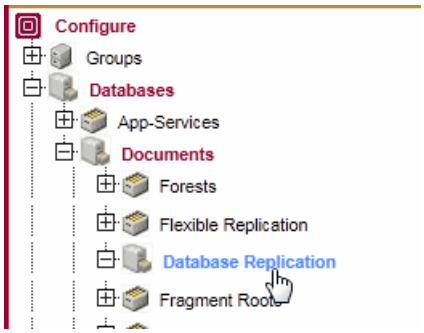
This section describes how to configure Database Replication from the Master cluster. You can also configure Database Replication on a Replica cluster. However, if you are replicating to multiple Replica clusters, it is more convenient to do all of the configuration from the Master cluster.

Note: When replicating a Master schema database, you must create a second empty schema database for the Replica schema database on the Replica cluster. For details see "Master and Replica Database Index Settings" on page 11.

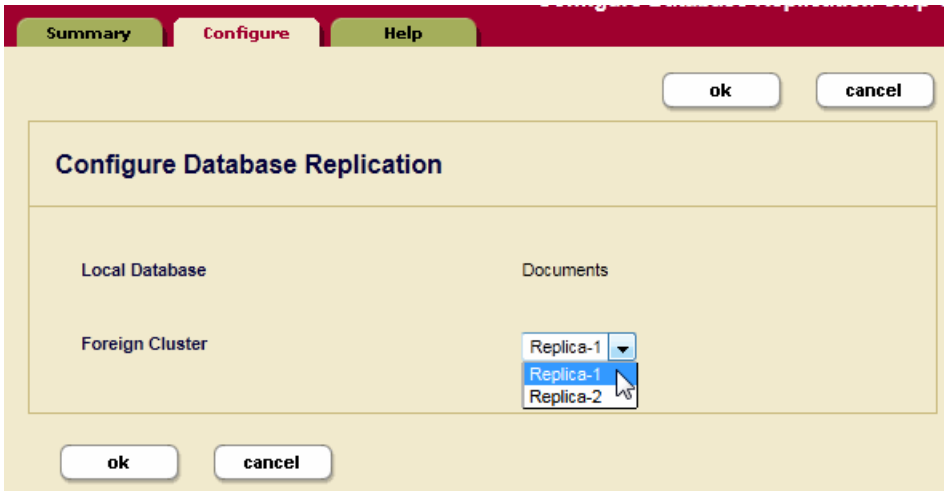
Warning Any documents with URIs in a Replica forest that are not in its respective Master forest will be automatically cleared when Database Replication is configured.

Repeat the following procedure for each replicated database and for each Replica cluster.

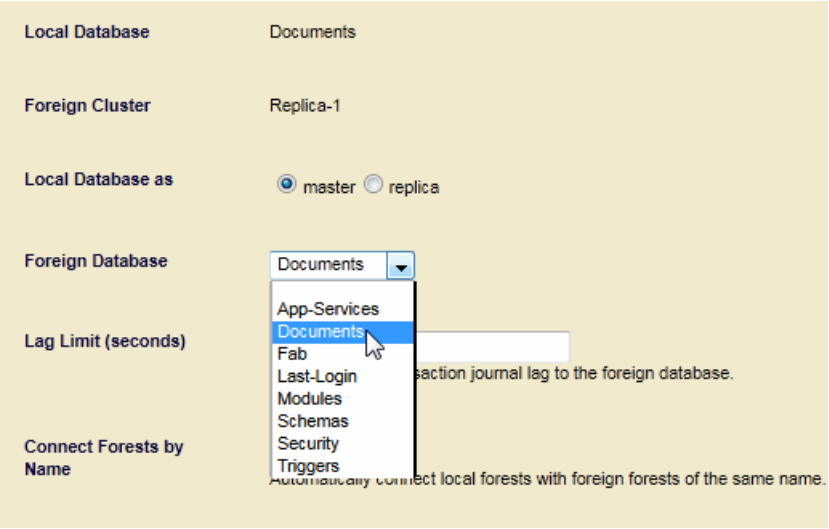
1. On the bootstrap host in the Master cluster, navigate to the database to be replicated in the left-hand menu and select Database Replication:



2. Select the Configure tab and then the Replica cluster in the Foreign Cluster pull-down menu:



3. The Configure Database Replication page will change. For Local Database as, select Master. Select the database to be replicated to in the Foreign Database pull-down menu.

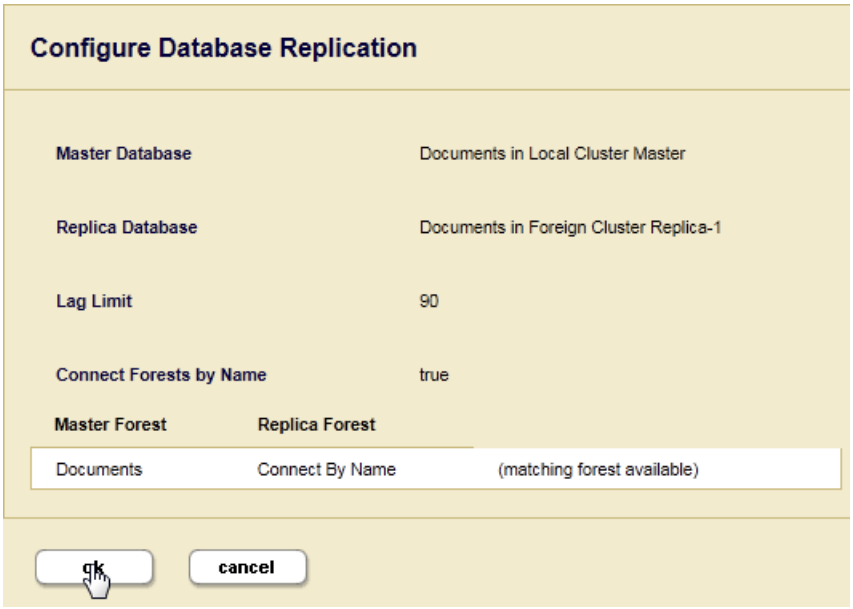


The other settings on the Configure Database Replication page are:

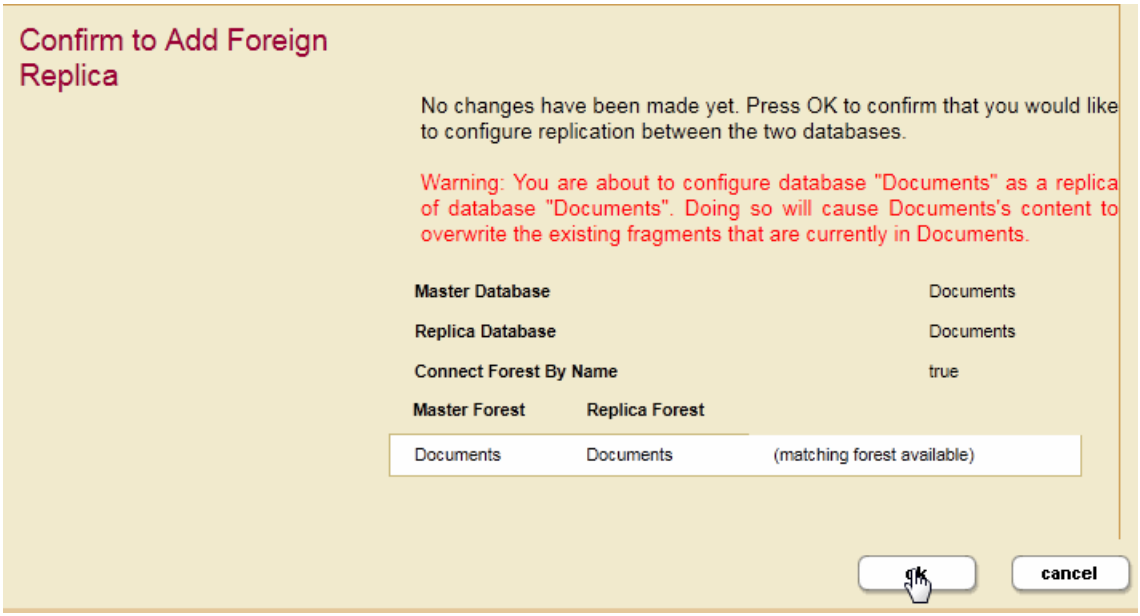
Database Replication Setting	Description
Lag Limit	The Replica cluster sends an acknowledgement to the Master cluster each time it receives a replicated journal frame. You can set a Lag Limit in your configuration that specifies that transactions on the Master will be stalled if the Master does not receive an acknowledgement from the Replica within the number of seconds specified by the Lag Limit. By default the Lag Limit is 15 seconds. For more information on Lag Limit, see “Replication Lag” on page 10.
Connect Forests by Name	If the forest names are the same on the Master and Replica clusters, set Connect Forests by Name to true. If new forests of the same name are later added to the Master and Replica databases, they will be automatically configured for Database Replication. If you set Connect Forests by Name to false, you must manually connect the Master forests on the local cluster to the Replica forests on the foreign cluster, as described in “Connecting Master and Replica Forests with Different Names” on page 38.

Database Replication Setting	Description
Foreign Admin UI Protocol	The communication protocol to be used to connect to the foreign Admin App Server. Select https if SSL is enabled on the Admin App Server on the bootstrap host in the foreign cluster. Otherwise, select http.
Foreign Admin UI Port	If the Admin App Server for the bootstrap host on the foreign cluster uses a port other than 8001, specify the port number.
Host in Foreign Cluster	The name of a host on the foreign cluster.

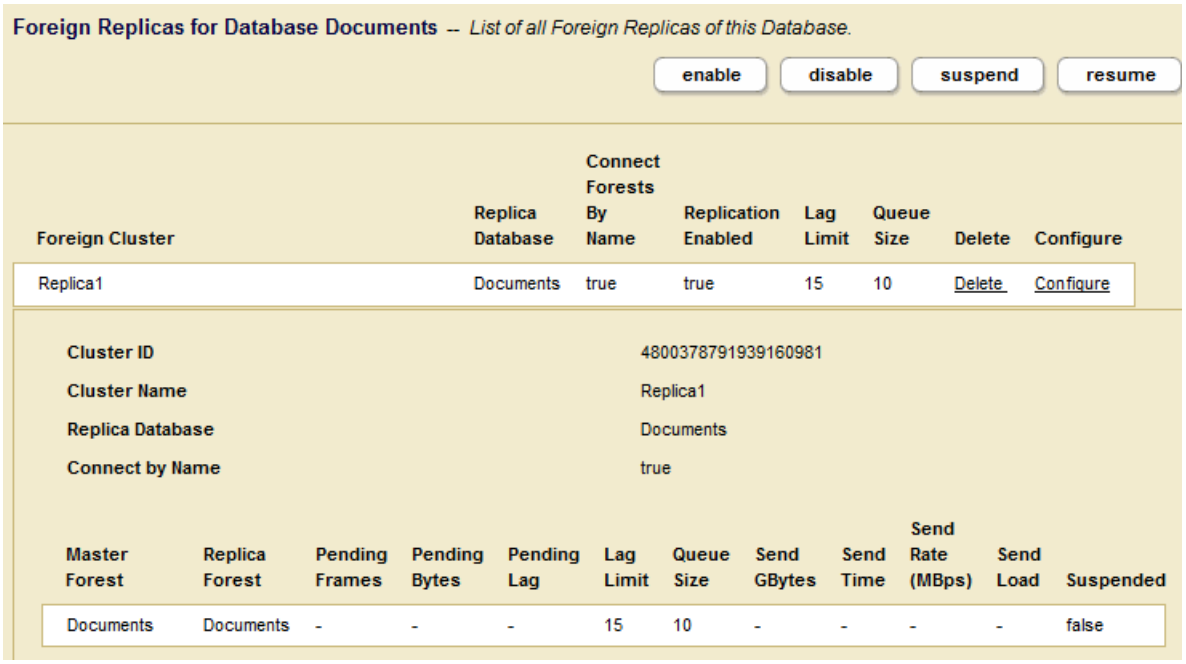
4. When you have finished, click OK. The final configuration page appears to confirm the Database Replication configuration. Click OK.



5. The Verify add of foreign Replica/Master page appears. Click OK.



6. The Summary page appears with the Database Replication settings.



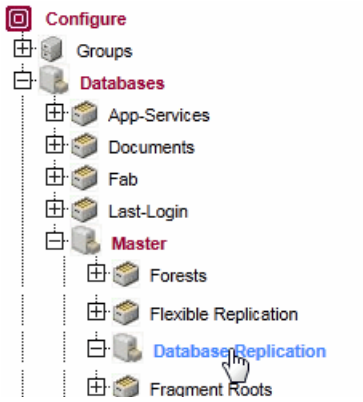
3.4 Connecting Master and Replica Forests with Different Names

If you are replicating between databases that contain forests with different names, you can manually connect Master to Replica forests.

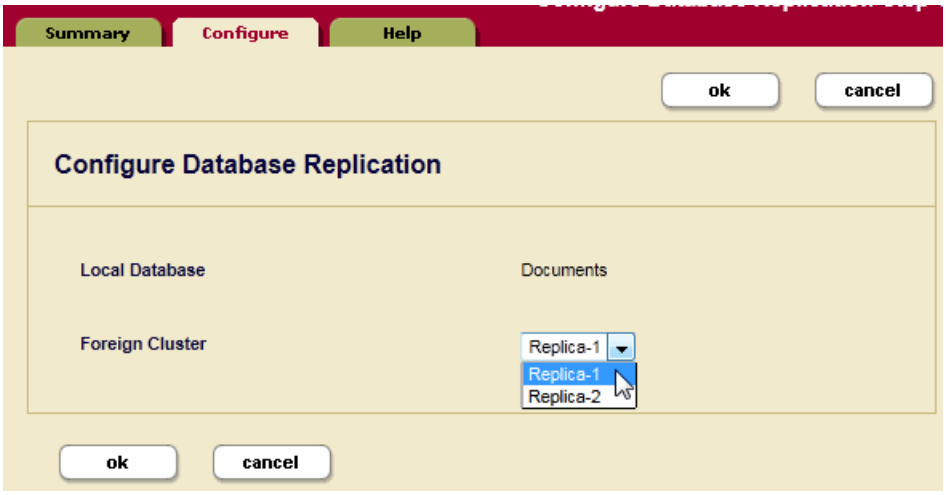
> **Note:** The Replica databases must have at least as many forests as the Master database. Otherwise, not all of the data on the Master database will be replicated.

For example, if you have a database named Master on the Master cluster and a database named Replica on the Replica cluster and each database has unique forest names, use the following procedure.

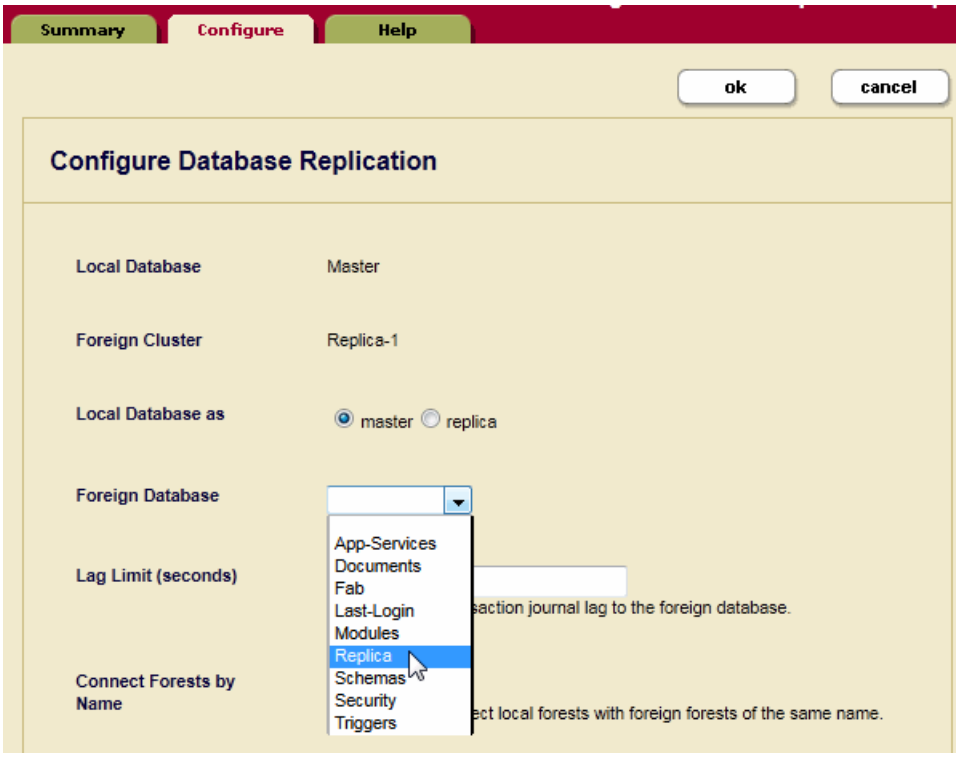
1. On the bootstrap host in the Master cluster, navigate to the Master database in the left-hand menu and select Database Replication:



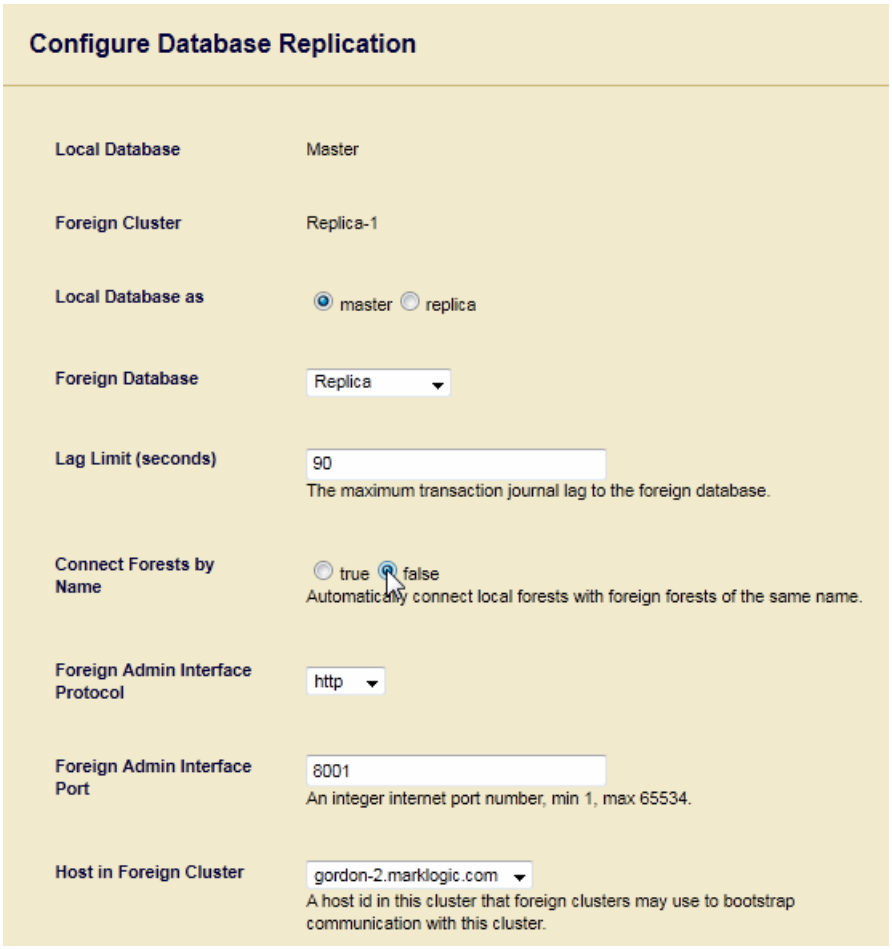
2. In the Configure Database Replication Step 1 page, select the Foreign Cluster that contains the Replica Database. Click OK.



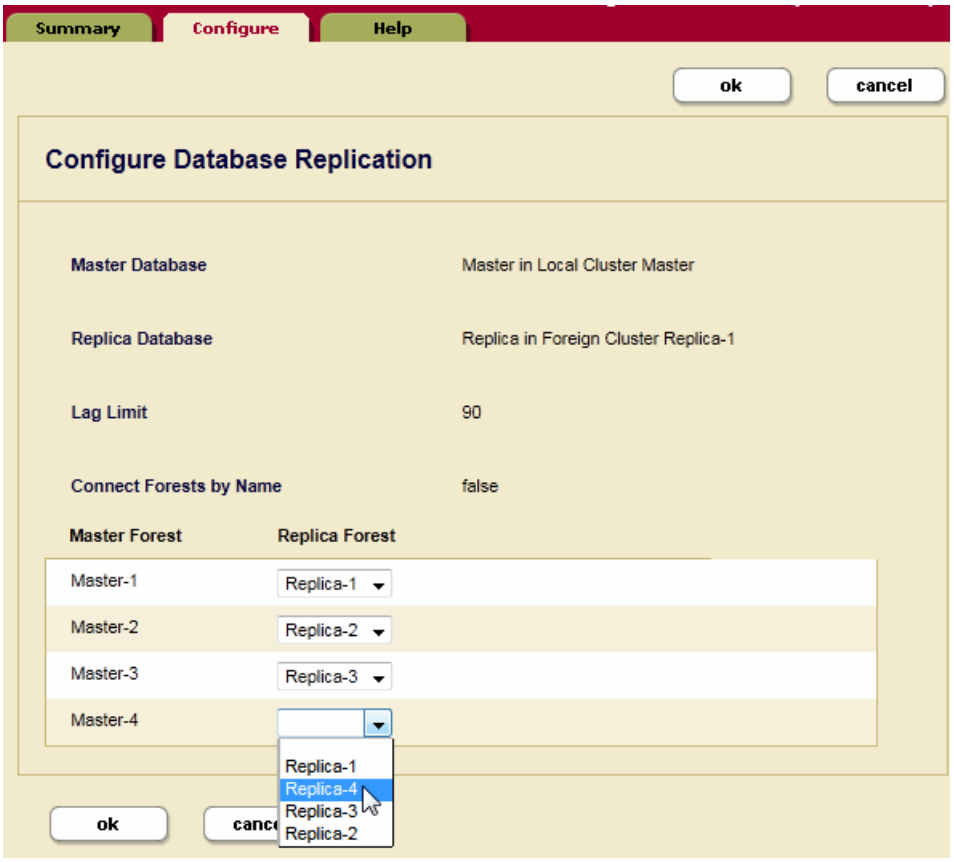
3. In the Configure Database Replication Step 2 page, select Replica for the Foreign Database.



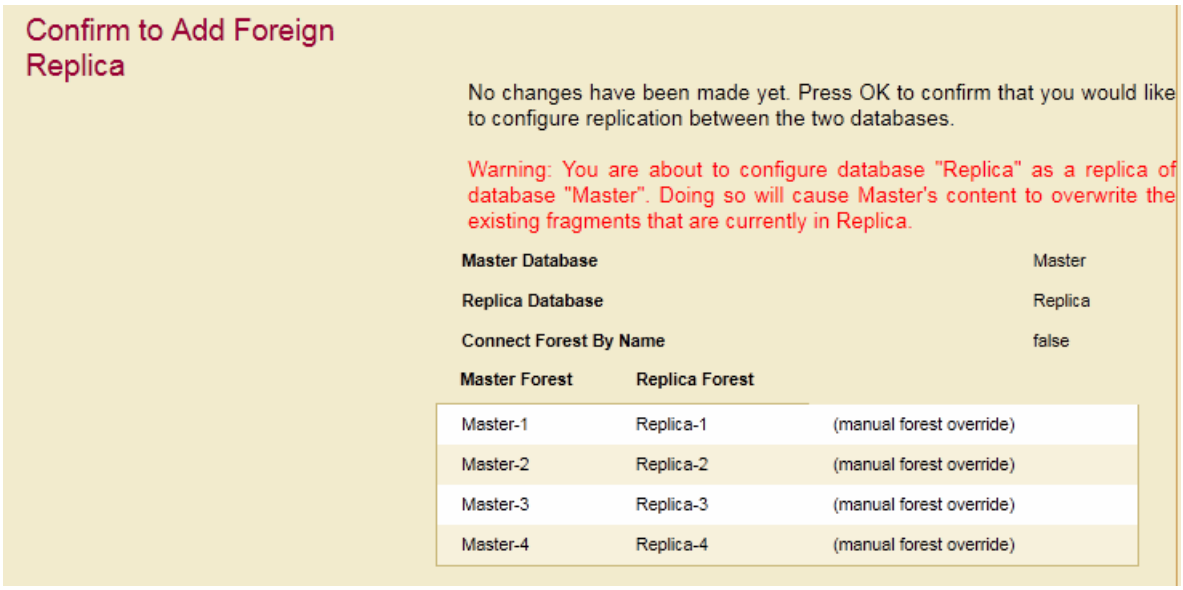
4. Select false for Connect Forests by Name and configure the other settings as described in “Configuring Database Replication” on page 32.



5. In the Connect Forests by Name table, use the pull-down menus to connect the Replica forests to the local Master forests. Click OK.



6. In the Confirm to Add Foreign Replica page, confirm the configuration and click OK.



3.5 Enabling, Disabling, Suspending, and Resuming Database Replication

At the top of the Database Replication Summary page are buttons to enable, disable, suspend and resume database replication.



Button	Description
enable	Enable database replication. This changes the database replication configuration so that database replication will remain enabled after restart or failover.
disable	Disable database replication. This changes the database replication configuration so that database replication will remain disabled after restart or failover.
suspend	Suspend database replication. This does not change the database replication configuration, so database replication will resume after restart or failover.
resume	Resume database replication. This does not change the database replication configuration.

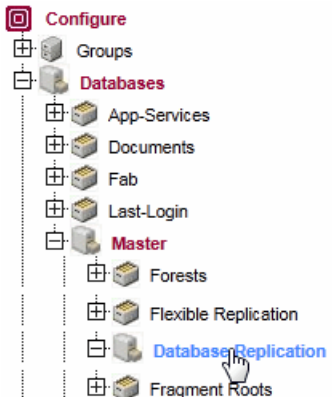
Note: The index settings on the master database are used on the replica for as long as replication is enabled. When replication is disabled, the replica's original index settings are reinstated. When replication is enabled, queries that rely on the original replica index settings will fail.

Note: Database replication may be suspended internally for a short period of time when rebalancing of documents occurs between two forests.

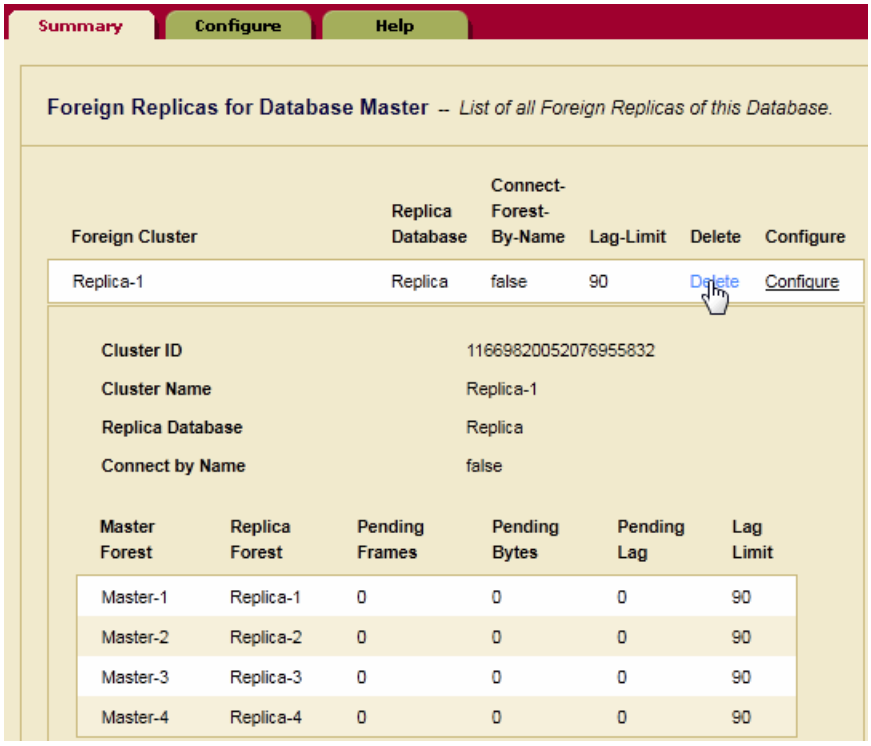
3.6 Deleting a Database Replication Configuration

You can delete a Database Replication configuration from a bootstrap host on either the Master or Replica cluster. The procedure described in this section describes how to delete a Database Replication configuration on the Master cluster.

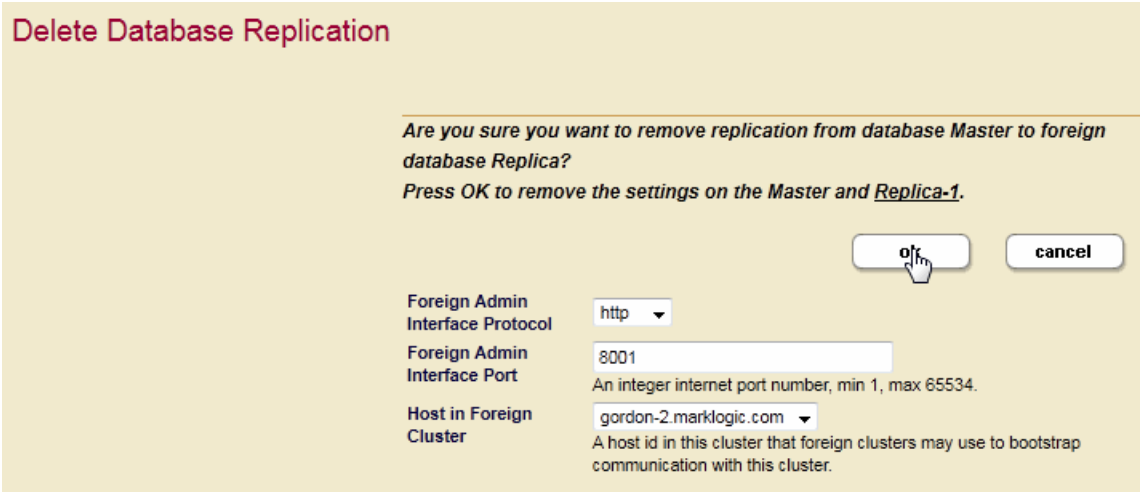
1. On the bootstrap host in the Master cluster, navigate to the Master database in the left-hand menu and select Database Replication:



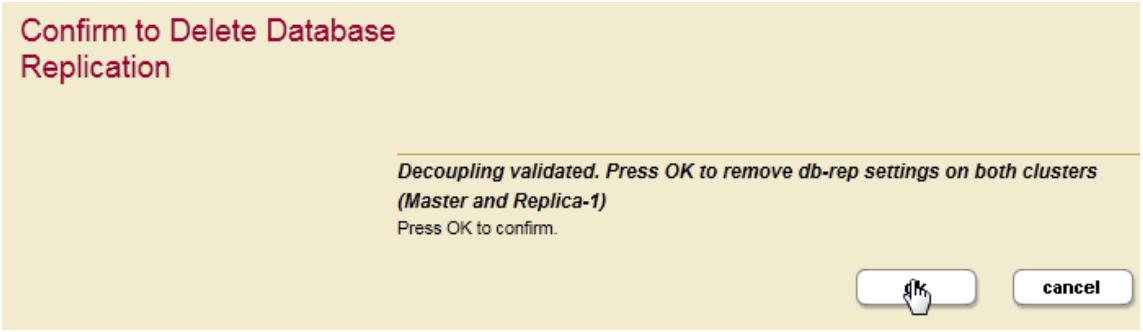
2. In the Foreign Replicas for Database section of the Summary page, click Delete.



3. In the Delete Database Replication page, select https as the UI Protocol if SSL is enabled on the Admin App Server on the bootstrap host in the Replica cluster. Click OK.



4. In the Confirm to Delete Database Replication page, click OK.



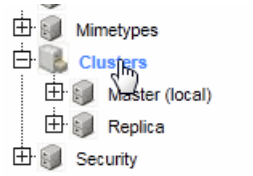
3.7 Decoupling the Local and Foreign Clusters

This section describes how to decouple two clusters.

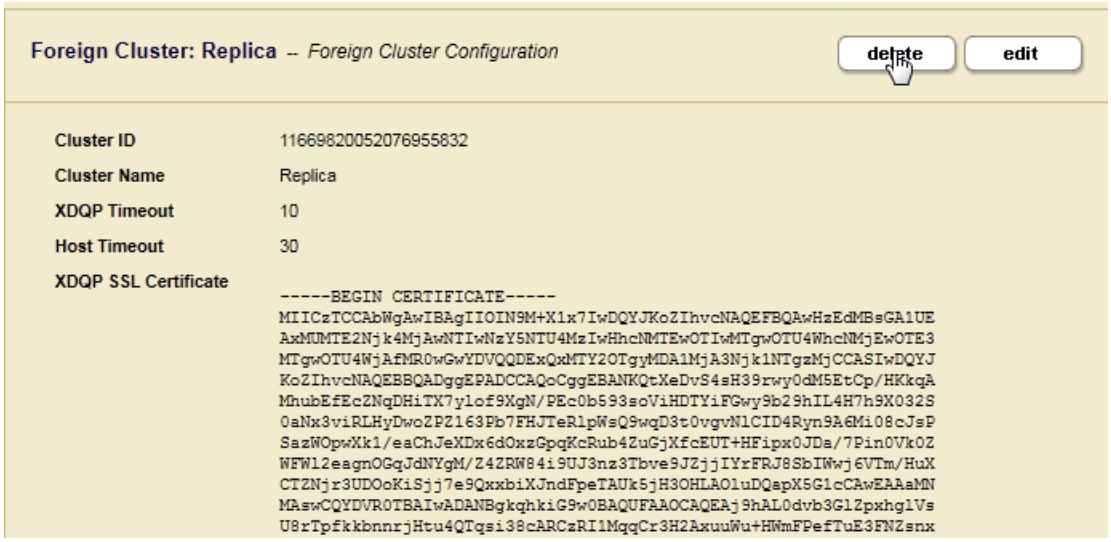
> **Note:** Before decoupling clusters, you must remove any Database Replication configurations between the local cluster and the cluster to be decoupled.

For example, to decouple a cluster named Replica from the local cluster, do the following:

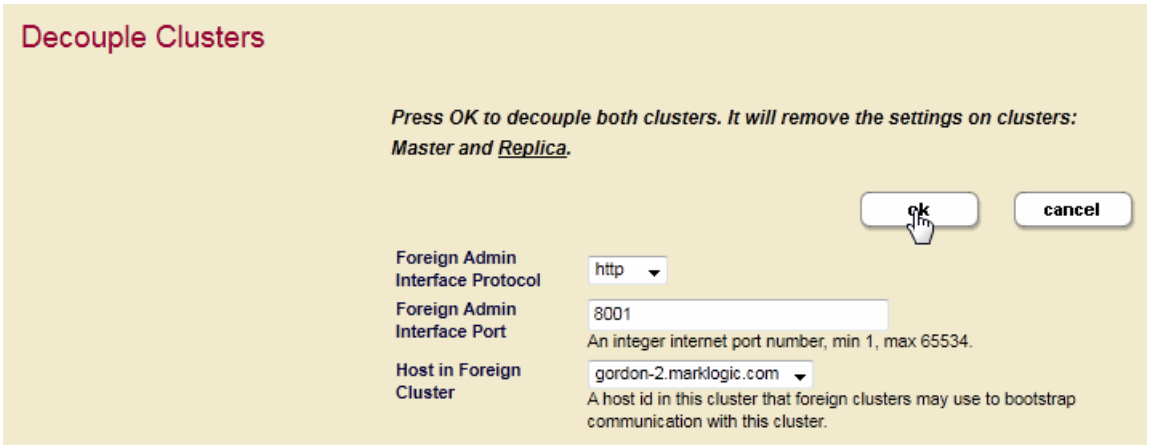
1. On the local host, select Clusters at the bottom of the left-hand menu:



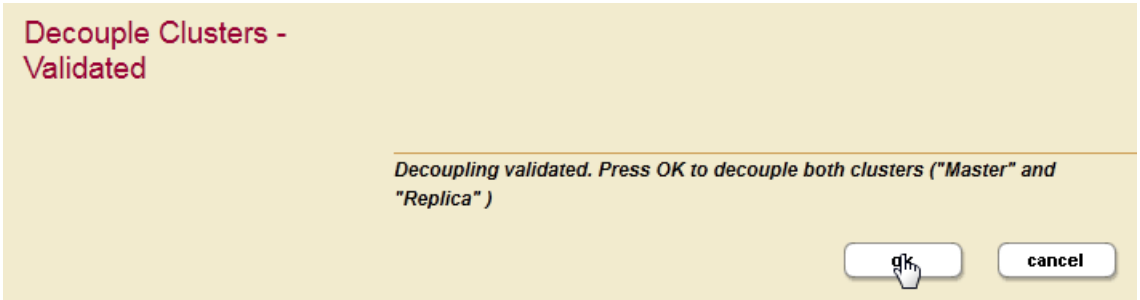
2. Locate the foreign cluster to be decoupled and click Delete:



3. Click OK in the Decouple Clusters page:



4. Click OK in the Decouple Clusters- Validated page:



3.8 Configuring App Servers on the Replica Cluster

As described in Reducing Blocking with Multi-Version Concurrency Control in the *Application Developer's Guide*, setting multi version concurrency control to 'nonblocking' on an App Server minimizes transaction blocking, which is useful if the App Server makes use of a replica database that significantly lags its master database.

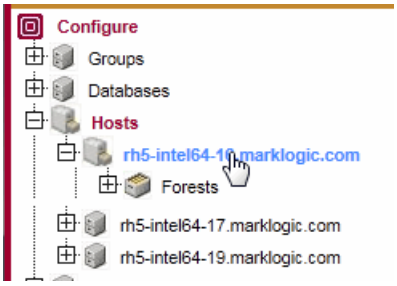
It is recommended that, at a minimum, you set multi version concurrency control to 'nonblocking' on the Admin App Server because the Security database is typically not frequently updated. The 'nonblocking' multi version concurrency control option minimizes transaction blocking at the cost of queries potentially seeing a less timely view of the database, so you should weigh these two factors when determining whether to set this options on other App Servers on your replica cluster.

3.9 Changing the Foreign Bind Port

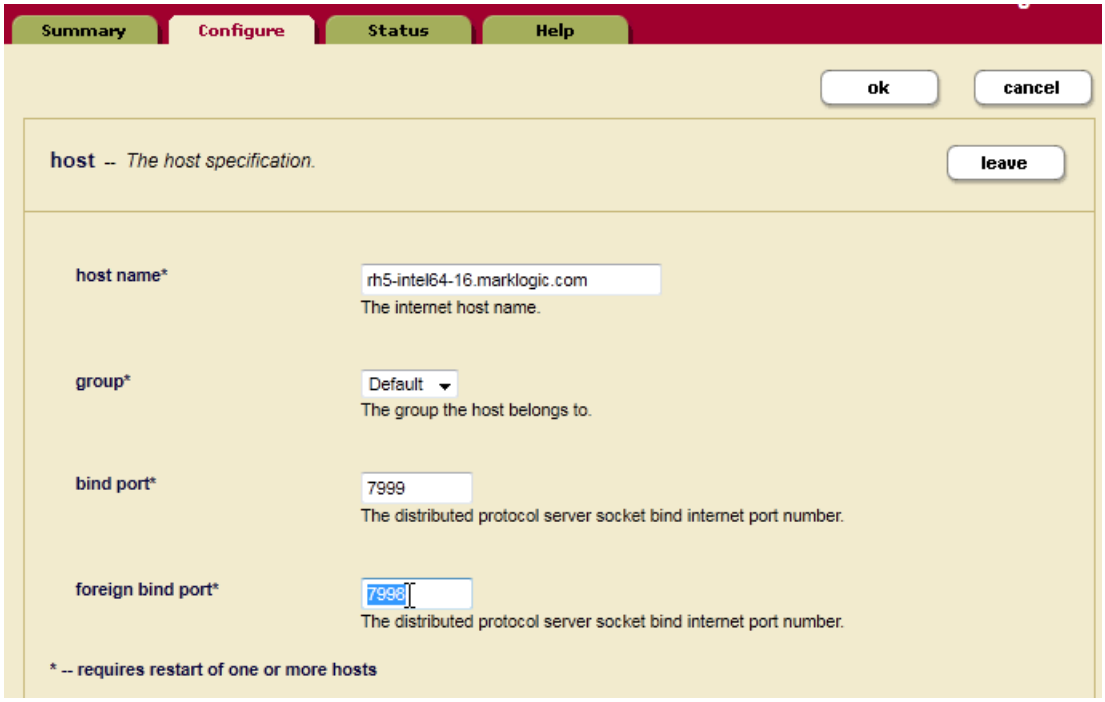
As described in "Inter-cluster Communication" on page 10, communication between clusters is done using the intra-cluster XDQP protocol on the foreign bind port. By default, the foreign bind port is port 7998. This section describes how to change the foreign bind port.

> **Note:** The procedure in this section must be repeated for each host in the cluster that is involved in inter-cluster replication.

1. Select a host in the local cluster under Hosts in the left-hand menu:



2. In the Host Configuration page, change the port number in the foreign bind port field:



3.10 TCP Tuning For High-Latency Environments

On Linux systems, if you have configured database replication where there is high-latency between the master and the replica environments (for example, if your master is in San Francisco and your replica is in Tokyo), you might need to tune the Linux TCP settings to increase throughput. The following are examples of the tuned Linux TCP setting (these settings are tuned with values greater than the typical defaults):

```
# sysctl -w net.core.rmem_max=8388608
net.core.rmem_max = 8388608
# sysctl -w net.core.wmem_max=8388608
net.core.wmem_max = 8388608
# sysctl -w net.ipv4.tcp_mem='8388608 8388608 8388608'
net.ipv4.tcp_mem = 8388608 8388608 8388608
# sysctl -w net.ipv4.tcp_rmem='4096 87380 8388608'
net.ipv4.tcp_rmem = 4096 87380 8388608
# sysctl -w net.ipv4.tcp_wmem='4096 87380 8388608'
net.ipv4.tcp_wmem = 4096 87380 8388608
# sysctl -w net.ipv4.route.flush=1
net.ipv4.route.flush = 1
```

To see your current TCP settings, run the following Unix command:

```
# sysctl -a | grep mem
```

The preceding setting will change the running system but will not survive a reboot. Once you have tuned your system to your satisfaction, you need to add these settings to your startup environment to persist them through system reboots.

For details on your TCP settings, see your operating system documentation. If you have questions about how these operating system parameters might behave with your MarkLogic environment and you have an active maintenance contract, you can contact MarkLogic Technical Support for help.

4.0 Checking Database Replication Status

This chapter describes how to check replication status for a Master cluster. This chapter includes the following sections:

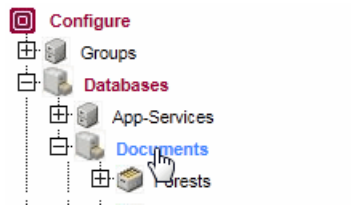
This chapter includes the following sections:

- Checking the State of the Replica Forests
- Checking the Journal Frames Received from the Master
- Checking the Update Lag on the Master

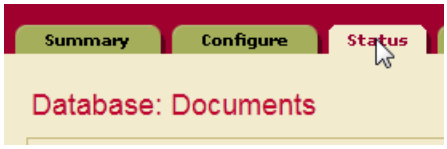
4.1 Checking the State of the Replica Forests

This section describes how to check the state of each Replica forest on the Replica cluster.

1. Access the Admin Interface on the bootstrap host in the Replica cluster. Navigate to the Replica database in the left-hand menu:



2. Select the Status tab:



3. The current state of each Replica forest appears near the top of the page.

Forest	Host	State	Documents	Fragments	Deleted Fragments	Stands	Size	Free Space
Documents	gordon-2.marklogic.com	open replica	51,456	101,556	2	3	50 MB	7,917 MB
Documents-1	gordon-2.marklogic.com	open replica	75,319	150,638	0	2	48 MB	7,917 MB
Documents-2	gordon-2.marklogic.com	open replica	53,867	105,296	2	3	51 MB	7,917 MB
Documents-3	gordon-2.marklogic.com	open replica	52,747	103,822	2	3	51 MB	7,917 MB
		Total	**233,389**	**461,312**	**6**	**11**	**200 MB**	

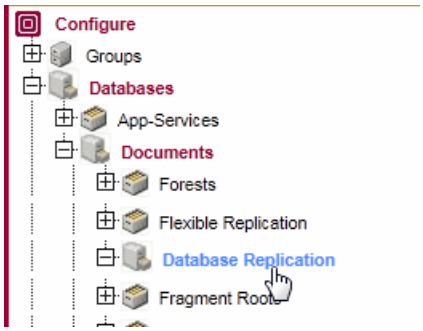
The possible states related to Database Replication are listed in the table below.

State	Description
Open Replica	The Replica forest is currently receiving replicated data in the form of journal frames from the Master forest. This is the normal state for Database Replication.
Syncing Replica	The Replica forest is "bulk synchronizing" with the Master forest. This occurs when a Master database containing documents is initially configured for Database Replication, after the Master and Replica have been detached for a sufficiently long period of time that journal replay is no longer possible, or following a local-disk failover. Once the Master and Replica databases are synchronized, the state is reset to Open Replica. **Note:** While in this state, the status page will report that the database forests are in an error state. There is nothing wrong with the forests and they will appear as normal when the database returns to the Open Replica state.

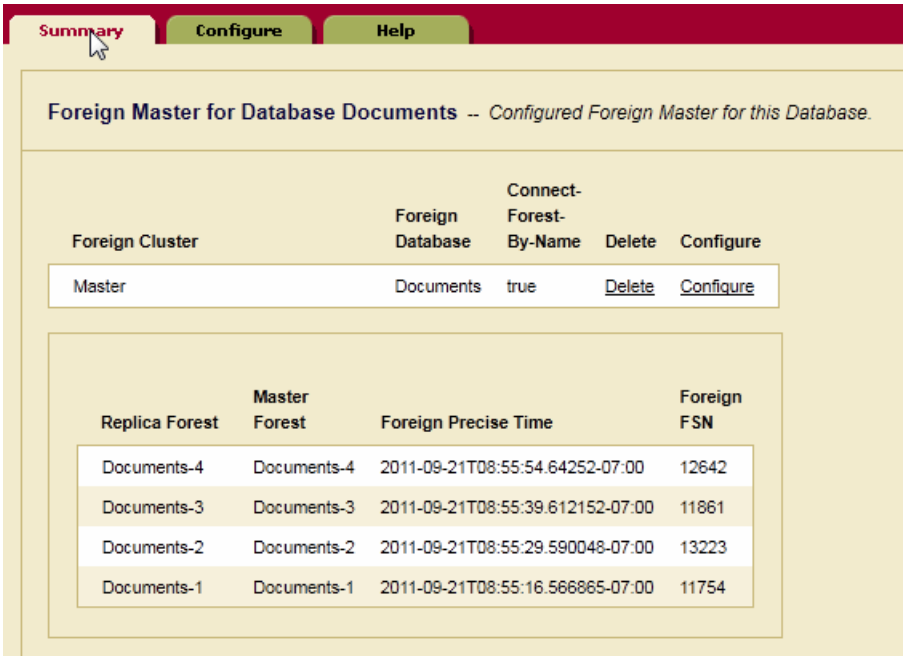
4.2 Checking the Journal Frames Received from the Master

This section describes how to check the current journal frames received by each Replica forest from the Master.

1. On the bootstrap host in the Replica cluster, navigate to the Replica database in the left-hand menu and select Database Replication:



2. Select the Summary tab:



Field	Description
Foreign Precise Time	The point in time the foreign Master forest asserted itself as a Master. If local-disk failover isn't configured on the Master, this is the time the forest was created, last cleared, last rolled back, and so on. If the foreign Master has local-disk failover configured, in addition to the previously mentioned events, this could be the time of the last failover.
Foreign FSN	The ID of the last journal frame received from the foreign Master. The ID is simply the current count of journal frames.

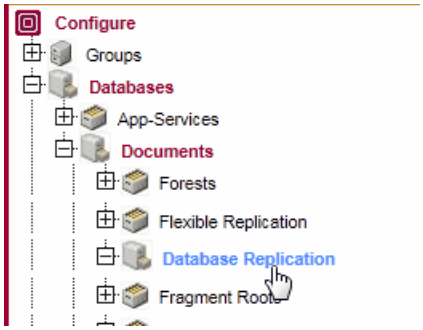
4.3 Checking the Update Lag on the Master

As described in “Replication Lag” on page 10, the Replica cluster sends an acknowledgement to the Master cluster when a replicated journal frame has been received and stored. The Lag Limit you set in your Master cluster configuration specifies that transactions on the Master will be stalled if the Master does not receive an acknowledgement from the Replica within the number of seconds specified by the Lag Limit. If the Pending-Lag value exceeds the Lag-Limit, new transactions that access the Master database are stalled.

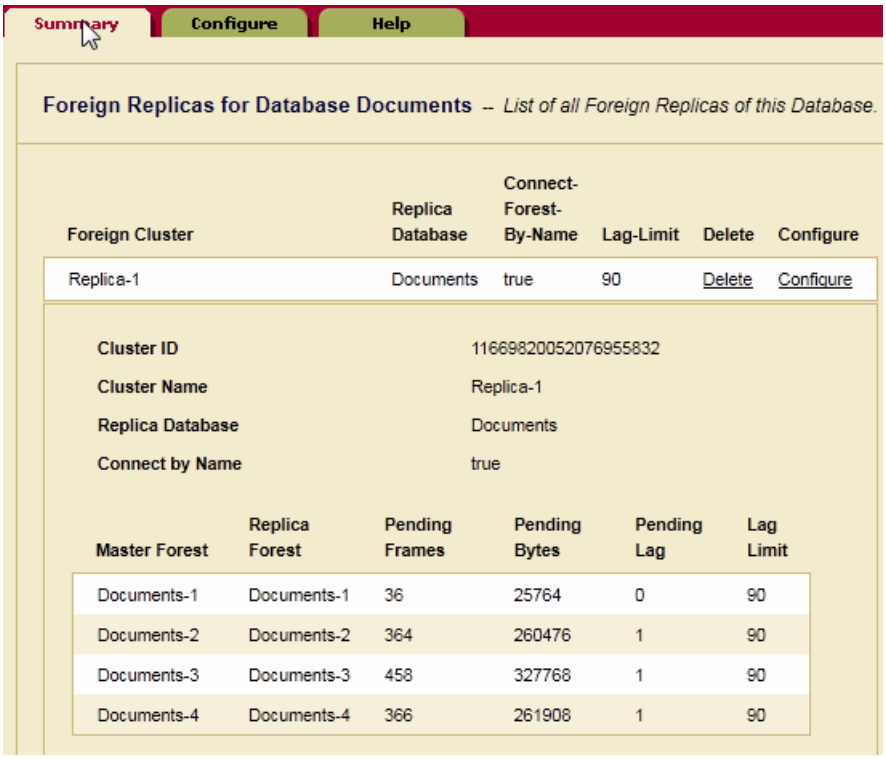
If the XDQP timeout value set for the Replica cluster is exceeded, the Replica cluster is assumed to have failed. In this event, the Master cluster will detach from the Replica and transactions will continue normally on the Master. When the Replica cluster is available again, Database Replication will resume and the Replica database will resynchronize with the Master database.

You can check for lagging updates at the bottom of the database status page. For example, to check the Database Replication status of the Documents database do the following:

1. On the bootstrap host in the Master cluster, navigate to the Master database in the left-hand menu and select Database Replication:



2. Select the Summary tab:



Field	Description
Local-Forest-Name	The name of the Master forest.
Foreign-Forest-Name	The name of the Replica forest.
Foreign-Database-Name	The name of the Replica database.
Pending-Frames	The number of unreplicated journal frames.
Pending-Bytes	The number of unreplicated bytes.
Pending-Lag	The number of seconds since the last journal frame was replicated.
Lag-Limit	The current lag limit set for this database. For details on the lag limit, see “Replication Lag” on page 10.

5.0 Technical Support

MarkLogic provides technical support according to the terms detailed in your Software License Agreement or End User License Agreement.

We invite you to visit our support website at http://help.marklogic.com to access information on known and fixed issues, knowledge base articles, and more. For licensed customers with an active maintenance contract, see the Support Handbook for instructions on registering support contacts and on working with the MarkLogic Technical Support team.

Complete product documentation, the latest product release downloads, and other useful information is available for all developers at http://developer.marklogic.com. For technical questions, we encourage you to ask your question on Stack Overflow.

6.0 Copyright

MarkLogic Server 9.0 and supporting products.
Last updated: November 18, 2019

COPYRIGHT

© 2019 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, US 8,892,599, and US 8,935,267.

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.

For all copyright notices, including third-party copyright notices, see the Combined Product Notices for your version of MarkLogic.