

# Data and Schema Migration

This section covers the following migration topics :

- Migration Recommendations
  - How to Perform Schema and Data Migration
  - Other Migration Topics
- 

## Migration Recommendations

Tamino Version 3 is upward compatible with previous Tamino versions (2.2 and 2.3) which means that schema and data migration are not mandatory. However, in order to take full advantage of the very significant functional enhancements and very considerable performance improvements provided with the Tamino Schema Definition Language, Version 3 (TSD 3), migration is necessary and strongly recommended. If, on the other hand, you are satisfied with the current functionality and performance provided with *TSD 2*, you need not perform any migration.

Migration can also be performed at the collection level, that is, one collection at a time, thereby avoiding the need of migrating an entire database.

The following topics discuss the specific migration process. The migration tools described are located in the migration subdirectory of the current Tamino installation directory.



### **Warning:**

**Any backup which was generated by Tamino 2.3 (or any previous version of Tamino cannot be restored using Tamino 3.1. After the migration to Tamino 3.1 has been successfully completed it is strongly recommended to perform a new backup under Tamino 3.1 immediately to secure your migration results.**

## How to Perform Schema and Data Migration

This section contains the following topics:

- Semi-automatic Schema and Data Migration from TSD 2 to TSD 3
- Manual Schema and Data migration from TSD 2 to TSD 3

### **Semi-automatic Schema and Data Migration from TSD 2 to TSD 3**

Tamino 3.1 provides a configurable migration tool for semi-automatic migration from both Tamino 2.2 and Tamino 2.3 to Tamino 3.1. This tool performs the schema conversion and, if necessary, also the data conversion. This tool works in essentially the same way for both Tamino 2.2 and Tamino 2.3. It uses an XML-based configuration file. If you start from a Tamino 2.3 installation some steps in data conversion may eventually be omitted. For more information see the documentation on the Tamino 3.1 Schema and Data Migration Tool.

## Manual Schema and Data migration from TSD 2 to TSD 3

If the semi-automatic migration using the migration tool fails (for example, due to errors in the configuration files), there is also the possibility to perform a manual migration from TSD 2 to TSD 3.

The manual migration process is described under the following headings:

### Schema Migration Overview

The following points must be borne in mind when migrating schemas to Tamino 3.1:

- The schema migration is relevant for users of all versions of Tamino 2.
- There is no explicit backward migration (i.e. from Tamino 3.1 to any release of Tamino 2).
- The schema migration utility is available on Microsoft Windows operating systems; Tamino schemas can be migrated on the same machine or across machines. The migration can run on one machine, with data being migrated from a second to a third machine.
- The migration utility cannot be used to port Tamino databases from Windows NT to other operating environments like for instance Solaris.
- The migration utility cannot be used to migrate any applications you may have written that use Tamino data at your site. You must inspect these applications and adapt them if and as appropriate.

### Prerequisites

The following is required for a successful schema migration with optimal results.

- Either the collection contains only TSD 2 doctypes or TSD 3 schemas (there is no mixed state between TSD 2 and TSD 3 allowed).

### Running the Manual Schema Migration

Before you execute the migration, Tamino 3.1 must be installed properly and the *Set Version* command must have been executed successfully within the *Tamino Manager*.

#### To run the migration

Set the *ignoreUpdate* option for all *X-Node* (including *X-Tension*) top level elements .

1. Transform complete TSD 2 schemas from one collection *collname* into one TSD 3 schema using the Tamino schema editor.
2. Check the generated Tamino version 3.1 schema. Manually adjust it, if necessary.
3. Rename the old collection *collname* to *collnameOld*
4. Define a TSD 3 schema for the new collection *collname*

When you have completed these steps you may continue with the following actions as the migrated schemas are now defined in the Tamino 3.1 XML database :

1. Migrate your XML data doctype-wise as described in the next section.
2. Verify the correctness of both your migrated schemas and data.
3. Undo the change of the *ignoreUpdate* option for all X-Node (including X-Tension) top level elements.
4. If you are sure that the migration has been performed correctly, delete the renamed collection *collnameOld*.

For more detailed information on schema migration also refer to 'Migration Strategies for the Manual Migration Process' and 'Migration from Old to New Schema Language'.

## XML Data Migration

This section describes the migration of user data from a Tamino 2.3 database to a Tamino 3.1 database:

### Migration Overview

- The migration of large volumes of data must be planned thoroughly. In particular, please note that the building of indexes in Tamino can be time-consuming, especially if text indexing is performed.
- For reasons of performance, the migration is restricted to one computer. Tamino 2.3 and 3.1 can coexist on the same computer (and also with Tamino 2.3).
- Data is migrated doctype-wise per collection, that is, the migration processes one doctype at a time. All doctypes of one collection are processed before doctypes of the next collection are processed.
- Data is not modified during data migration.

### Running the Migration

Migration is done by unloading data from a version 2.3 database to the file system and loading data from the file system back to a version 3.1 database. Tamino 3.1 provides some command line utilities that are useful for performing the migration. For this purpose, the following tools can be used:

- For unloading the data from Tamino to the file system:
  - The unload option of the *Tamino Data Loader* utility
  - The unload option of the Java loader utility
- For loading the data from the file system to Tamino:
  - The Tamino Data Loader utility
  - The Java loader utility

The procedure described here needs no temporary collection as working storage. The following prerequisites have to be fulfilled:

- The collection to be migrated contains natively mapped data only.

- Tamino version 3.1 has been installed and the Tamino version parameter of the respective database has been set from 2.3 to 3.1 using the *Tamino Manager* (or the **argbatch** tool).
- A backup of the Tamino 2.3 database has been performed successfully.



**Warning:**

**It is strongly recommended to perform a backup of your data prior to starting any migration efforts. If you do not have a correct backup of your Tamino 2.3 database and unforeseen problems arise during the migration process, you may not be able to recover to the state of your database prior to the migration process.**

**▶ To migrate a collection doctype-wise from Tamino 2.3 to Tamino 3.1**

1. Unload the data (doctype-wise into the filesystem):

```
java - cp ".\JavaLoader.jar;.\xerces.jar"
com.softwareag.tamino.db.tools.loader.TaminoLoad
-unload
-u http://localhost/tamino/<database_name>/<collection>
-query /<doctype>
-o c:\tmp\TaminoMigration\...<directory_path>...\<doctype>.xml
- format loadRequest
```

2. \_undefine the data corresponding to the old Tamino *schema* definitions doctype-wise

(remove the old data representation in the corresponding doctype)

3. \_define the new schemas doctype-wise

(the result is a schema of the new *schema* type prepared to contain data in the data representation of Tamino 3.1 and 2.3 but currently containing no data)

4. Load the data (doctype-wise from the filesystem into the appropriate collection):

**Examples:**

1. **With the Java loader:**

```
java - cp ".\JavaLoader.jar;.\xerces.jar"
com.softwareag.tamino.db.tools.loader.TaminoLoad
-u http://localhost/tamino/<database_name>/<collection>
-query /<doctype>
-f c:\tmp\TaminoMigration\...<directory_path>...\<doctype>.xml
- format loadRequest
-
```

2. **With the Tamino Data loader:**

```
inoxmld.exe Database=<server name> Collection=<collection>/<doctype>
Input=<input filename>
```

You now have converted a doctype to the new data storage format. Verify that the migration of the doctype has been performed correctly and proceed in the same way with the next doctype of the collection until all doctypes of the collection are processed.

**Warning:**

**If your schemas use external object references, you have to be careful regarding the order of the `_undefine` and `_define` commands. It is strongly recommended not to use any command execution sequence other than undefining one single schema and immediately defining the same schema again. If you deviate from this execution sequence you may run into problems. For example, trying to undefine an object that references another object that has already been deleted, but not been restored yet, would lead to an error condition in the processing of the `_undefine` command of the referencing object. To avoid this, we strongly recommend that you follow the above mentioned execution sequence of `_undefine` and `_define` commands strictly. Otherwise it may be necessary to be aware of the exact relations of the external objects of your schemas when migrating them to Tamino 3.1 to be able to solve any problems that might arise.**

## Other Migration Topics

This section deals with the following topics:

- Non-XML Data Migration
- SQL Data Migration
- Server Extension Migration
- Application Migration

### Non-XML Data Migration

There is no need for special migration efforts for non-XML data.

### SQL Data Migration

There has been no significant change in the format of Tamino SQL data between Tamino versions 2.3 or Tamino 2.2 and 3.1. No special migration efforts are required for SQL data.

### Server Extension Migration

In general, Tamino server extensions developed for Tamino 2.3 or Tamino 2.2 should also execute under Tamino 3.1. There should be no need for any migration of Tamino server extensions.

If nevertheless there are problems with Tamino server extensions designed for Tamino 2.3 or Tamino 2.2 in a Tamino 3.1 environment, the following procedure is recommended for migrating server extensions:

1. Identify the server extensions to be migrated.
2. Install Tamino 3.1, if not already done.
3. Create a new server extension project.
4. Edit the server extension source.
5. Compile and package the server extension.
6. Install the server extension.

The result of this procedure is that Tamino 2.3 or Tamino 2.2 and 3.1 can work in parallel with their related server extensions. Each version can be managed with its own *Tamino Manager*.

## Application Migration

This section explains under which conditions applications written for Tamino 2.2 or Tamino 2.3 will work when trying to use them with Tamino 3.1. In general you should consider that all Java applications written for Tamino 2.x use the old *Java API* that Tamino 3.1 still supports but later versions may not. Furthermore, the new *Java API* is more stable and offers more functionality to the programmers, so it is a good idea to adapt your applications now to the new Tamino Java API.

### Tamino 2.3

Applications developed for Tamino 2.3 should also run under Tamino 3.1 apart from the following exception (but not vice versa).

Applications developed for Tamino 2.3 that use the API function `getTotalCount` will not run under Tamino 3.1. Change your application so that it no longer uses this function to make it compatible with Tamino 3.1.

### Tamino 2.2

The following restrictions have to be taken into account when migrating applications from Tamino 2.2 or earlier versions to Tamino 3.1:

1. Applications using negative indices in filter expressions (for instance, expressions like `[-1]`) must be adapted in such a way that this does not occur anymore, for example, by combining the position operator with the `last()` function as in

```
/cruise/route[harbor[last()] = "ATHENS"]
```

All other applications which do not use negative indices should work without any modification of their source code.

2. Applications developed for Tamino 2.2 or earlier versions that use the `ino:count` function will not run under Tamino 3.1.
3. Applications developed for Tamino 2.2 or earlier versions that use the API function `getTotalCount` may not run under Tamino 3.1.

