

Tamino

Replication Guide

Version 10.1

April 2018

This document applies to Tamino Version 10.1 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 1999-2018 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA, Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Use, reproduction, transfer, publication or disclosure is prohibited except as specifically provided for in your License Agreement with Software AG.

Document ID: INS-REPLICATION-101-20180413

Table of Contents

1 Replication Guide	1
How does Replication Work?	2
What are the Goals of Replication?	4
How to Create a Replication Database	6
Administration of a Replication Database	9
Restrictions	11
Parameter Setting for Replication	12
Error Situations	12
Index	13

1 Replication Guide

■ How does Replication Work?	2
■ What are the Goals of Replication?	4
■ How to Create a Replication Database	6
■ Administration of a Replication Database	9
■ Restrictions	11
■ Parameter Setting for Replication	12
■ Error Situations	12

Tamino users often need to maintain duplicates of a database on remote servers, for a number of reasons. Perhaps the most common reason is to have a “hot stand-by” system with up-to-date copies of all transactions available for applications to work with in case the primary system fails. Unlike a conventional backup that must be restored from disk, tape or CD, the replicated database is available to applications as soon as they can be pointed to it.

This chapter describes the basic principles and approaches that can be used for replication in Tamino. The following topics are covered:

How does Replication Work?

What are the Goals of Replication?

How to Create a Replication Database

Administration of a Replication Database

Restrictions

Parameter Setting for Replication

Error Situations

How does Replication Work?

In Tamino, replication is an automated process. The current state of a Tamino database can be rebuilt with a backup, which is a snapshot of the database contents at a specified point in time, and by applying the subsequent *log spaces*.

This means that replication is based on log spaces. Updates to Tamino databases (insertion, modification and deletion of both schema definitions and documents) are logged to log space files. These updates are then replicated on a foreign database by requesting the log space data from the original (master) database and applying them to the replication database. The ability to transfer log data offers a good alternative to copying the whole database repeatedly.

The prerequisites for re-applying the master database changes to the replication database are:

- The master database node must be accessible within the network;
- If the master database and the replication database are on different nodes, these must be configured to use the same XTS directory server;
- The master database must be active;
- The log space data must be accessible (to the master database).

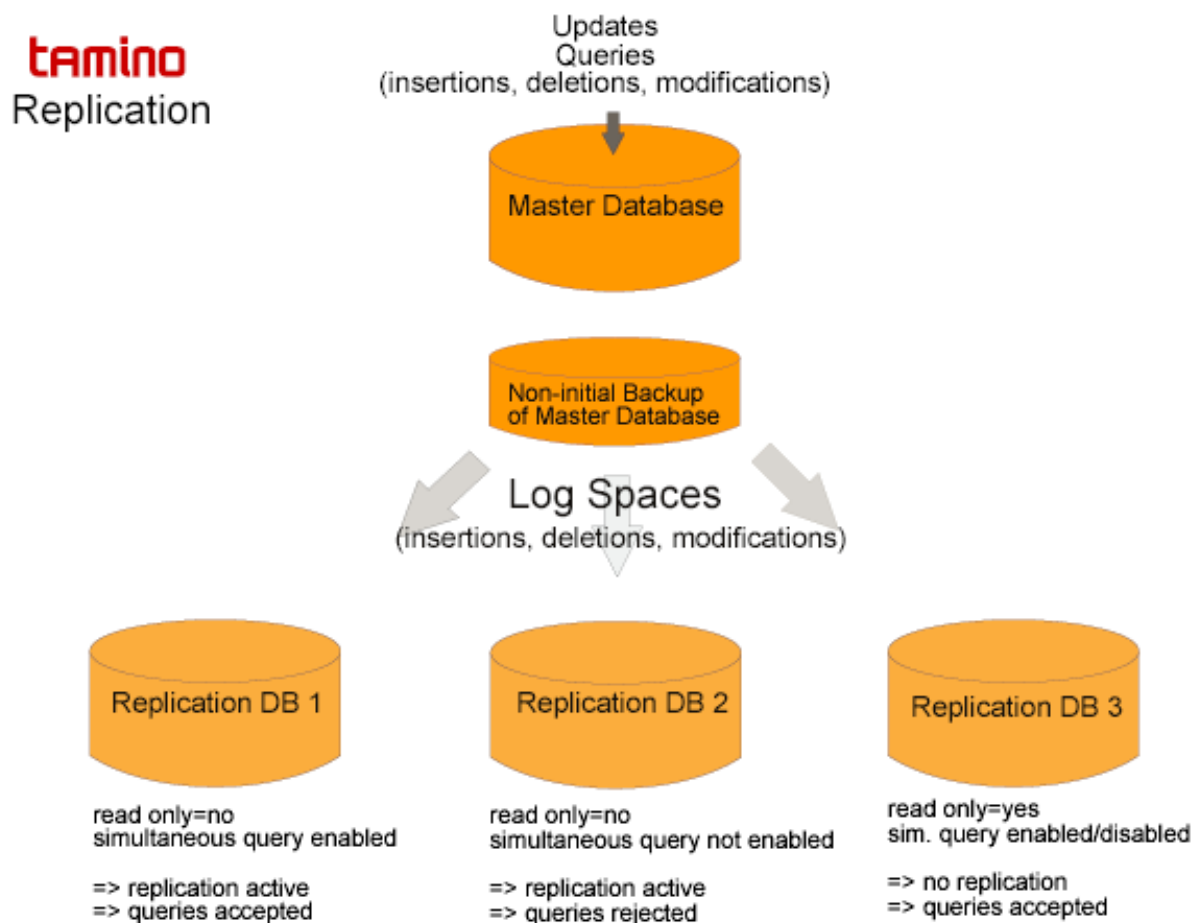
In principle, a replication is a never-ending restore/recover process from a backup, interrupted only by server shutdowns. In Tamino, replication is always *asymmetric* (a one-sided master/slave relationship) and *asynchronous* (to avoid conflicting updates, the replication database reflects

changes in the data with a certain time delay only). This has the advantage that replication works even if the replication database is not permanently online.

Tamino supports simultaneous queries on a replication database while it is replicating. Simultaneous query during replication uses a locking approach that maintains the consistency of queried data. It is not necessary to shut down replication databases and restart them in read-only mode. This capability enables Tamino to run in load-balancing applications where data in replication databases is still accessible during the replication from a master database.

Alternatively, the replication database can also be set to run without the simultaneous query option. In this case, the replication database still automatically requests log space data for replication, but it cannot be accessed while it is replicating updates from the master database (the database parameter `read only` is set to `no`). To have read (and query) access, the replication database needs to be stopped and then restarted in read-only mode (the database parameter `read only` needs to be set to `yes`). In this read-only mode, the replication database does *not* request log space data for replication, but guarantees read access.

The following diagram gives an overview of replication in Tamino:



What are the Goals of Replication?

Replication can be used for a number of cases, for example to provide a hot stand-by database that can be used in the case of a failure of the master database, in a mobile computing environment, or to enable load balancing for queries and updates. The following section describes Tamino replication usage cases to help you decide how to set up your own replication scenario:

- [Hot Stand-By](#)
- [Mobile Work](#)
- [Load Balancing](#)
- [High Availability](#)

Hot Stand-By

Tamino replication provides a reliable way to set up a hot stand-by database in case the master database fails. Simply set up a replication database without simultaneous query (for a detailed description of the steps to be accomplished, see section [How to Create a Replication Database](#)). If your master database fails, for example due to a computer crash or disk failure, just reset your replication database to a normal database and have your applications run against it. The easiest way to redirect your applications to this database is to rename it to the name of the original database. Although in this case a few transaction that occurred just before the master database crashed might be lost due to the replication delay, you have no restore time and can quickly continue normal database operations.

Mobile Work

For many applications, a database does not need to have the most recent status. Often it is sufficient to have a defined status, for example that of last night. Consider the following case: A sales person takes a copy or parts of a company's master database (for example a parts list) to a customer visit. After the visit, he or she updates the local copy as a result of the customer contact, and later updates the company's master database when returning to the office.

If you have client applications for which access to such a defined state of the database is acceptable, you can set up one or more replication databases for these clients. In the example above, you can set up a replication database on each sales person's mobile device. If you run this database in replication mode (with or without simultaneous query), it will synchronize to the most recent state when connected to the company network. The database can then be switched to read-only mode to keep this state, and queries can be executed (for example during the customer visit). When later switching back to replication mode, the replication database is automatically synchronized with the master database.

Note that in principle, you can run the database in replication mode with simultaneous query being switched on all the time. If the connection to the master database is lost, read access is still allowed, but replication activities are interrupted. Replication is automatically resumed when the

connection to the master database is re-established. However, some documents that were being updated shortly before the interruption occurred might still be locked, as the transaction on the master database has not been finished. These documents do not have read access.

Load Balancing

Another good reason for replication is the possibility of load balancing. Tamino automatically balances the load between the CPUs of a multiprocessor computer. However, it is not possible to have Tamino servers running on different computers accessing the same databases. In this sense, replication is not an option for load balancing - the reason for this being that Tamino caches a lot of data to improve performance, and these caches would have to be synchronized for each access. This would lead to a performance loss. To avoid performance losses, consider the following solutions to achieve load balancing:

- With replication: **Separating read-only applications from modifying applications**;
- Without replication: **Vertical partitioning of the database**, or **multi-tier load balancing**.

Separating Read-Only Applications from Modifying Applications

Consider the case of a flight booking system: queries for existing flights, available seats or departure times are frequent, but real reservations are less frequent. Reservations must not be withdrawn after confirmation. Queries can be directed to read-only databases, whereas reservations are directed to update databases.

Another usage scenario exists for product catalogs: data is rarely updated, but frequently requested by a large number of users. The database must always be available and searchable and have a high fault tolerance. This can be accomplished by having multiple read-only servers containing replicas of the original data, and one master database where updates are made.

In both cases, Tamino's ability to provide read access to replication databases is used for load balancing purposes. Applications that need read access only and can accept a slightly outdated state of the database (typically a few seconds) can be directed to a replication database. Note that only the master database can be updated. It is not advisable to direct some requests from an application to the master database and some to its replication databases, as this might interfere with Tamino's concept of sessions. Also, the data in the master database and the replication database might be slightly different, due to the replication delay. This may lead to inconsistent results if both are combined within a single application.

Vertical Partitioning of a Database

One possibility for load balancing without replication is vertical partitioning of a database. Doctypes and collections that are not addressed together in a single query can be distributed over multiple Tamino databases, which in turn can be distributed over multiple computers.

Multi-Tier Load Balancing

Another possibility for load balancing without replication is multi-tier load balancing. Although it is not possible to distribute a single Tamino database over multiple computers, the various components of an application using Tamino can and often should be distributed. Multiple copies of an application accessing Tamino can run on multiple computers, which are different from the computer hosting Tamino. Multiple web servers on multiple computers (which are different from the computer hosting Tamino) can be used to access a Tamino database.

Typically, load balancing at this level - particularly moving applications and web servers away from the Tamino computer - is very efficient.

High Availability

Although replication might be a solution for high-availability failover scenarios, another option is using Tamino in combination with high-availability cluster software. Replication might not be appropriate in some cases; for example, if a failure occurs during the asynchronous propagation of database transactions, it might happen that not all of the transactions have been replicated at that time. If a database with some replication delay is not acceptable in your usage scenario as a substitute for the lost database, replication as a method to achieve high availability is not a choice. Instead, we recommend using a combination of Tamino and high-availability cluster software to build failover solutions. High-availability clusters avoid the problem of continuing with a database as a substitute that may not reflect the most recent update activity. Tamino supplies a high-availability add-on that integrates Tamino with Veritas Cluster Server, HP MC/ServiceGuard and IBM/HACMP. Please contact your software supplier for details.

For further information, see the documentation about *High Availability*.

How to Create a Replication Database

Communication between a master database and a replication database is done via XTS (eXtended Transport Service).

XTS manages communication targets in a directory, which is automatically filled by the Tamino databases at startup.

XTS offers two methods of maintaining the directory:

1. Software AG directory server

A standalone service, included in the distribution as an installable package

2. Flat file implementation

Can be used if all participants are on the local machine. The flat file implementation is used under the condition that an environment variable XTSDIR is defined and contains a valid path pointing to a directory where the flat file is created and maintained by XTS.

Starting with Tamino v8.0 the default mechanism for XTS was changed from the Software AG directory server to the flat file implementation.

That means that for each client accessing a Tamino database the XTSDIR environment variable must be defined and contain the same value as the database server uses (the XTSDIR setting of the Tamino 9.12 databases can be found in the registry under *HKEY_LOCAL_MACHINE\SOFTWARE\Software AG\System Management Hub\Products\Tamino\v912\Environment*).

The script `useXTSDIR.[cmd|sh]` on `$SAGROOT/Tamino/v912/bin` can be used to set the XTSDIR default value (parameter: enable) or to remove the XTSDIR entry in the registry (parameter: disable).

"Client" in this case means the instance addressing the database server via XTS. For example, if Tamino is accessed via an Apache web server then the XTSDIR environment variable must be defined for the Apache web server process.

However, for replication, due to the fact that the databases reside on different nodes, it is necessary to use the Software AG directory server.

Further reading:

Switching between the two XTS directory methods	<i>Installing Tamino XML Server > Complete the Installation</i>
Setting up the Software AG directory server	<i>Setting Up Tamino > Before You Start Using Tamino > Other Issues</i>

There are three steps you need to perform for creating a replication database:

- [Step 1: Create a database from a non-initial backup of the master database](#)
- [Step 2: Add a replication database name to the master database's list of replication databases](#)
- [Step 3: Set up the new database as a replication database](#)

Step 1: Create a database from a non-initial backup of the master database

The first step is to create a new database from a backup of the master database. For detailed information about this first step, refer to the Tamino Manager documentation, section *Database Backups*.



Note: The master database of a replication database is addressed by the database name. At first session startup (or backup), a unique identification pattern is written into the master database. The replication database inherits this pattern, as it is created from a non-initial backup of the master. When setting the type as "replication" and during the replication

process, a check is made to ensure that the patterns are equal. This identification pattern is the reason why the replication database has to be created from a *non-initial* backup of the master database (a backup that has been made *after* an initial backup).



Note: The new database that is created with this step must remain offline, indicated by a red traffic light in the Tamino Manager display. It must not be started before it has been declared as a replication database (step 3). Also, it must have a different name than the master database.

Step 2: Add a replication database name to the master database's list of replication databases

The second step is to tell a (master) database that a replication database will be allowed to replicate the master's updates. To do so, the name of the replication database must be added to the master database's list of permitted replication database names.

➤ To add a replication database name to the list of replication databases

- 1 Select the master database in the Tamino Manager, and choose **Replication Management**.
- 2 From the context menu, choose **Add Replication Database** to add database names to the list of replication databases which are allowed for this database.



Note: The master database must be online, indicated by a green traffic light.

- 3 In the **Add Replication Database** dialog, enter the name of the replication database (the one you have created from backup), and choose **OK**.

A message is displayed, stating that the database was added into the list of replication databases that are allowed for the master database.

You can enter the name of a replication database that has not yet been created. Thus, you can create a “pool” of replication database names that can be used at a later point in time, so that you do not have to repeat this step each time a new replication database is to be created.

If you now choose **Replication Management** in the master database tree, the names of the replication databases allowed for the master are displayed in the right window.

Step 3: Set up the new database as a replication database

The third step is to identify the database which you have created from backup in the first step as a replication database for the master database by specifying the name of its master database. Also, you can choose whether simultaneous query is to be allowed while replication is taking place.

➤ To set up the database created from backup as a replication database:

- 1 Choose **Replication Management** in the tree of the new database that you have created from backup.
- 2 Choose **Set Type to Replication** from the context menu.
- 3 In the resulting dialog, enter the name of the master database. If you want to enable queries to this database while replication is in progress, select the **Simultaneous Query** checkbox.



Note: Once a database has been set to a replication database, the subtree under the database object in question changes and contains less options than a subtree for a normal database. Also, fewer commands are available for a replication database.

Administration of a Replication Database

Once a replication database has been created and started, it now automatically replicates the operations from the master to the replication database. In the following section, you will find further information about the administration of a replication database.

- [Log Spaces and Job Logs](#)
- [Changing the Simultaneous Query Option](#)
- [Resetting a Replication Database](#)
- [Database Administration Functions](#)

Log Spaces and Job Logs

Additional Data in *Log Spaces* of Master Database WITH Simultaneous Query

If the option **Simultaneous Query** has been selected for at least one of its replication databases, additional locking data is transferred in the log spaces by the master database, besides the usual update data. This enables the replication databases to establish the same isolation levels for queries as if the requests were directed to the master database. The master writes such additional data to the log spaces only if at least one replication database is set with simultaneous query enabled.



Note: The amount of additional locking data depends very much on the application. The more documents are modified or locked in protected mode, the higher is the amount of additional locking data.

Additional Messages in the Master Database's *Job Log*

At session start, a message similar to the following is displayed in the information window, stating whether simultaneous query is allowed or not:

```
At least one replication database allows simultaneous query
```

```
No replication databases allow simultaneous query
```

Additional Messages in Replication Database's *Job Log*

A replication database set up with simultaneous query checks at the beginning of processing a new log space whether the log space contains extra locking data. When replication is started (or resumed), a corresponding message is displayed in the job log:

```
Replication of server session 37 from 2008-07-29 13:38:05 started. Simultaneous ↵  
query permitted.
```

If replication is started (or resumed) without simultaneous query being permitted, a warning message is displayed. The latter may be the case if an old log space is used, in which no replication database had been defined with simultaneous query:

```
Replication of server session 38 from 2008-07-29 14:02:53 started. Simultaneous ↵  
query not permitted.
```

Changing the Simultaneous Query Option

If you have set up a new database as a replication database with the simultaneous query option enabled, it is still possible to disable this option. The **Set Type to Replication** option remains active, and you can change the **Simultaneous Query** option any time. The state of the simultaneous query option is displayed in the database information window. To display it, choose the name of the replication database in the Tamino Manager and then the option **Replication Management** in the tree of database options.



Note: The replication database must be offline when changing the simultaneous query option.

Resetting a Replication Database

It is possible to reset a replication database to a normal database, for example in the case of loss of the master database. Thus a replication database can be used as a substitute for the lost master database.

➤ To reset a replication database:

- 1 Choose **Replication Management** in the tree of the replication database. Then, choose **Set Type to Normal Database** from the context menu.

A window confirms the operation.



Note: The replication database must be offline, indicated by a red traffic light.

2 Choose **OK**.

Processing information is displayed, informing you that the database has been reset to normal type.

The database is no longer a replication database, and cannot be switched back to replication database. In the database tree, all the functions of a normal database are available again.

Database Administration Functions

The following administrative functions for replication are available in the Tamino Manager:

- Display the type of the database;
- Display a list of replication databases allowed for a master database;
- Display the database replication status, including the timestamp of the last replicated modification.

For information about these functions, see section *Replication* in the Tamino Manager documentation.

Restrictions

With the current version of Tamino, the following restrictions have to be taken into account:

- The Tamino backup files are binary files, which may cause problems when transferring backup files between architectures with different byte orders (big-endian versus little-endian).
- Security settings are the same for the master database and the replication database. It is not possible to define different settings for the replication database.
- The function **Set Version** is not possible for existing replication databases. After you have set the version of a master database, you must recreate its replication databases from a backup with the new version of the master database.

Parameter Setting for Replication

The update interval of a replication database is controlled by a server parameter called `replication delay`. It is specified on the master database and applies to all replication databases of that master database.

The default value for the replication delay is set to 5 seconds. The smaller this value, the more up-to-date is the replication database; however, the master database has to handle more frequent requests for new log space data.

Error Situations

The following error situations may occur:

- A restore without a full recover was done for the master database;
- The log space data is no longer available for the master database (for example, if the number of backup generations was set too low).

When replication processing is terminated due to an error, replication cannot continue.

If the master database is lost, a replicated database can be converted to a normal database, and it must be made available with the name of the original database.

In any other case, the only way to get an active replication database again is to delete and re-create the replication database, using a non-initial backup of the master database (to speed up the replication process, it is recommended to choose the latest backup).

For further information on error situations and strategies to save your data, see also the *Backup Guide*.

Index

C

create
 replication database, 6

D

database
 replication, 1
duplicate
 of database, 1

H

high availability, 6

L

load balancing, 5

M

master database, 2
mobile work, 4

Q

queries
 while replicating, 3

R

read access
 for replication, 3
replicate database, 1
replication, 1
 administration of, 9
 asymmetric, 2
 asynchronous, 2
 create, 6
 error situation, 12
 log space, 2
 mobile work, 4
 reset, 9
 restriction, 11
 setting, 12
replication database, 2
reset

 replication database, 10
 replication database to master database, 9
restriction
 replication, 11

S

simultaneous query
 for replication, 3
stand-by database, 4

