

Xpert.NET Configuration Management

Importing and Exporting Configuration Data in Xpert.NET

Table of Contents

1	Introduction	5
1.1	Purpose of the Document	5
1.2	Addressees of the Document	5
1.3	Remarks on the Content of this Document	5
1.4	Overview of the XCM	5
2	XCM Configuration	6
2.1	Checklist - English not implemented yet	6
2.2	Data Export	6
2.3	Data Import	8
2.4	Partial Import	13
2.5	Partial Import - Technical Background	13
2.6	Errors During the Import	19
2.7	Conflict Resolution and Mapping Entities via Secondary ID	20
2.8	Saved Conflict Resolutions	23
2.9	Checklist - English not implemented yet	24
2.10	Checklist - English not implemented yet	24
3	Dependencies and Transferred Data	25
3.1	Overview of Data not to be Transferred	29
3.2	Automated Event Management	29
3.2.1	Connectors	30
3.2.2	Filter Rules	30
3.2.3	Value Rules	30
3.2.4	Event Ticket Mappings	30
3.3	Configuration Management (CMDB)	30
3.3.1	CMDB Relation Types	30
3.3.2	CMDB Life Cycles	30
3.3.3	CMDB Schemas	30
3.3.4	Import	30
3.3.5	Configuration Management Database	31
3.4	Computer Telephony Integration	31
3.5	Dashboard	31
3.6	Early Warning System	31
3.7	Expense Management	31
3.7.1	Projects	31
3.7.2	Expense Types	31
3.7.3	Allocation Types	31

3.8	Knowledge Management	31
3.8.1	Categories	31
3.8.2	Schemas	31
3.9	Messaging and Collaboration	32
3.9.1	Dynamic Placeholders	32
3.9.2	Information	32
3.9.3	Mail2Ticket	32
3.9.4	Messaging Schemas	33
3.9.5	Templates	33
3.10	Reporting Management	33
3.11	Service Level Management	33
3.11.1	Early Warning System	33
3.11.2	Service Level Management	33
3.12	Service Portfolio Management	33
3.12.1	Service Live Cycles	33
3.12.2	Service Schemata	33
3.12.3	Service Catalogs	33
3.12.4	Service Categories	33
3.12.5	Services	33
3.12.6	Service Transactions	34
3.13	Task Management	34
3.13.1	Task Status	34
3.13.2	Task Schema	34
3.13.3	Task Life Cycles	34
3.14	Ticket Management	34
3.14.1	Ticket Number Ranges	34
3.14.2	Ticket Templates	34
3.14.3	Ticket Filter Management	34
3.14.4	Ticket Status	34
3.14.5	Ticket Conversions	34
3.14.6	Classifications	34
3.14.7	Ticket Wizard	35
3.14.8	Ticket Schema	35
3.14.9	Ticket Actions	35
3.15	User Management	35
3.15.1	Team Management	35
3.16	Workflow Management	35
3.16.1	Workflows	36
3.16.2	Forms	36
3.16.3	Dispatching and Rule Management	36
3.17	DataViews	36
3.18	Search Templates	36
4	Glossary	37

5	Statistics and Change Management	38
5.1	Statistics	38
5.2	Change Control	38

1 Introduction

1.1 Purpose of the Document

Xpert.NET is a very dynamic help desk solution that can be configured in nearly every direction. Daily routine and experience have shown that users get acquainted with the most important modules and configurations quickly. Nevertheless, some questions regarding the functionalities of the *Xpert.NET* modules often remain. Thus, the individual modules cannot be used to their full potential.

1.2 Addressees of the Document

This document only addresses administrators of *Xpert.NET*.

1.3 Remarks on the Content of this Document

This document describes all the functions of the *Xpert.NET* Configuration Manager (XCM) module. Nevertheless, the range of functions may vary due to configuration, licensing, and version. If you miss certain functions listed in this document in your *Xpert.NET* installation, please contact our support directly.

1.4 Overview of the XCM

Via the XCM, which provides all features required for an import and export, all configuration data as well as configuration item dynamic data can be transported between two *Xpert.NET* entities or clients, e.g. a test and a live system. In the process, interdependent data will be recognized and made available in a consistent XML transport archive. This makes sure that configuration data can be exchanged between multiple entities quickly without the need for a lot of personnel and time.

The following overview shows how a change is transferred from a test to a live system via the XCM:

Figure 1.1: Diagram of the transport from a test to a live system

Chapter 2 describes the export of data from a source system as well as the import to a target system. Chapter 3 lists dependencies and offers an overview of the transferred data. Chapter 4 provides a glossary of the most important terms related to the XCM.

2 XCM Configuration

The XCM itself does not require a separate configuration. As soon as it has been activated on your system, it is ready to use.

The export and import options of the *Xpert.NET* Configuration Management can be accessed via the top menu under **TOOLS -> XPERT.NET CONFIGURATION MANAGER**. The two buttons allow for creating and exporting transport packages as well as for importing them to the target system later on.

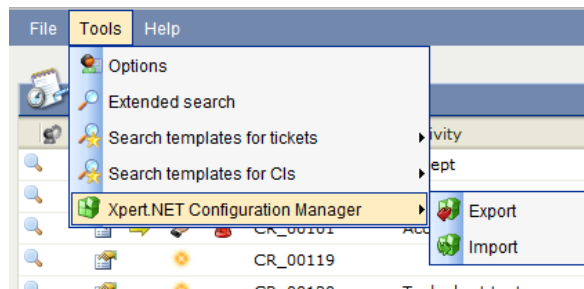


Figure 2.1: Transport packages can be exported and imported via the top menu

Please note

The XCM module is only visible for members of the root administrator group.

2.1 Checklist - English not implemented yet

2.2 Data Export

The menu item *Export* allows for compiling transfer packages and starting exports. In the process, it is possible to export only configuration data or to integrate configuration items (CIs) into the respective transport package as well. However, dynamic data such as tickets generally will not be taken into account during exports.

Configuration of the Export

Clicking on the *Export* button opens the configuration dialog.

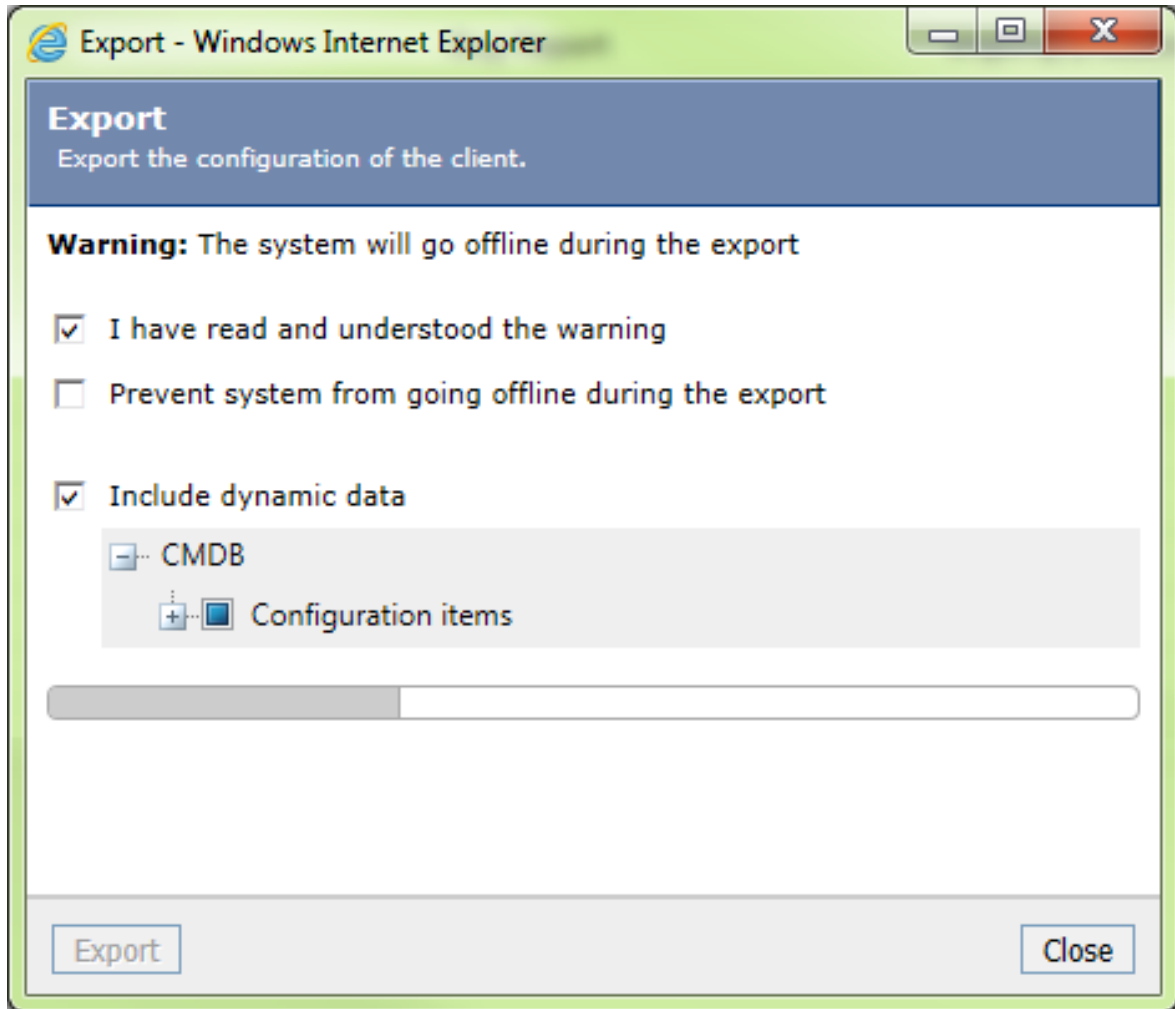


Figure 2.2: **Left:** If transport packages have already been created, the last one created will always be available for download. **Right:** After the export has been completed, the new transport package can be downloaded.

The following two options are available in this dialog:

- **I HAVE READ AND UNDERSTOOD THE WARNING:** This option is a safety measure making sure that that no export is started accidentally.

Please note

The first warning message must be confirmed before the export can be executed!

Please note

A warning message has to be specified in the maintenance overview under *System Shutdown*, otherwise no information will be displayed on the login page when the system shuts down.

- **PREVENT SYSTEM FROM GOING OFFLINE DURING THE EXPORT:** This option allows normal users and administrators to log onto the system and carry on working with *Xpert.NET* normally, even if an export is performed at the moment.

The offline mode should generally not be prevented during an export. Otherwise, the integrity of the transport package cannot be fully guaranteed.

Additionally, CMDB dynamic data (CIs) can be selected for the export.

This particular option for configuration items (**Include dynamic data**) is available because some configurations regard certain CIs as configuration data instead of dynamic data. If CIs are to be exported, a list of selectable CI schemas (including multiple selection) will be displayed. All CIs of the selected schemas will be exported then.

After a click on the *Export* button, all required data will be compiled. Subsequently, the *Export* button will turn into download link, which can be used for downloading the transport package as a ZIP file in order to save it locally.

Warning

Inconsistent systems will not be exported. Here, inconsistent means that there are links to deleted objects or objects removed from the database. When such a faulty system is exported, an error message will appear. The inconsistencies have to be removed before an export can be performed.

In each case, a complete export of all data as well as the selected CMDB dynamic data will be performed. The data to be imported later on (partial or full import) has to be specified during the import.

Please note

It is important that all entities to be transported have a language-dependent name. Otherwise, errors may occur during the import, e.g. whenever a conflict has to be resolved and its resolution is hindered by missing names.

Please note

In order to avoid inconsistent data within the transport package, it is imperative to set the system offline for the export period. If there are still users logged-in on the system during the export process, they will be logged out automatically.

Please note

Never stop services manually on the server/servers as long as *Xpert.NET* is creating the transport package!

Export Failure Due to Unavailable SQL Service

If the SQL service becomes unavailable during the export, no further export can be performed, even after availability has been restored. In order to re-enable the export of the data, the *isonet agent* has to be restarted first. As soon as all workers have been read in by the agent, a new export can be started.

2.3 Data Import

The overview of the XCM import menu can be accessed via the menu item **IMPORT**. The button *Load transport package* allows for importing data previously exported from the source

system to the target system. Via this dialog, a transport package saved on the local computer can be selected and uploaded.

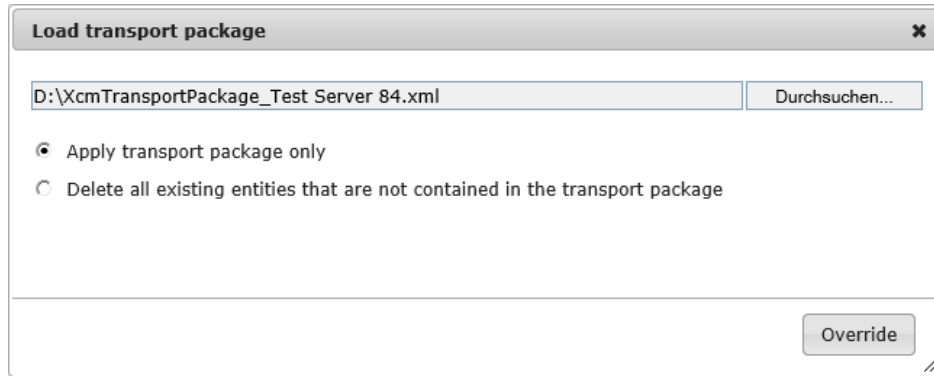


Figure 2.3: The dialog for the data import

Please note

If a package has already been loaded and is displayed, and if a new package is now loaded into the cache, a warning message will be displayed. However, it only refers to the currently displayed transport package. Already imported data will not be overwritten, only the transport package in the cache.

The following options are available here:

- **Apply transport package only:** This is the default option. The package is uploaded to and applied on the server. Data not available on the target system will be created and already existing data, e.g. users, will be updated. This has no impact on other data.
- **Delete all existing entities that are not contained in the transport package:** The package is uploaded to and applied on the server. Data not available on the target system will be created and already existing data, e.g. users, will be updated. All other data will be deleted. This is the recommended option whenever settings and entities are to be imported onto a completely new system, for example.

Please note

Import packages created with an *Xpert.NET* version number differing from the one currently running on the target system (e.g. 3.10.x and 2015.x) or containing syntactical errors will be rejected and cannot be used.

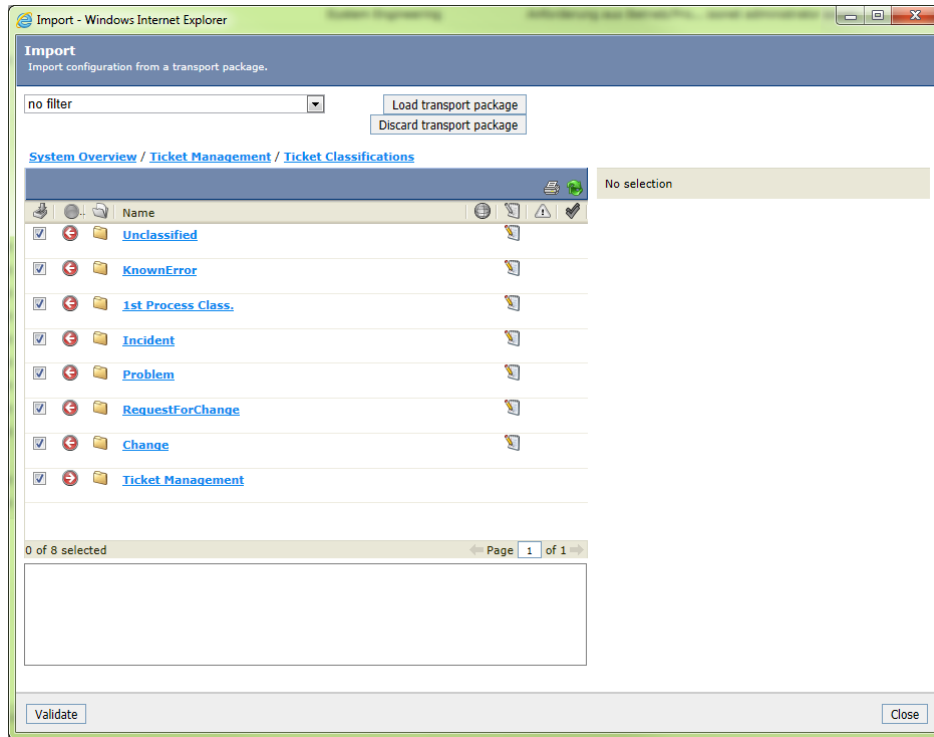


Figure 2.4: After uploading the package (*Browse* button) its content will be displayed. If a transport package has already been uploaded to the target system, the button *Overwrite* will be displayed. In this case, only the old transport package will be overwritten by the new one.

The main entities of *Xpert.NET* are displayed on the left side in the package view. The filter above this view allows for displaying peripheral system content and conflicts individually. If no filter is set, all data to be imported will be displayed.

Peripheral system content are entities that access external data sources and may have to be adjusted, for example. These can be URLs, SQL statements, database connection settings, e-mail addresses, etc.

Warning

Remember to check the peripheral system content accurately! Often other settings are used on test and live servers, for example, as to not put the consistency of data at risk.

Under **Conflicts only**, all conflicts to occur during the import and in need of a manual rework will be displayed as soon as the transport package has been validated.

The view only contains entities from the transport package, including all dependencies, which are displayed below the respective main entities within the hierarchy as well. Every entry in this hierarchy has an option field that is enabled by default. Here, the entities to be imported can be selected before the import.

Please note

Workflow plug-ins, Ticket Wizards, and Ticket Wizard controls cannot be deselected individually. They always belong to their superordinate entities.

Whenever an entity is selected, all entities the selected entity is dependent on will be selected as well. For example, a workflow a selected ticket schema is depending on will be selected for the import automatically.

The checkbox results in the following behavior:

- **Checked:** The entity and all its successors will be imported.
- **Checked, gray background:** The entity will be imported, but not all of its successors.
- **Empty:** The entity will not be imported, and thus no dependency types 1 and 2. Successors with the type 3 may, however, be imported (see Chapter 3).

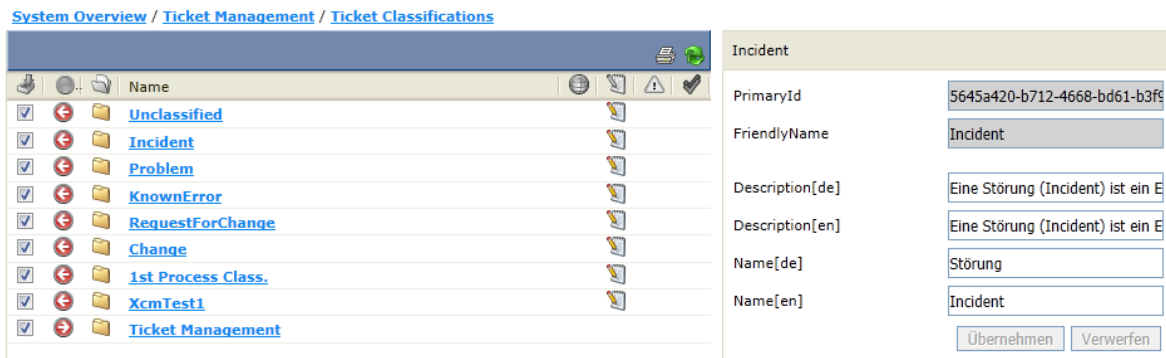


Figure 2.5: The import overview

The columns in the import view have the following meanings:

- 1st column: Enable/disable import for an entity and all its dependencies. Dependency type 3 entities that are not successors as well will not be included here.
- 2nd column: Displays whether another entity depends on the current entity (import activated).
- 3rd column: Displays the entity type.
- 4th column: Displays the name of the respective entity.
- 5th column: Displays whether the entity is relevant for peripheral systems.
- 6th column: Allows for editing an entity (e.g. for editing peripheral system content).
- 7th column: Displays whether there is a conflict for an entity.
- 8th column: Returns the validation result.

As soon as an object has been selected on the left side and the icon for editing has been used, information on the object will be displayed on the right and can be edited if necessary.

Before the actual package import, it is necessary to use the *Validation* button. During the validation, each entity to be imported is checked by the Configuration Manager. It checks whether there is a need to newly create entities or to update/delete existing ones. Conflicts to be solved manually will be displayed simultaneously.

Validation Results

Information on conflicts and validation results can be found in the next to last column of the overview. Information icons will be displayed whenever

- the object does not exist on the target system yet and has to be created first;
- the transport package contains an entity that exists on both systems, but has been marked as deleted on the source system;
- the object already exists on the target system, but will be overwritten/updated by the import;
- there is already another object with the same secondary ID (e.g. the FriendlyName) on the target system. This conflict has to be solved manually. Such conflicts are also displayed in superordinate entities (e.g. in a ticket schema for a ticket status), but only if there is no conflict in the superordinate entity, as conflicts in subordinate entities are only validated after these superordinate conflicts have been resolved.
- a deletion conflict occurs. These conflicts occur whenever the entity to be deleted is still used on the target system. The conflict solution dialog displays all dependencies of this entity and the options *Delete*, *Do not delete*, and *Cancel import* are available here.

Clicking the IMPORT button starts the import process after the confirmation screen.

Please note

An import can only be performed after all occurring conflicts have been solved.

Please note

If there are still users logged into the system during the import, they will be logged out automatically.

Please note

If an administrator logs on during the import, he will be forwarded to the XCM overview stage automatically.

Before importing, a complete backup of the configuration data contained on the system is created in form of an XCM export package, in order to allow for a rollback in case of a failed import.

Please note

In order to avoid inconsistent data on the target system, the system will be set offline during the import. Users still logged into the system will be logged out automatically. In order to display information on the login page during the system shutdown, a warning has to be deposited in the maintenance overview under SYSTEM SHUTDOWN.

Please note

Do not close any services manually on the server/servers while *Xpert.NET* is importing the transport package!

After the import has been completed, the system will be set online automatically and normal operation can be resumed. In the process, the system cache will be updated automatically, in order to implement the changed configuration settings immediately. After the import, a check will be performed while going live whether there have been escalations during the system shutdown and all queued actions will be executed subsequently.

2.4 Partial Import

It is possible to import individual entities onto a new system. For this option, the following distinctive characteristics have to be taken into account: Transport packages always include all entities of the system. If various entities are deselected during the import, because they are not needed on the target system, a manual check of the functionality of the imported entities will become necessary.

In the process, the XCM registers the dependencies of certain entities on other ones before the import and imports them automatically as well.

Example

The import of a ticket schema requires a workflow and an SLM priority.

Furthermore, a dispatching rule is used in the workflow. However, only one condition and parameter value is necessary here, even though multiple parameters and conditions exist.

During the import, the XCM detects existing dependencies necessary for the the ticket schema functioning smoothly. Thus, **only** these entities will be imported as well.

This means that the workflow will be imported including its configuration in this example. Simultaneously, the dispatching rule used will be transferred along with the used parameter and its condition. The other parameters and conditions contained in the dispatching rule will, however, **not** be imported.

The necessary priority deposited in the ticket schema will be imported along with the superordinate expense and the contract as well. However, further expenses deposited in the contract as well as their priorities will **not** be taken into account during the import.

As a consequence, the imported ticket schema will be fully functional on the system. However, ticket schemas or workflows created later on, which may be based on the SLA contract or the dispatching rule imported as well, will most likely not be not be executable.

Warning

If you only execute a partial import, you have check manually whether entities transferred as dependencies are to be imported as fully functional.

2.5 Partial Import - Technical Background

Dependencies

There are five different dependencies between objects to be transported:

I) Existential Dependency

Definition: a is existentially dependent on b, if a cannot exist without b.

Example: A ticket field (a) cannot exist without the corresponding ticket schema (b).

II) Functional Dependency

Definition: a is functionally dependent on b, if a cannot function correctly without b.

Example: A workflow (a) using a certain ticket field (b) cannot function correctly, if this field does not exist. The workflow, however, can theoretically exist without the ticket field.

III) Link

Definition: a is linked to b, if there is a connection from a to b.

Example: A CI linked to another CI.

IV) Partially Existential Dependency

Definition: a is partially existentially dependent on a set of entities B, if a cannot exist without at least one entity from B.

Example: A user is partially existentially dependent on the set of his parent groups. That is, the user can only exist if at least one of his parent groups exists.

V) Default Entity of a Partially Existential Dependency

Definition: In the set of entities B of a partially existential dependency, one entity is emphasized. This entity is called the default entity of the partially existential dependency.

Example: The main group in the User Management. It is an emphasized group among the parent groups of a user.

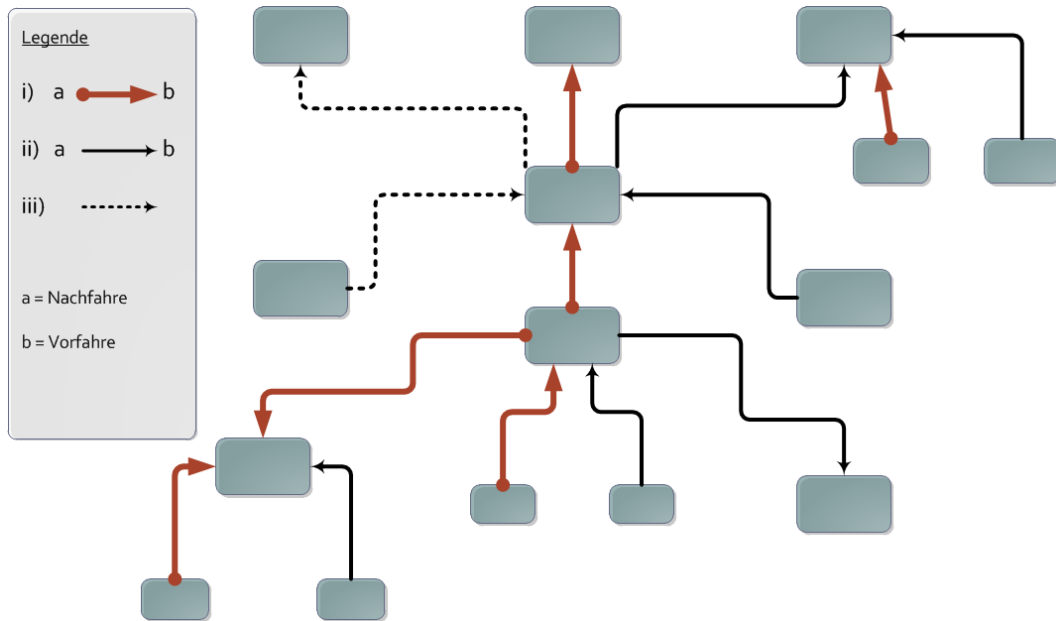


Figure 2.6: Initial situation: This overview contains all dependencies theoretically possible for an entity (parents/successors with the type I-III).

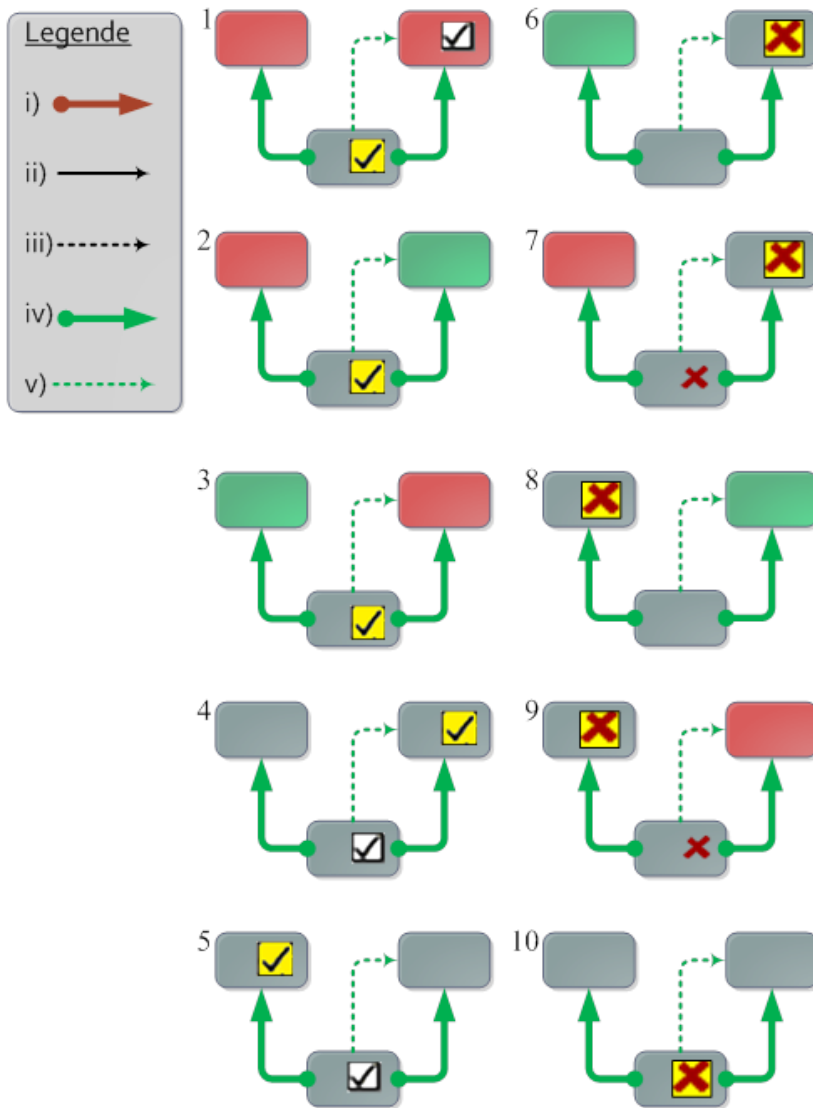


Figure 2.7: Partially existential dependency and default entity of a partially existential dependency (types IV and V). The following applies for the fields: red field = entity deselected, green field = entity selected, gray field = no preference, yellow field = is currently edited manually, rest is automatic

Behavior

The following behavior has been set for these cases (partial dependencies behave analogously to dependencies of the types I-III):

1. Do not allow inconsistent imports as per type I and II.
This is checked during the validation. A package not “complete” in this sense will be classified as invalid and the import will be denied with an error message.
2. Allow for entities linked optionally (link as per type III) not to be imported.

3. The UI forces the compliance to 1. during the selection/deselection of entities by an immediate automatic update of the checkbox of dependent entities in the entire import package.
4. In order to guarantee 3., all successors with the type I will be activated during the activation of the import check box for an entity as well. Likewise, all parents with the type I and II will be activated as well.
Dependencies with the type III will not be activated automatically.
Example: A CI *isosrv13* with the schema *Server* is activated. This CI is assigned to a user *Max Schweizer* (owner) and linked to another CI *Windows Server 2012*.
If the CI is activated, the import for the schema will also be activated (dependency type I). Additionally, the user *Max Schweizer* will be activated as well (dependency type II). However, the import for the CI *Windows Server 2012* will not be activated automatically (dependency type III).
5. In order to ensure 3., successors of the type I and II will be deactivated as well whenever the import for an entity is deactivated. The activation of the imports for successors of the type III will not be changed.
Example analogous to 4.: The import of the schema *Server* is deactivated. Thus, the import will be deactivated for the CI *isosrv13* as well. However, the activation of the import for the user *Max Schweizer* will not be changed.
6. A panel at the bottom of the UI displays information on the automatic changes to `EnableImport` as per the points 3, 4 and 5.
7. The UI displays all dependencies in both directions. The type and direction of the dependency for the selected parent will be indicated by icons.
8. Workflow plug-ins: These cannot be deselected separately, as they can only be transported as part of the workflow.
Analogous behavior for the Ticket Wizard and the Ticket Wizard controls is displayed.
9. The current navigation path will be displayed in the breadcrumb at all times. By clicking on an entity in the breadcrumb, navigating directly to this location is always possible.
However, it is not possible to select an entity that is already in the current navigation path from the list.
Example: The user navigates to the group *Support*. Here, he can see all contained users, among other things. After selecting the user *Hans Muster*, he can see all entities connected to Hans Muster, including the group *Support*. If he wants to select this group, a respective message will point out that the group can already be found in his navigation path. He can, however, still return to the group *Support* via the breadcrumb.
10. Three state checkbox for the display of `EnableImport`:
 - a) Checked: Entity and all existential or partially existential successors are selected for import.
 - b) Checked gray: Entity is selected for import. However, not all of its existential or partially existential successor are selected for import as well.

- c) Empty: Entity will is not selected for import, and thus no dependencies type I and II as well. However, successors with type III may be selected.

Auxiliary Definition “Child”:

An entity a is a child of another entity b, if a is dependent on b.

Example:

A ticket field can be found in a ticket schema. Thus, the field depends on the schema and is a child of the schema.

Definition “Successor”:

An entity a is a successor of another entity b, if a is a child of b or there is a path spanning multiple entities, which are all children of each other.

Example:

A service transaction *Incident* can be found in the service *Outlook Web Access*. This service in turn can be found in the catalog *IT Management*.

Incident is thus a successor of the service *Outlook Web Access* and also a successor of the catalog *IT Management*.

Definition "Parent":

A parent is the reversal of a successor.

Example:

Based on the example for successors: The catalog *IT Management* as well as the service *Outlook Web Access* are both parents of the service transaction *Incident*.

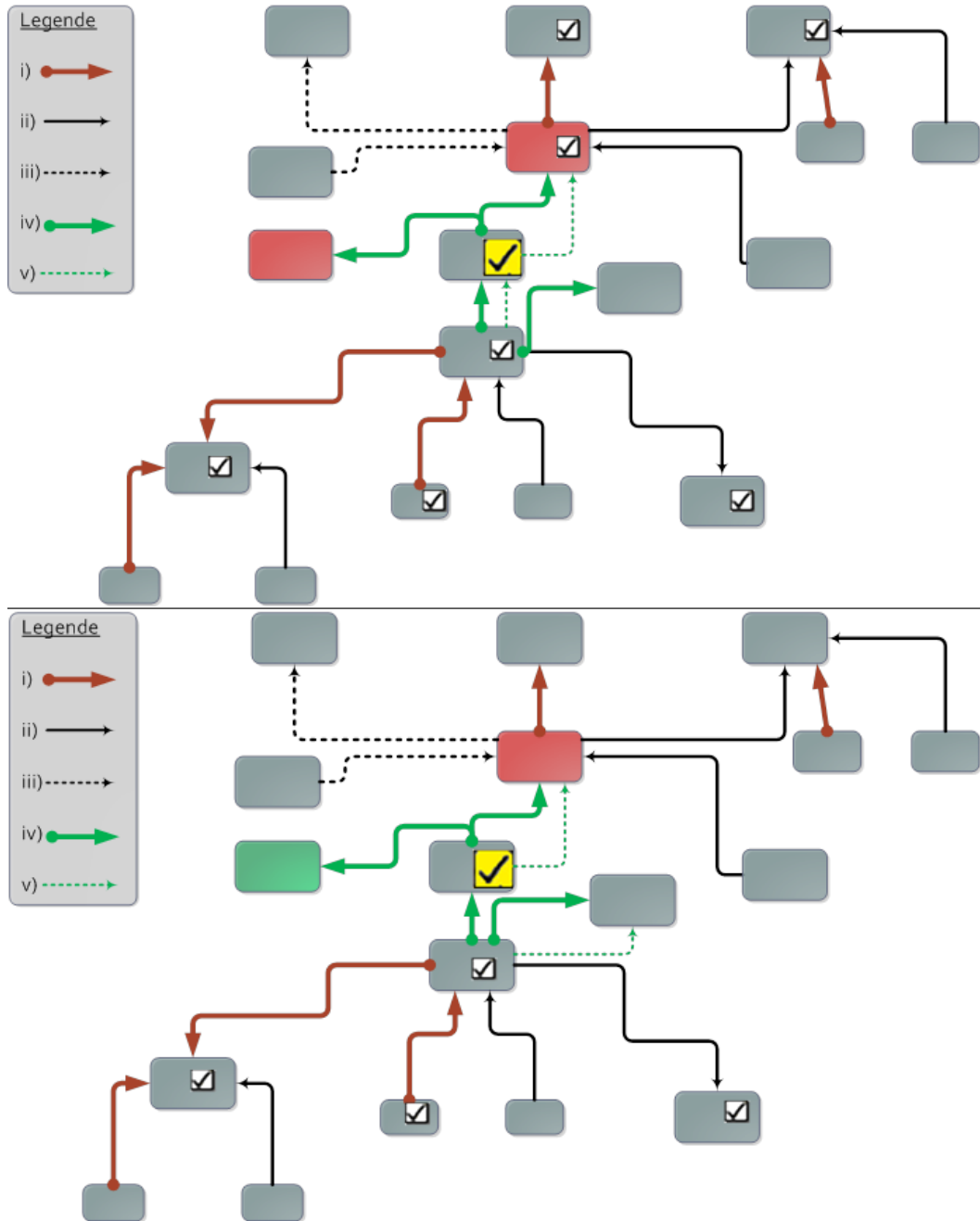


Figure 2.8: Import Activation: Whenever the import of an entity is activated (checkmark with yellow background), the import of the marked entities (checkmark with white background) will be activated as well automatically.

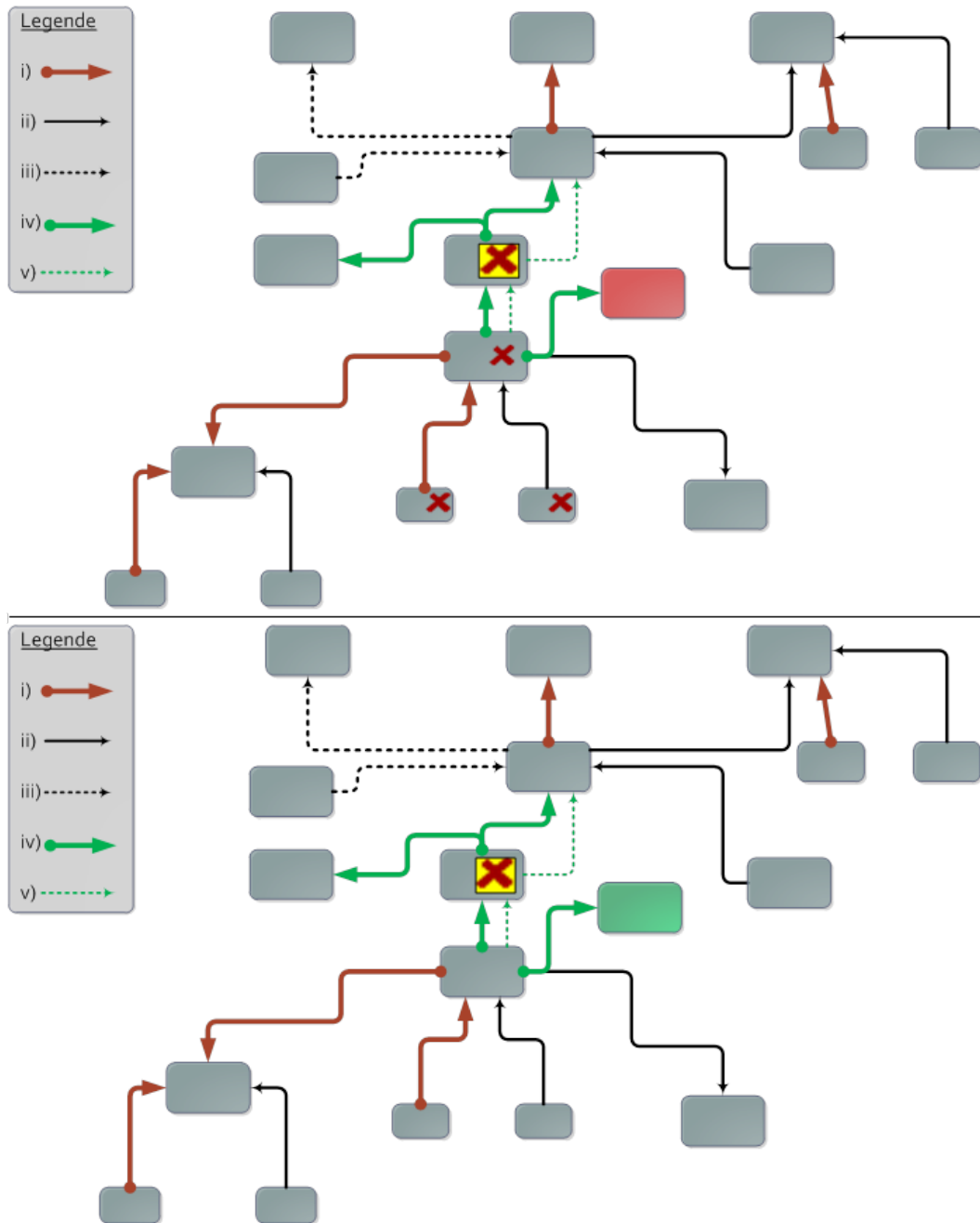


Figure 2.9: Import Deactivation: Whenever the import of an entity is deactivated (cross with yellow background), the import of the marked entities (cross with white background) will be deactivated as well automatically.

2.6 Errors During the Import

Various errors can occur, for instance if

- configuration changes are made in the background during the **export** (e.g. because the offline mode has not been activated),
- important services fail or are canceled during the **import**,

- database problems occur (e.g. due to insufficient hard drive space),
- or other circumstances preventing a proper operation of *Xpert.NET* occur at least temporarily (e.g. due to a power outage).

Please note

Manual manipulations of a transport package may also cause incompatibilities or unpredictable behavior of the target system!

If such an error occurs, the import has definitely not been executed correctly and thus, data consistency cannot be guaranteed. An error dialog appears and all information on the error is written into the log file additionally.

If a rollback is necessary, the currently logged in administrator will be redirected to the rollback dialog automatically.

Please note

If an error occurs at the time of import, the offline mode will remain in effect. If the data of a transport package cannot be written into the databases, a dialog displaying information on the error will appear. Simultaneously, further information on this can be found in the logfile.

The error dialog will now offer the option of a rollback. Thus, the origin data of the target system from before the import will be restored in order to maintain a functioning *Xpert.NET* system. This is done via a complete export of the system configuration before the export, which can then be re-imported in case of an error.

Warning

The transport package created for the rollback will be overwritten as soon as an attempt to import another transport package after a failure is made. However, the rollback dialog has to be closed manually for this purpose.

Rollback Failure

In case of an error that prevents both the correct import as well as the subsequent rollback (e.g. a database server breakdown), a further error dialog will be displayed. Exports and imports will not be possible from that moment on, until a rollback is successfully executed. Every time the XCM module is opened, the dialog will be displayed and the system will remain offline.

2.7 Conflict Resolution and Mapping Entities via Secondary ID

During the import, the entities will be mapped according to their distinct ID first. Usually, this is a GUID, which is distinct on all systems and clients. If multiple entities that have the same name within their scope are created on the source and the target system manually, these entities have to be mapped manually before the import as well. For this mapping, the secondary ID - usually the name (the `FriendlyName`, not the localized name) - is used.

Entities from the transport package that cannot be assigned to an entity on the target system distinctly are used for mapping. For these entities, all entities with the same name, the same type, and the same scope will be searched during validation. In this case, the scope is the superordinate entity.

If an entity with the respective name can only be found once, it will be mapped automatically. If there are multiple entities with the same name and scope, the user has to perform the mapping manually.

Examples for the Scope

A message template's scope is the template group it has been created in.

A priority has the expense of an SLM contract as the scope. The expense in turn has the SLM contract as a scope.

The scope of an entity makes sure that entities cannot be mapped into another (superordinate) entity by mistake (e.g. a priority into another SLM contract).

The following section describes possible cases in detail, using the entity *Ticket Status* as an example. The source system is the transport package here.

Case 1

Source System

Global ticket status "New" with ID 1

Target System

Global ticket status "New" with ID 2

In this case, the entity will be mapped automatically, as only one entity with the same name can be found on the target system.

Case 2

Source System

Global ticket status "New" with ID 1

Global ticket status "New" with ID 2

Target System

Global ticket status "New" with ID 1

The entity with the ID 1, which can be found on both systems, will be mapped automatically via the primary key. The entity with the ID 2 will be newly created on the target system, as no ticket status identity with the same name can be found there.

Case 3

Source System

Global ticket status "New" with ID 1

Global ticket status "New" with ID 2

Target System

Global ticket status "New" with ID 1

Global ticket status "New" with ID 3

The entity with the ID 1 will be mapped automatically via its primary key, as it can be found on both systems. The entity with the ID 2 cannot be identified distinctly on the target system. Only one ticket status entity with the same name can be found on the target system. Thus, this entity will be mapped automatically (see also Case 1).

Case 4

Source System

Global ticket status "New" with ID 1

Global ticket status "New" with ID 2

Global ticket status "New" with ID 3

Target System

Global ticket status "New" with ID 1

Global ticket status "New" with ID 4

This case is a hybrid of the cases 1 to 3.

The entity with the ID 1 can be found on both systems and thus will be mapped via its primary key. The entities with the IDs 2 and 3 cannot be clearly identified on the target system. However, a similar entity with the ID 4 can be found there, which has the same name as the other two. This is a conflict that has to be solved manually. The conflict resolution dialog allows for mapping one of the entities with the ID 2 or the ID 3 on the entity with the ID 4 (target system). The entity not mapped here will be newly created on the target system and can be found via its primary key during the next import.

Case 5

Source System

Global ticket status "New" with ID 1

Global ticket status "New" with ID 2

Target System

Global ticket status "New" with ID 3

Global ticket status "New" with ID 4

In this case, none of the entities can be clearly matched. Via the conflict resolution dialog, at least one of the entities on the source system has to be mapped on an entity on the target system manually. The second entity will then be mapped on the remaining entity with the same type automatically.

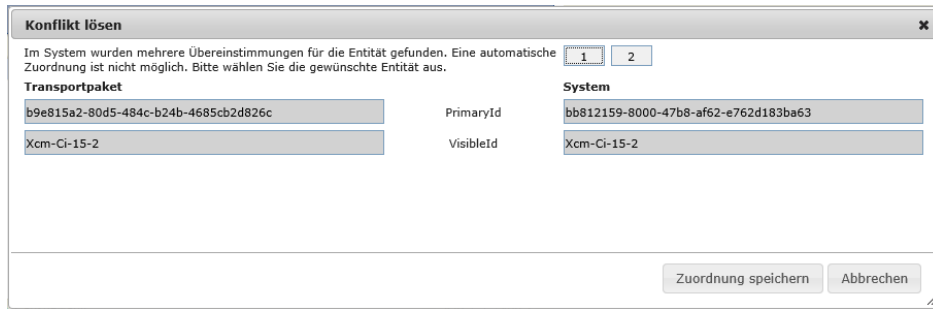


Figure 2.10: Conflicts require a manual intervention

Case 6

Source System

- Global ticket status "New" with ID 1
- Global ticket status "New" with ID 2
- Global ticket status "New" with ID 3

Target System

- Global ticket status "New" with ID 4
- Global ticket status "New" with ID 5
- Global ticket status "New" with ID 6

This case is an extension of Case 5 and can be solved analogously. Here, the first two entities have to be mapped manually via the conflict resolution dialog. The third entity will be mapped onto the remaining entity automatically.

Case 7

Source System

Global ticket status "New" with ID 1 is to be deleted.

Target System

Global ticket status "New" with ID 1 still in use in

Ticket schema „Change“ and „Incident“ and the corresponding workflows

The conflict resolution dialog shows the use in the two ticket schemas and the corresponding workflows. The options here are: not deleting the status in order to not jeopardize the consistency of the ticket schemas and workflows; deleting the status nevertheless, which makes additional manual work on the ticket schemas and workflows necessary; canceling the import.

2.8 Saved Conflict Resolutions

Already resolved conflicts (manually as well as automatically) will be saved by the XCM in order to minimize the necessary rework during the next import.

Warning

Deletion conflict solutions (case 7) will only be saved within the loaded package but not stored permanently for further imports.

The conflict resolutions will be saved as follows:

Recognition of Performed Secondary Mappings

A secondary mapping table with the following content will be created:

- Entity type,
- source system primary ID,
- target system primary ID,
- meta information, e.g. client, instance, ...

The import will then be processed as follows:

- Applying the primary ID mapping;
- applying the secondary ID mapping via secondary ID mapping table (exchange of primary IDs);
- automatic as well as manual ID mappings write the mappings into a temporary table (similar to the secondary mapping table);
- transferring the content of the temporary table into the main table in case of a successful import.

Improving the Conflict Resolution for Multiple Transferred Entities with the Same Secondary ID

- Before the secondary ID mapping (after the primary ID mapping), the entities are grouped within the scope:
 - First according to type,
 - then according to secondary ID (deleted entities will be treated in the same way as not deleted entities in this grouping).
- If there is more than one entity within this grouping on the source system **and** at least one match on the target system, a conflict for all entities in this group will be displayed.

2.9 Checklist - English not implemented yet

2.10 Checklist - English not implemented yet

3 Dependencies and Transferred Data

Generally, there are five different dependencies:

1. Existential Dependency

Definition: A is existentially dependent on B, if A cannot exist without B.

Example: A ticket field (A) cannot exist without the respective ticket schema (B).

2. Functional Dependency

Definition: A is functionally dependent on B, if A cannot function correctly without B.

Example: A workflow (A) using a certain ticket field (B) cannot not work correctly, if this field does not exist. The workflow can, however, exist without the ticket field theoretically.

In case of one of these two types of dependencies, dependent objects will be selected for import automatically, in order to avoid an inconsistent target system.

3. Link

Definition: A is linked to B, if there is a connection from A to B.

Example: A CI linked to another CI.

Dependencies of this type will not be imported automatically during partial imports.

4. Partially Existential Dependency

Definition: A is partially existentially dependent on a set of entities B, if A cannot exist without at least one entity from B.

Example: A user is partially existentially dependent on the set of his parent groups. That is, the user can only exist if at least one of his parent groups exists.

5. Default Entity of a Partially Existential Dependency

Definition: In the set of entities B of a partially existential dependency, one entity is emphasized. This entity is called the default entity of the partially existential dependency.

Example: The main group in the User Management. It is an emphasized group among the parent groups of a user.

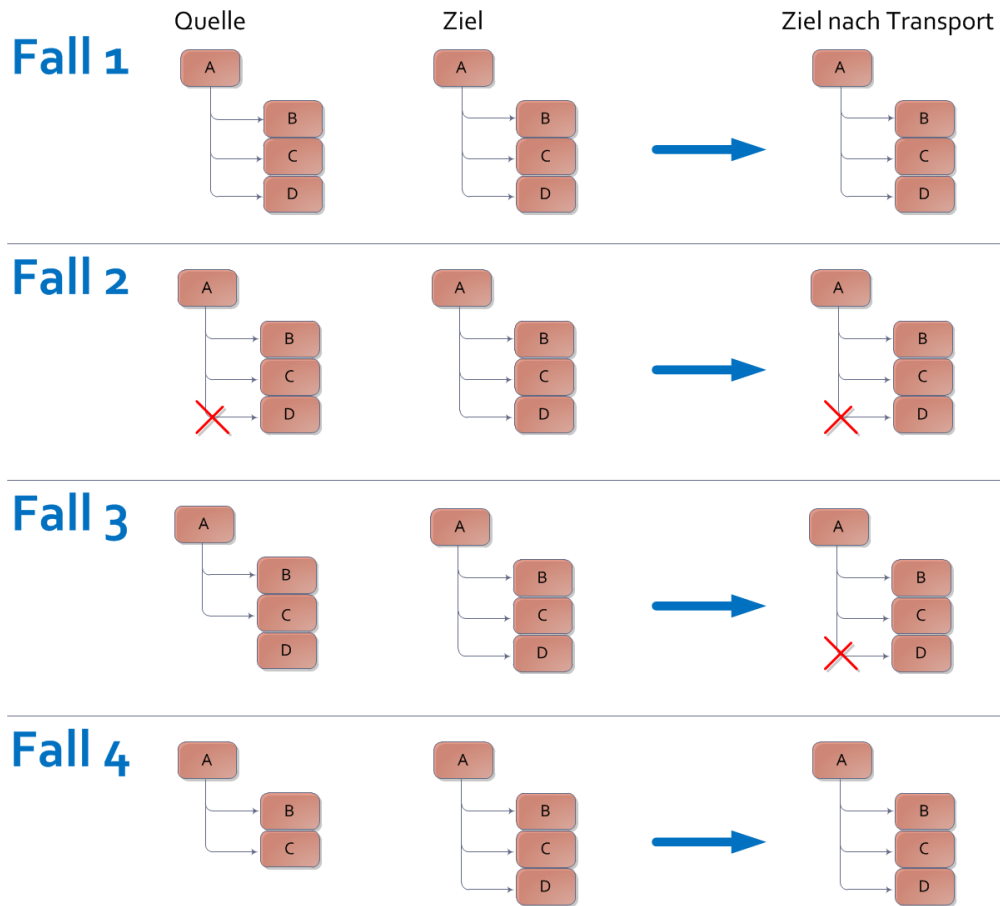
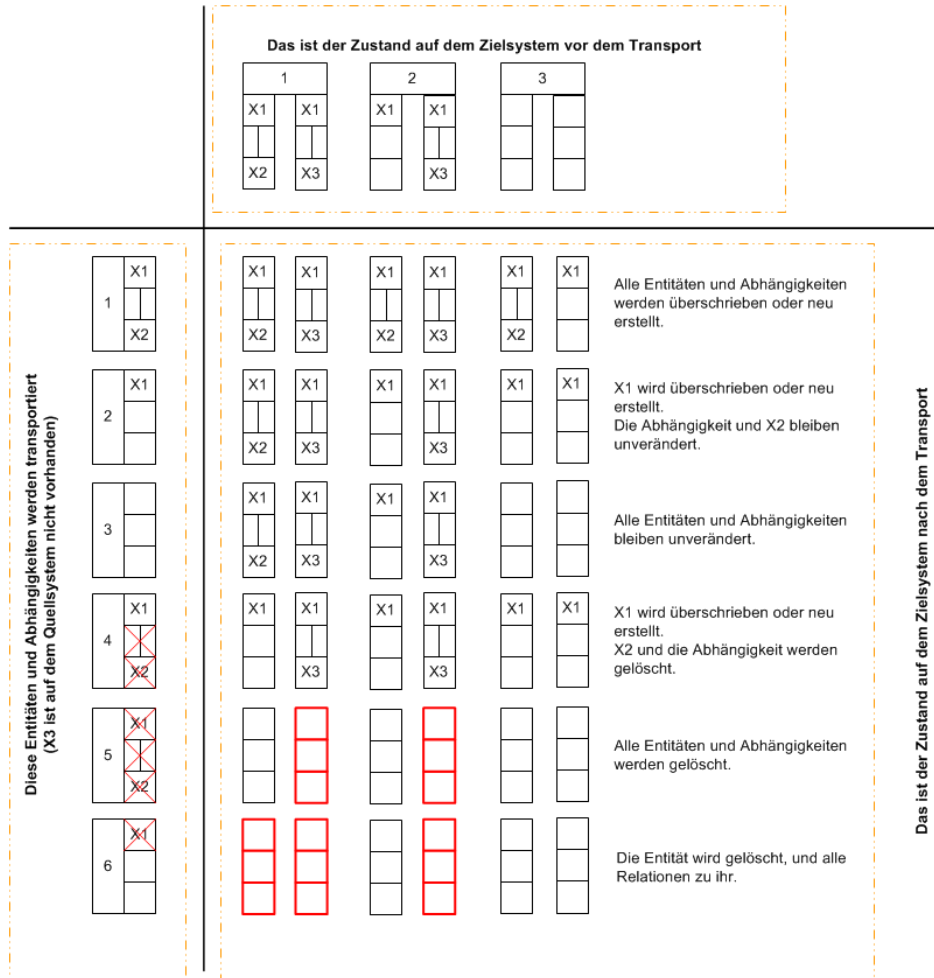


Figure 3.1: The transfer behavior of lists

Transportmatrix I

beschreibt Regeln für existentielle Abhängigkeiten zwischen Entitäten



Legende

- X1 Entität 1
- X2 Entität 2
- X3 Entität 3
- Abhängigkeit zwischen 2 Entitäten
- X1 Als gelöscht markierte Entität
- Als gelöscht markierte Abhängigkeit

Figure 3.2: The transfer matrix for the behavior of the dependency types 1, 2, and 3

Transportmatrix II / III

beschreibt Regeln für Funktionale Abhängigkeiten / Verknüpfungen zwischen Entitäten

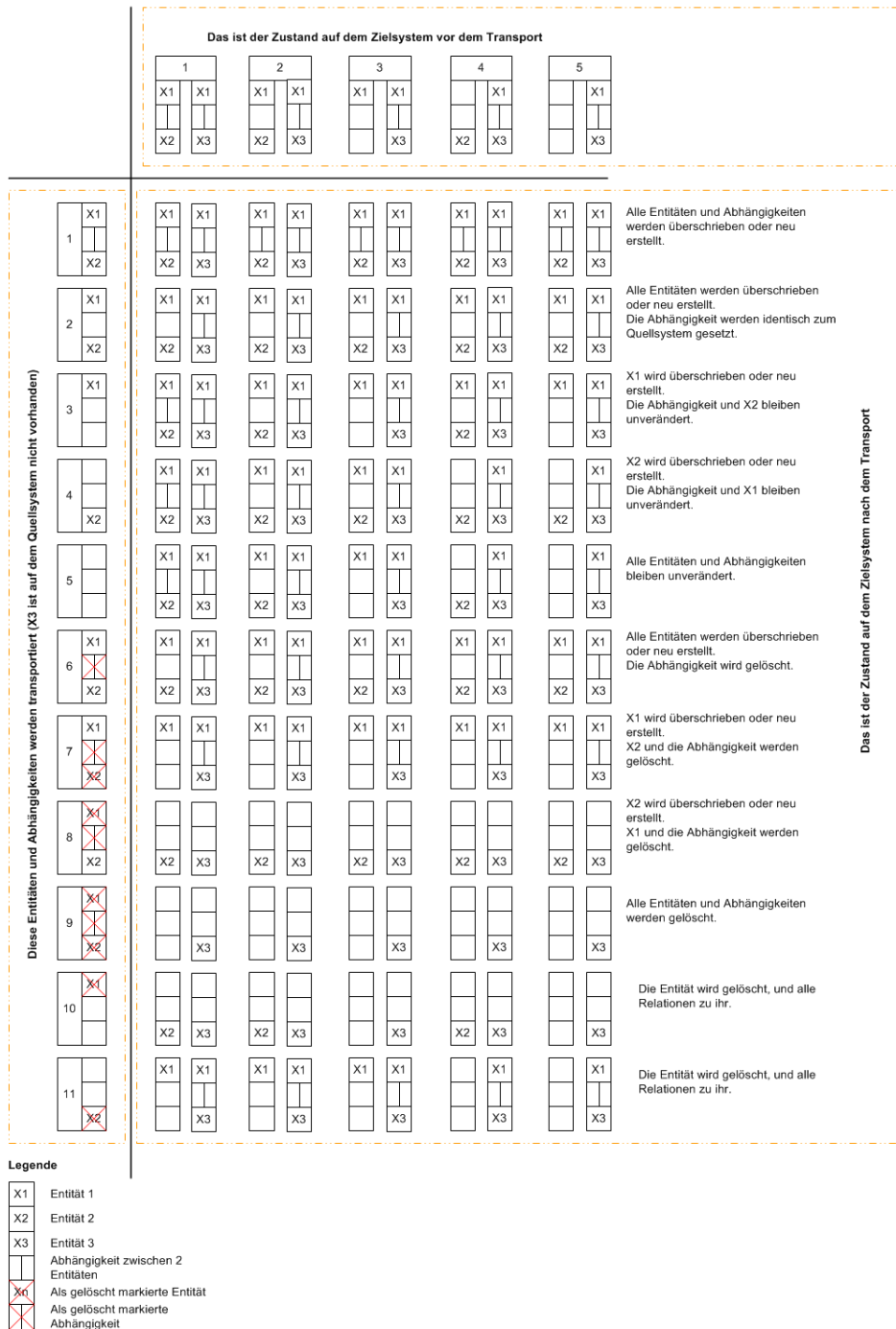


Figure 3.3: The transfer matrix for dependencies and links between entities

Dependencies are treated as follows here:

- If the linked entities are dependent on the origin entity (e.g. child elements on their parent elements), the origin entity does need not a list of dependencies on the linked entities. This is always the case for the dependency type 1 (never dependency of parent element on its children necessary). In this case, there are no actual lists.

- If the linked entities do not know about the origin entity (e.g. a ticket status does not “know”, in which workflow plug-ins or ticket actions it is used), the origin entity has to have dependencies on the linked entities (here: lists).
In this case, the XCM compares the list content of the target system to the transport package. Thus, the XCM has the following functionality for the list matching:
 - Only dependencies of the type 1 and 2 not known in the transport package can be removed on the target system. Unknown entities have to remain. This applies to the effective import as well as the validation.
 - In some cases, there are mutual dependencies, e.g. CI-CI (dependency type 3). The same logic as above applies here (“match” list but ignore “unknown” entities).

3.1 Overview of Data not to be Transferred

XCM allows for the export and import of configuration settings. The following data of an *Xpert.NET* system is not part of these settings or an exception.

Data not transported:

- DYNAMIC DATA (tickets, expenses, KB articles, mails, dispositions, tasks, task folders)
- Image files, e.g. icons for statuses. These have to be copied to the right position in the file system.
- IMPORTED SERVICES and TRANSACTIONS of the Service Portfolio
- ARCHIVE SETTINGS
- CMDB IMPORT QUEUE and IMPORT CONFLICTS
- Basic configuration
- Maintenance
- Worker Management
- USER-DEPENDENT DATA, such as accounts, filters, favorites, and widgets
- REPORTS and REPORT GROUPS
- CONFIGURATION FILES and their contents (*.config, worker configurations)
- FIELDS TRIGGERING ACTIONS and saving no values
- DISPLAY OPTIONS saving no values
- CROSS-CLIENT SETTINGS
- The SQL TABLES ClientConfiguration, GlobalConfiguration, and ConfigSettingsSystem (including XCM versioning pattern)

3.2 Automated Event Management

The following Automated Event Management data can be transferred via the XCM:

3.2.1 Connectors

All connectors including all configuration settings can be transferred via the XCM.

3.2.2 Filter Rules

All filter rules can be transferred via the XCM. In this process, the hierarchy of the filter rules, their order, and all conditions of every single rule will be taken into account.

3.2.3 Value Rules

All value rules including their hierarchy and conditions can be transferred via the XCM.

3.2.4 Event Ticket Mappings

All mappings can be transferred via the XCM.

3.3 Configuration Management (CMDB)

If the option *Include dynamic data* is activated during the export, CIs will be treated as configuration data and those, whose schemas have been selected, will be transferred as well. See also *Section 3.3.5*.

Additionally to the CIs, the following CMDB data can be transferred via the XCM:

3.3.1 CMDB Relation Types

All created relationship types including all fields will be taken into account.

3.3.2 CMDB Life Cycles

All configured life cycles can be transferred. Every life cycle contains status configurations as well as their transitions. Icons will not be exported, only a link to the file on the import server.

3.3.3 CMDB Schemas

All configured CMDB schemas can be transferred; they are divided into CI and zone schemas. Types are taken into account by the XCM as well.

CI Schemas

All information for CI schemas can be transferred. Icons will not be exported, only a link to the file on the import server.

Zone Schemas

All configured zone schemas including the default hierarchy will be transferred. Icons will not be exported, only a link to the file on the import server.

Types

All configured types will be transferred. Icons will not be exported, only a link to the file on the import server.

3.3.4 Import

All configured imports will be taken into account and transferred. Conflicts and conflict solutions, however, are dynamic data and thus will not be transferred. The same applies to the import queue.

3.3.5 Configuration Management Database

The entire zone structure as well as all CIs contained in it can be transferred, depending on the export settings.

File attachments will be exported binary along with a hash.

CI relations will be transferred as a type 3 dependency here.

3.4 Computer Telephony Integration

All CTI data, like extensions or master numbers, can be transferred via the XCM including their configuration.

3.5 Dashboard

The XCM allows for transferring all available categories and the master widgets created therein, including their entire configuration.

User-specific widget customizations, however, are dynamic data and will not be taken into account by the XCM.

3.6 Early Warning System

The XCM can transfer all available early warning items including all details and all configured rules.

3.7 Expense Management

The following Expense Management data can be transferred via the XCM:

3.7.1 Projects

All projects and all configured fields contained therein as well as cost centers and their fields can be transferred.

3.7.2 Expense Types

All configured expense types including all their fields can be transferred.

3.7.3 Allocation Types

All allocation types including all configuration settings can be transferred.

3.8 Knowledge Management

The XCM allows for transferring all of the KB categories as well as the knowledge base schemas. KB articles cannot be transferred, as they are dynamic data similar to tickets.

3.8.1 Categories

All categories including localized texts and their hierarchies can be transferred via the XCM.

3.8.2 Schemas

The XCM allows for transferring all existing schemas including localized texts, validity, configured actions, resubmission configuration, and status transitions.

3.9 Messaging and Collaboration

The following messaging data can be transferred via the XCM:

3.9.1 Dynamic Placeholders

XCM is able to transfer all of the dynamic placeholders. In the process, all of the (localized) default and placeholder texts as well as all specified conditions will be taken into account.

3.9.2 Information

All defined categories can be transferred. All of the information can be transferred as well, irrespective of whether it is marked as active/inactive or whether the current date lies within the start and end date. In the process, all the fields filled in the information will be transferred as well, just like the image data of the articles.

3.9.3 Mail2Ticket

All active and inactive Mail2Ticket configurations will be transferred.

In order to prevent unwanted retrieval of e-mails, the `ACTIVE` flag will be deselected as soon as a configuration has been transferred to the target system. If the configuration already exists, the flag will not be overwritten, unless the configuration is changed by the XCM.

- All account information settings will be transferred as well.
 - `CREATE USER`: This information constitutes a dependency on the User Management. Imported users that are in use will not be imported in the process.
 - `AFFECTED USER`: This information constitutes a dependency on the User Management. Imported users that are in use will not be imported in the process.
 - `TICKET SCHEMA`: This field constitutes a dependency on a ticket schema.
 - `SERVICE TRANSACTION`: This field constitutes a dependency on a service transaction.
 - `CONFIGURATION ITEM (CI)`: This field constitutes a dependency on a CI from the CMDB. The CI dependency will only be exported, if the CI is part of a schema that has been activated for the export via the selection of the dynamic data.
 - `REFERENCE TO`: This field constitutes a dependency on a ticket field within a ticket schema.
 - `TEXT TO`: This field constitutes a dependency on a ticket field within a ticket schema.
- The settings made on the *Configuration* tab will be transferred completely. Entered passwords are transferred in encrypted form.
- All keyword mappings will be adopted. There is a dependency on each referenced ticket field.
- The configuration of the SLM priorities of a Mail2Ticket account can be transported as well. There is a dependency on the Service Level Management.

- The mapping of the reply e-mails can be transferred to a workflow using the ticket status. There is a dependency on each referenced ticket status and workflow, which can be resolved correctly by the XCM.

3.9.4 Messaging Schemas

The Configuration Management transfers all configured message schemas including their groupings and the list of configured groups. The assignment to process schemas will be maintained in the process.

3.9.5 Templates

The XCM transfers all templates. In doing so, all (localized) default and placeholder texts as well as all defined conditions will be taken into account.

3.10 Reporting Management

Neither reports nor their configuration will be transferred.

3.11 Service Level Management

The following SLM data can be transferred via the XCM:

3.11.1 Early Warning System

The XCM allows for transferring all configured early warning rules. All default fields and defined rules including the specified conditions will be taken into account in the process.

3.11.2 Service Level Management

The entire settings of contracts, expenses, priorities, and coverages will be taken into account during an export. Static file attachments will be exported binary along with a hash.

3.12 Service Portfolio Management

The XCM takes service life cycles and schemas into account. Configured service catalogs, services, and service transactions will be transferred as well.

3.12.1 Service Live Cycles

Service life cycles will be transferred along with the contained data and options (details and status). Icons will not be exported, only a link to the icon file on the server.

3.12.2 Service Schemata

Service schemas will be transferred with the specified data and options (details and dynamic fields).

3.12.3 Service Catalogs

Service catalogs will be transferred including all defined settings (details and attachments).

3.12.4 Service Categories

Service categories will be transferred including name and description.

3.12.5 Services

Services will be transferred along with their specified data and options (details, field values, CI links, SLP assignments, and attachments).

3.12.6 Service Transactions

Service transactions will be transferred including all data and options (details, role assignments, SLP assignments, and file attachments).

3.13 Task Management

The following Task Management data can be transferred via the XCM:

3.13.1 Task Status

All task statuses including the corresponding fields will be transferred - regardless of the visibility. Icons will not be exported, only a link to the icon file on the server.

3.13.2 Task Schema

All task schemas will be transferred including their corresponding configuration settings.

3.13.3 Task Life Cycles

All task life cycles including the corresponding fields (name, description, and initialization workflow) will be transferred.

Additional configurations for the individual actions will be transferred as well.

3.14 Ticket Management

The following Ticket Management data can be transferred via the XCM:

3.14.1 Ticket Number Ranges

All configured ticket number ranges will be transferred via the XCM.

3.14.2 Ticket Templates

All configured ticket templates as well as their grouping will be transferred. All file attachments will be exported binary along with a hash.

3.14.3 Ticket Filter Management

All globally defined filters as well as selected statuses will be taken into account. Defined search texts will be also adopted.

3.14.4 Ticket Status

All configured ticket statuses will be transferred. Icons will not be exported, only a link to the icon file on the server.

3.14.5 Ticket Conversions

All ticket conversions will be transferred, regardless of whether they have been marked as active or inactive. All fields defined for the availability will be adopted. Statuses and workflows will be transferred as dependencies as well. The mapping of the origin fields on the target fields will be transferred completely.

3.14.6 Classifications

All configured classifications for tickets will be exported.

3.14.7 Ticket Wizard

The Ticket Wizard will be transferred completely, including all pages, sections, fields, field configurations, and general options. All configured values will be taken into account.

3.14.8 Ticket Schema

All configured schemas will be transferred including all existing fields.

3.14.9 Ticket Actions

All configurable fields of an action will be exported. In the process, all configuration fields will be transferred, even if a configuration value constitutes a dependency.

3.15 User Management

All users and groups as well as existing relations within a client will be transferred.

Please note

User passwords are stored in encrypted form in the transport package.

Warning

Users created on the source system whose passwords have been changed subsequently, can only log onto the target system, if they have logged onto the source system with their new password at least once in order to initialize it.

A group or a user can be a member of several groups. In such a case, the group or user will be displayed in the tree view as a member of each of these groups. However, it still remains always the same group or user - these are merely multiple group memberships of the same entity.

Nearly all permissions granted or denied in all areas/categories for all user groups in the export package will be exported. Inherited permissions will not be taken into account, as they result from group memberships.

The following permissions are exempt:

- permissions on dashboard statuses
- permissions on dashboard status changes
- permissions on report groups and individual reports
- permissions on task folders and task packages

Please note

Users and groups are entities relevant for peripheral systems!

3.15.1 Team Management

All defined skills and their links to users and groups will be transferred.

3.16 Workflow Management

The following Workflow Management data can be transferred via the XCM:

3.16.1 Workflows

All workflows including their corresponding plug-in configurations and all dependencies will be taken into account and transferred.

3.16.2 Forms

Forms contained in workflow plug-ins will be taken into account and transferred including their specified attributes and configurations as well.

3.16.3 Dispatching and Rule Management

All dispatching rules as well as all corresponding parameters and conditions will be exported.

3.17 DataViews

All DataViews (e.g. the ticket and activity list) and their configurations can be transferred via the XCM.

3.18 Search Templates

All search templates created in the back end and their configurations can be transferred via the XCM. Custom search templates, however, will not be taken into account.

4 Glossary

Entity: Instance as well as type of an *Xpert.NET* item (its purpose can be deduced from the context). The entity is a subset of a module. Entities in *Xpert.NET* are, for example, CI schema, ticket schema, and knowledge base category.

Export: The option to extract and save a system's configuration. See also the keyword "Transfer".

FriendlyName: An entity's name. The FriendlyName does not depend on the language, in contrast to the localized names. The FriendlyName is used by the XCM as a secondary identifier for detecting existing entities.

Import: The option to import a saved configuration onto a new system. The XCM import can be executed either completely or partially (only parts of a configuration). See also the keyword "Transfer".

Client: In *Xpert.NET*, clients are isolated, independent instances within the same *Xpert.NET* main instance. For example, different companies as *Xpert.NET* users, whose data has to be isolated from each other, can be configured as clients. The data storage of every client is separated from the other clients. Individual users can be members of multiple clients – in such cases, the user can switch between the different clients in the user interface.

Rollback: A rollback is the "reset" of the system to the state before the start of the import. This option is only available after a failed import. In contrast to a restore, no system changes can be made between an import and the rollback (provided that the offline mode has been enabled as recommended). Thus, no data will be lost during a rollback.

Transfer: The transfer is the export of *Xpert.NET* configuration data from the source system as well as the import of this configuration data on the target system.

Transport package: A transport package is an XML file resulting from the export. All entities to be transferred as well as their configurations are saved in it. This transport package will be, in turn, used for the import.

XCM: Abbreviation for the *Xpert.NET* Configuration Manager.


5 Statistics and Change Management

5.1 Statistics

Created by	Maik Wisatzke
Creation date	29.09.2011
Doc-ID	DOC-290911-010
Version	2015-1
Status	Approved
Replaces version	3.9-2
Release date	12.06.2015
Valid from	Immediately
Valid until	Cancellation
Document name	Xpert.NET Configuration Management 2015-1_EN

5.2 Change Control

Version	Date	Executed by	Comments
3.9-1	04.02.2013	Maik Wisatzke	Created
3.9-2	06.05.2014	Maik Wisatzke	Created
2015-1	12.06.2015	Alexander Schmidt	Updates for 2015



Copyright© 2015
isonet ag, Engineering und Beratung,
Zürich

This document is intellectual property of isonet AG and protected by copyright. It must not be saved, copied, duplicated, or passed on photo mechanically, digitally, or through any other means. Nor must it be used for the execution of projects. The companies directly addressed in this project have the right to use this document for their purposes exclusively within the scope of this offer.