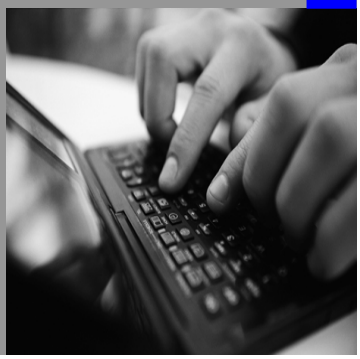


How-to Guide
SAP NetWeaver 7.0 (2004s)

How To... Consolidate Data in Qualified Tables

Version 1.00 – June 2007

Applicable Releases:
SAP NetWeaver 7.0 (2004s)
SAP NetWeaver Master Data Management 5.5 SP04 & SP05
Data Unification

1 Scenario

Repositories often contain data in qualified tables that must be consolidated as well. Typical examples are persons, business partners, or employees that contain address data in qualified tables.

MDM offers a powerful engine for cleaning and consolidating data in the main table but data in qualified tables can only be replaced or deleted. Nevertheless by using MDM standard functionality it is possible to implement a scenario where both Main table and Qualified tables can be independently consolidated.

2 General principle

In order to give a more understandable description of the implementation and use of the scenario described above, we will describe the particular case of having a repository containing Persons with Address information. The same ideas can nevertheless be applied to any other kind of business objects where a similar problematic appears.

2.1 Data Model

In our specific example we will take a simple repository that contains Persons data including a Qualified table where Address data is maintained (see Figure 1).

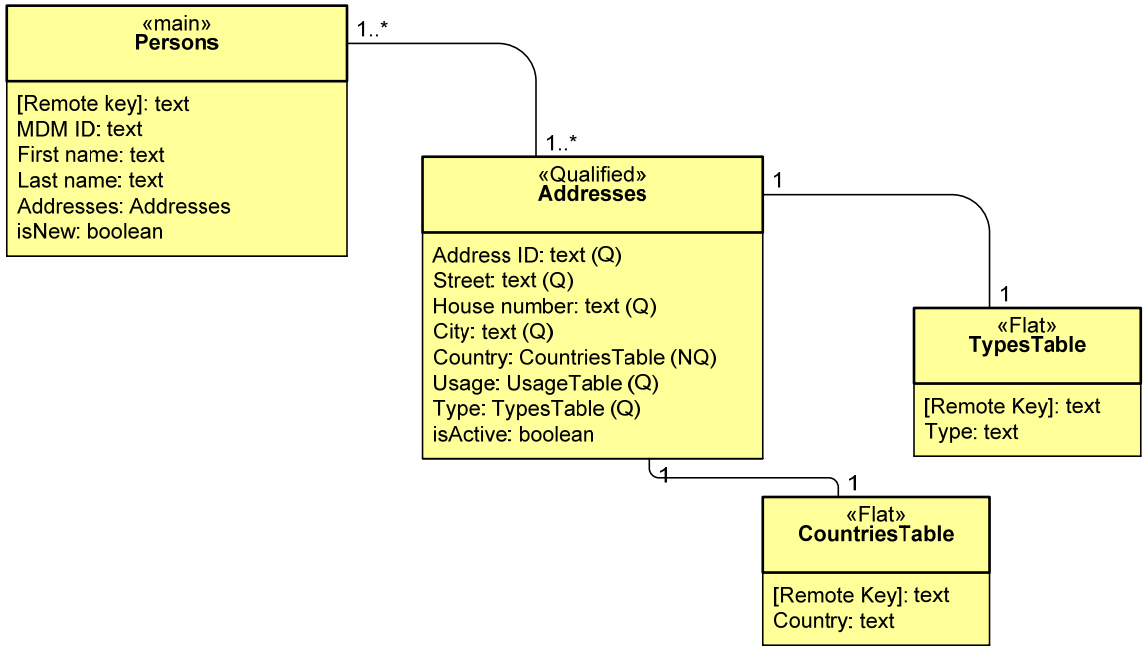


Figure 1 - Simple Persons repository used to illustrate this guide

The MDM ID field is not a Standard MDM Auto-ID field. It is a simple text field that will be updated at the end of the process with the actual MDM ID that was generated by the "Number range" functionality during Syndication.
In order to be able to consolidate the content of the Addresses Qualified table, we need a second repository for the Addresses (Figure 2). This repository is used for

Consolidation and Cleansing processes. As it will be seen later this Repository does not need to be installed in the same server where the Persons repository is.

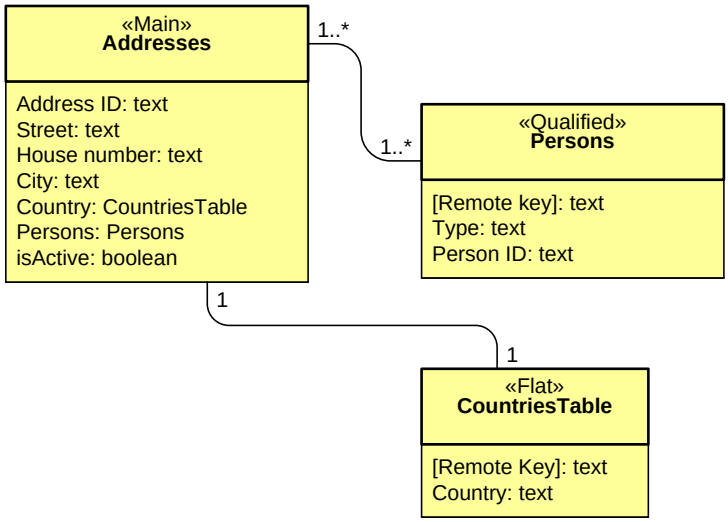


Figure 2
Figure 3 – Addresses repository used for Address consolidation

In the Addresses repository a Qualified table called “Persons” is used to keep track of the “role” that a particular address may have for a given Person.

2.2 General considerations concerning data model

In the data model described above, almost all fields used in the Persons repository are implemented as well in the Addresses repository. In principle we could have done as for the Addresses repository, and implement in the qualified table only the data necessary to keep the links to the Addresses. However the Persons repository contains full address data in order to:

- Allow central creation from the Data Manager
- Allow search and matching in the Persons repository using both Person and Address data.

2.3 Processes

In this guide we describe two basic processes: a Persons creation / update / consolidation scenario, and an Addresses update / consolidation/cleansing scenario. As you will see in the Step By Step Solution section the processes can be run independently, nevertheless both Syndication server and Import Server are used in order to keep both repositories synchronized.

2.3.1 Person creation, update or consolidation

An authorized user can create, update or consolidate “Persons” information in the Persons repository. An automatic syndication will than send the data to the addresses repository

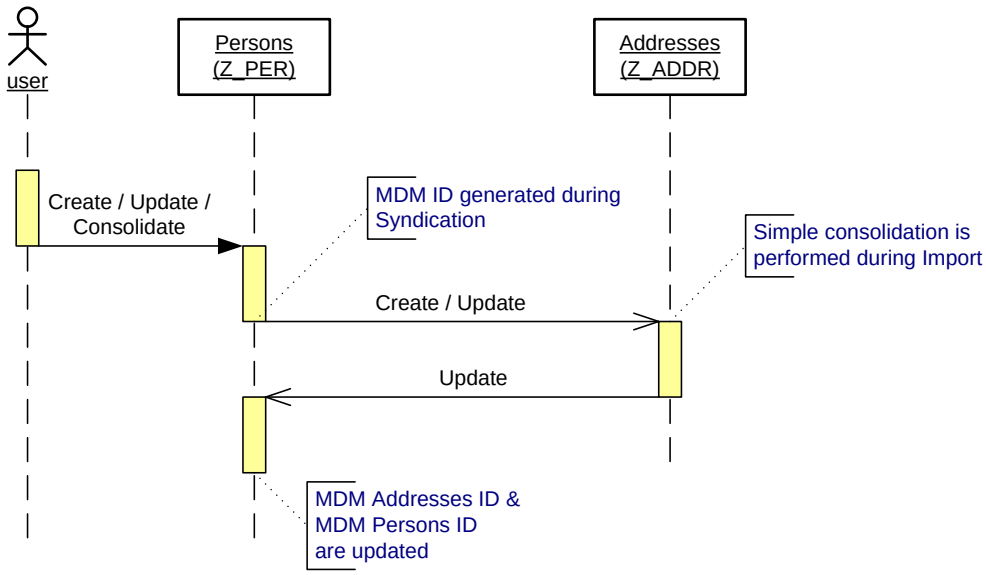


Figure 4 – Creation / Update / Consolidation of Persons data

During the automatic import to the addresses repository, a simple matching on address data is performed in order to reduce the number of duplicate entries. During the implementation of the outbound of the Persons repository the following points must be considered:

- The “isNew” Boolean fields, together with the “Suppress Unchanged Records” are used in order to control the data that is sent to the Addresses repository. The isNew control field is necessary in order to avoid a data flow loop between both repositories.
- The MDM ID is generated during Syndication by the Key Generation feature. In that way a key-mapping for Persons data is automatically generated. That is important for the cleansing and consolidation of Persons data. In that way the link between Persons and Addresses are automatically maintained. The generated MDM ID will be send back by the Addresses repository.
- During syndication the Persons repository must send basic Persons data (IDs), but, as expected, mainly the Address data is mandatory.

2.3.2 Addresses update and consolidation

In this process is possible to maintain or consolidate the data in the addresses repository independently of the processes that are running in the Persons repository. After each update the data is automatically sent to the Persons repository. This process has the following features:

- An address correction automatically propagates to the Persons repository.
- A merge between two addresses will propagate as well only and only if the content of the Persons qualified table is appended during the merge process.
- In order to avoid duplicate address entries in the Persons repository, the Address data key is given by an Auto-Id field.
- Rather than deleting, a status field is used in order to distribute an obsolete Status for an address. Later on a specialized process may decide what to do with the addresses (important for a possible GIS integration).

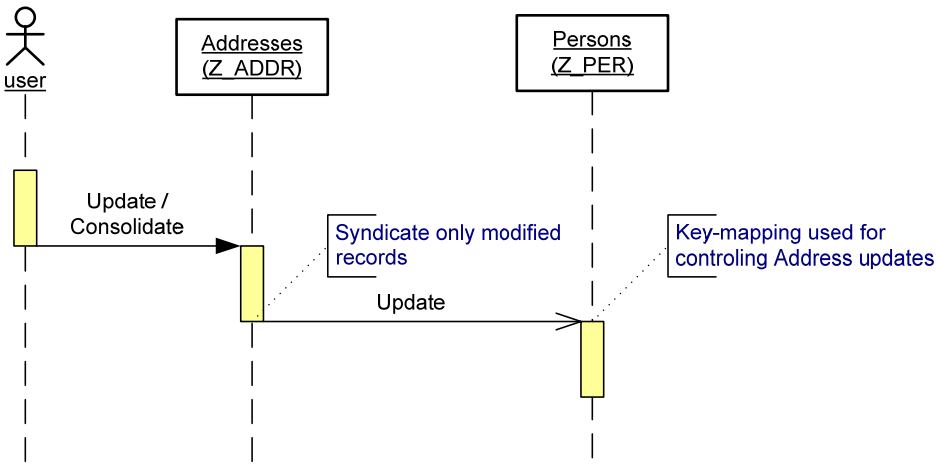


Figure 5 – Update and Consolidation of Address data

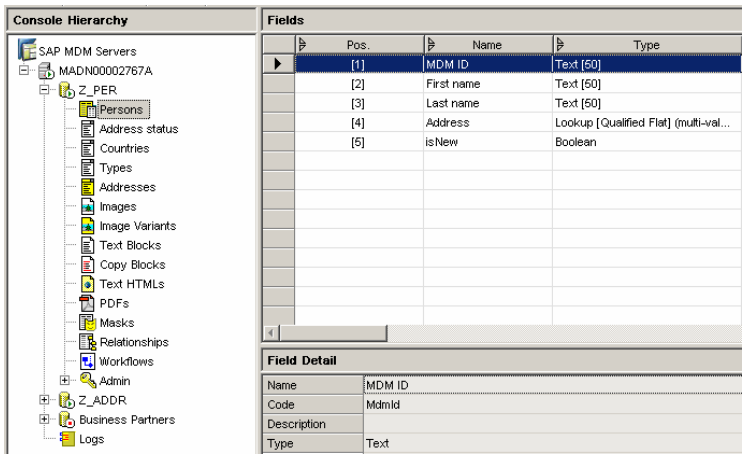
In our example we chose that the creation of new addresses will not generate the creation of a new “Person” in the Persons repository. Persons can only be created in the Persons repository. Nevertheless during initial load, the customer could upload an “address directory” to the Addresses repository that can be used for “matching” the incoming Persons data.

By merging records users should append the content of the qualified tables. In that way the merged content will be automatically sent to the counterpart repositories.

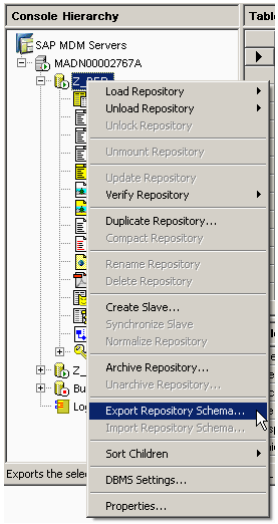
3 Step By Step Solution

3.1 Implementation of Persons repository

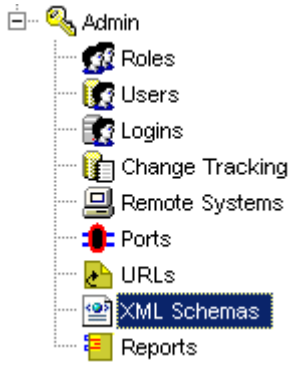
1. Implement Persons Repository (in our example *Z_PER*)



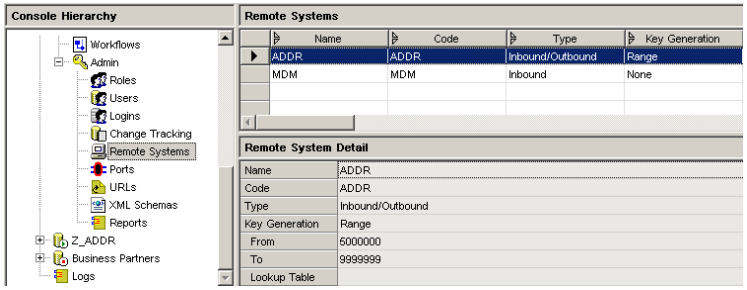
2. Generate XSD schema from the Existing repository. Please see corresponding How-to Guide: *"How to generate XSD Schemas from existing MDM 5.5 SP4 & SP5 repositories"*



3. Import the XSD to the XML Schemas into MDM using the Console.



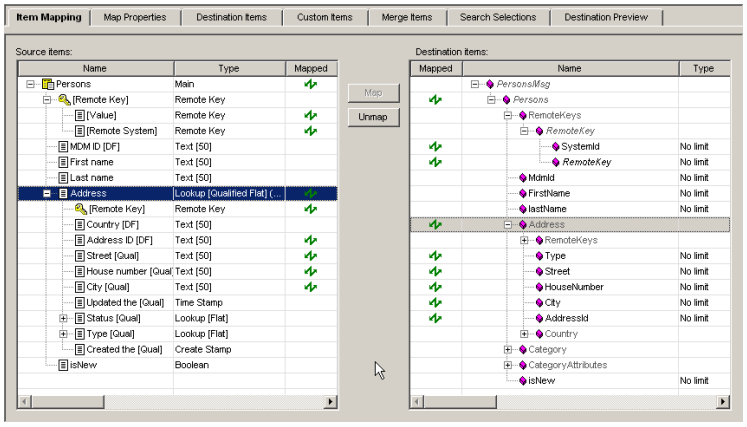
4. Define remote system ADDR, that corresponds to the repository where Address data will be consolidated. In our case we defined a Number Range that starts at 5000000. ADDR should be configured for both Inbound and Outbound



5. Define and store a Syndicator Map that will be used for sending the Address data to the Addresses repository (in our case *PER_OUT_G0*).

6. A good practice is to Syndicate test data that will be later used for the Address repository Inbound.

7. Please notice that only the remote keys, and the Address data are exported. There is no need to export data that will not be used by the target repository.

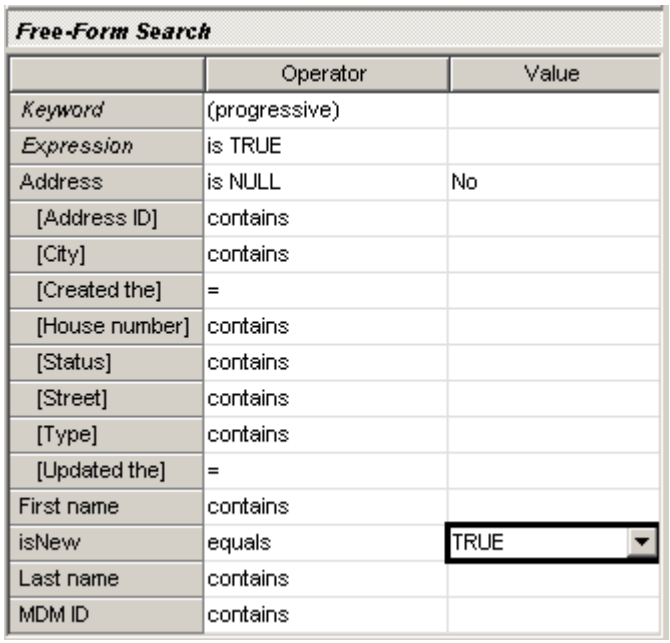


8. **IMPORTANT:** Use the parameters "`isNew equals TRUE`" and "`Address isNULL No`" in order to filter the records that will be Syndicated.

9. **IMPORTANT:** In "Map Properties" select the "Suppress Unchanged Records" option.

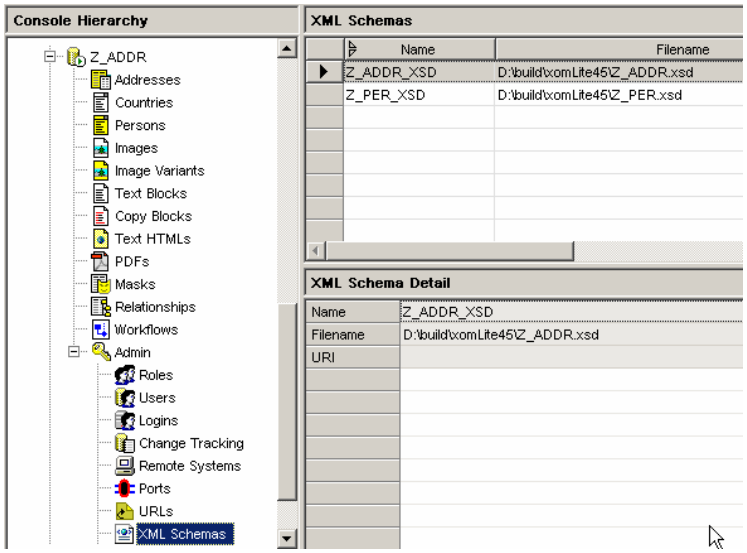
10. **IMPORTANT:** the default value of the field isNew should be set to TRUE in the console.

11. **Points 8 till 10 prevent your MDM system from creating a "Data flow loop" between your Persons and Addresses repositories.**

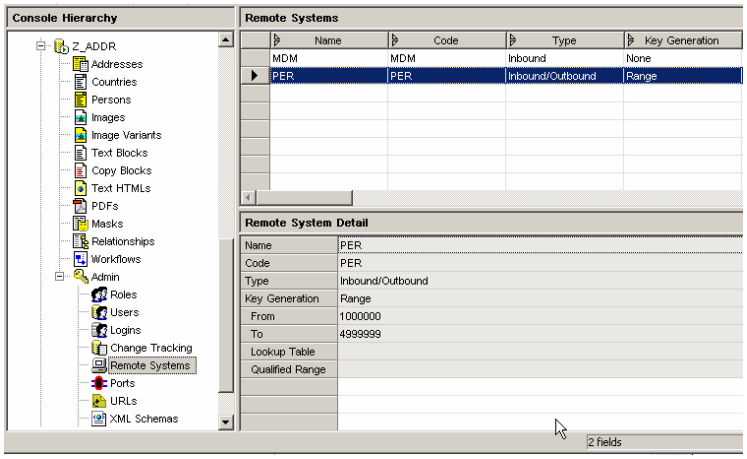


3.2 Implementation of Addresses repository

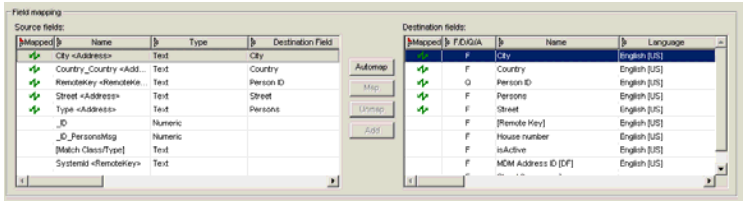
1. Implement Addresses repository (in our example Z_ADDR)
2. As for 3.1 export the XML-Schema of the repository, generate and load in MDM the corresponding XSD schema (in our example Z_ADDR_XSD).
3. Load as well the XSD used for Syndicating the data from Z_PER.
4. At the end of this step you should have in XML Schemas both Z_ADDR_XSD and Z_PER_XSD schemas.
5. Since addresses cannot de deleted from the Persons repository a status field will register either or not an address has become obsolete. Later cleansing processes can than use the content of this field.



6. As for the Z_PER repository define a Remote System enabled for both Inbound and Outbound. The use of a Number Range is facultative.

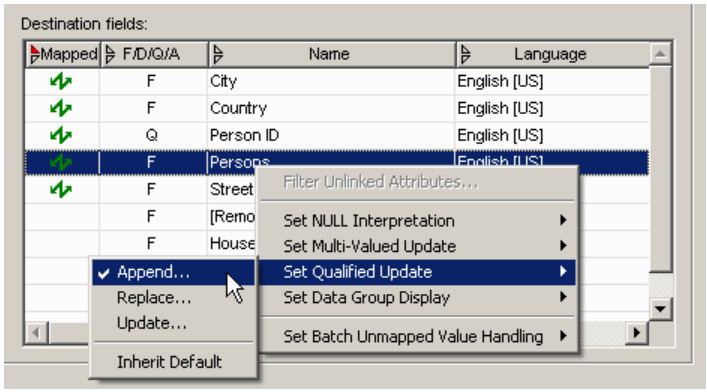


7. In the import manager and using the test data exported from the Z_PER repository define the necessary map for importing the data in the Z_ADDR repository.



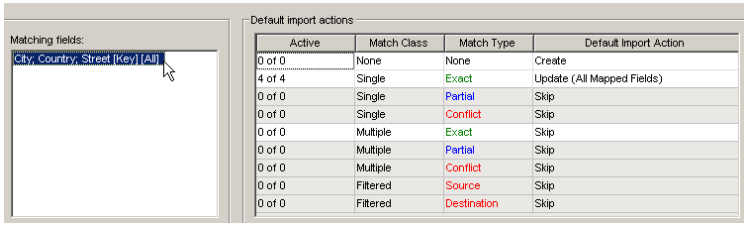
8. The Qualified Update was set to Append

9. The parameter PersonID is used as Matching Qualifier.



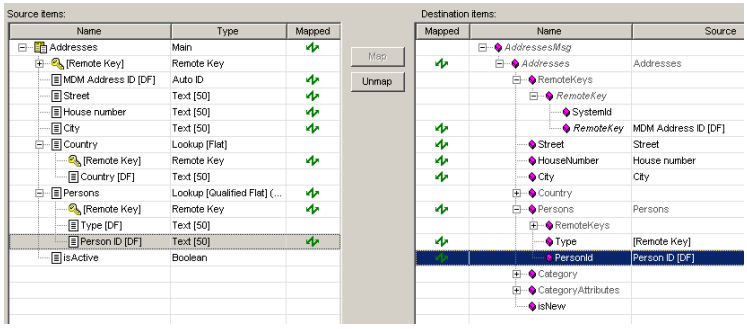
10. We used a combination between Street, House Number and City as matching attributes.

11. This way Identical addresses will be automatically consolidated.

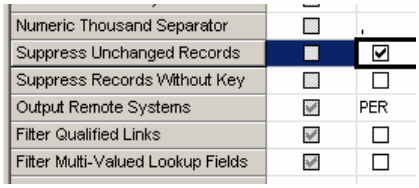


12. Create and save a Syndicator map that will be used for sending the updated/consolidated data back to the Z_PER repository.

13. As we did for Z_PER it is good practice to export sample data that will be used for defining the Z_PER inbound mapping.

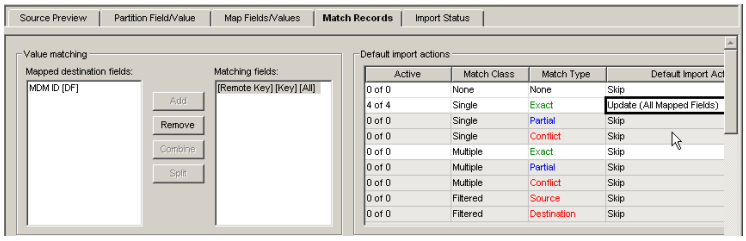


14. Contrarily to Z_PER, only the “Suppress Unchanged Records” option is selected. In this way all Address changes and updates are transported.

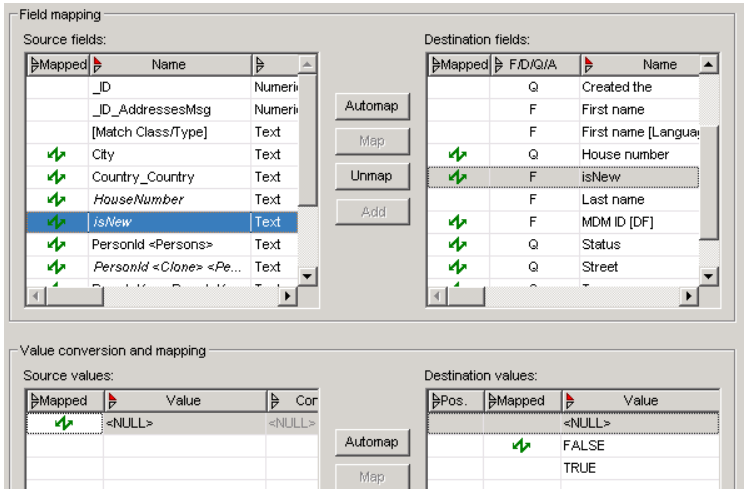


3.3 Sending data back to the Persons repository

1. To complete the MDM configuration, the data Syndicated from Z_ADDR must now be imported in the Z_PER repository.
2. The first step is to import the Z_ADDR_XSD schema into the MDM Console.
3. After doing that Load the repository and define a mapping in the import manager that will Import only the Address updates into the Persons repository.
4. The matching must be done against the “Remote Keys”. In that way eventual Merges will be correctly maintained during an address update.

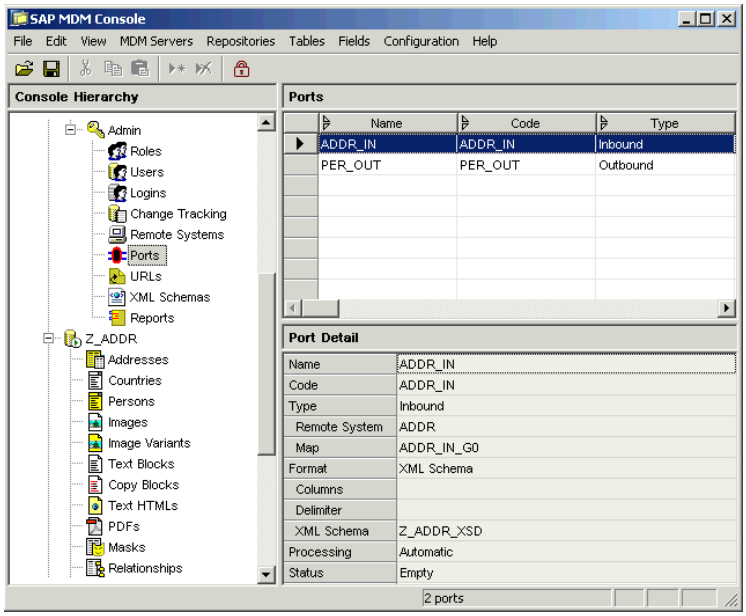


5. **IMPORTANT**: When the Address data is imported back to the Main repository, we must set the `isNew` flag to FALSE.
6. We do that by adding en empty field in the Import map. And by mapping the <NULL> value to FALSE.
7. By doing that the `isNew` flag is automatically set to FALSE during import from the Addresses repository.



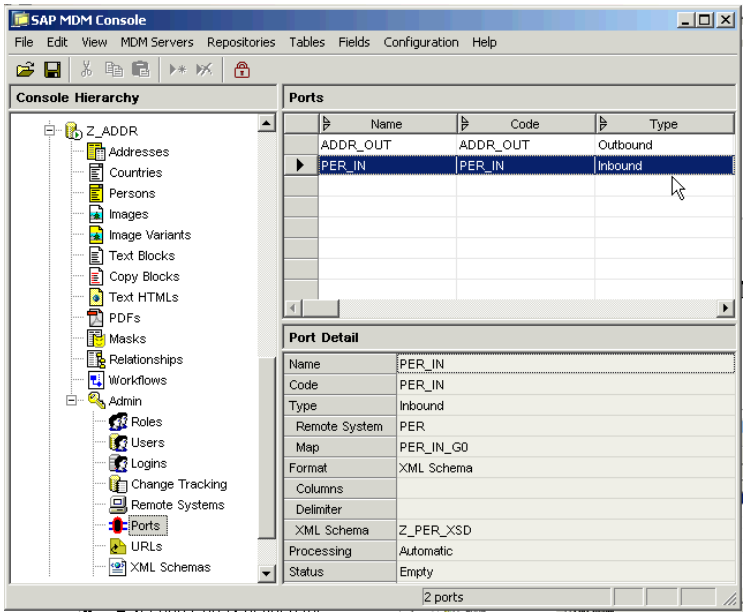
3.4 Configuring the Ports in Z_PER

1. After building all Interfaces and Maps the ports in Z_PER can be configured.
2. We define a Port PER_OUT that sends the new Address data to the Addresses repository (Z_ADDR). The corresponding mapping features is described in section 3.1
3. A second Port is defined for Importing the Updates and consolidated addresses from Z_ADDR. The corresponding interfaces and mapping features are described in 3.3.



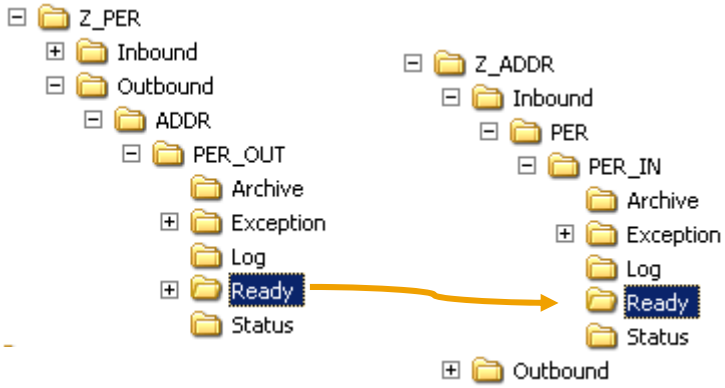
3.5 Configuring the Ports in Z_ADDR

1. After building all Interfaces and Maps the ports in Z_ADDR can be configured.
2. We define a Port PER_IN that imports the new Address data to the Addresses repository (Z_ADDR). The corresponding mapping features is described in section 3.2
3. A second Port (ADDR_OUT) is defined for exporting the Updates and consolidated addresses from Z_ADDR. The corresponding interfaces and mapping features are described in 3.2.

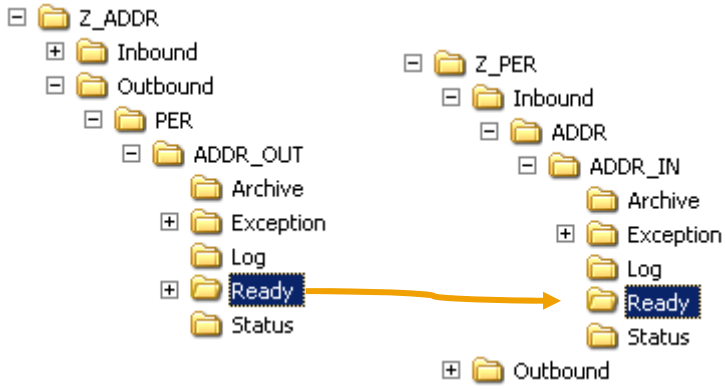


3.6 Link between repositories

After all Ports are configured we need to configure an automatic mechanism that transports the XML documents from the PER_OUT to the PER_IN Ports (see Figure below)



The same must be done between the Ports ADDR_OUT (Z_ADDR) and ADDR_IN (Z_PER)



That can be done either by using an Application that will automatically move the data between the respective “Ready” repositories (see Appendix), or to use the PI File Adapter: in that way the Z_PER and Z_ADDR repositories can even be placed in different machines.

4 Appendix

Attached to this file you will find a zip archive (SFS.zip) containing:

- `DISCLAIMER.txt`
- `launch.bat`*: Windows script for launching SFS.jar*
- `SFS.jar`*: a Java jar archive.*
- `Z_MDM.txt`*: Text file that contains information about the folders that should be synchronized at start-up.*

`SFS.jar` is a simple java application thought to allow the reader to build and experiment in a local machine (or stand-alone laptop) the scenario described in this paper. You can visualize the available commands by typing “help” after the SFS prompt. Please read DISCLAIMER.txt before starting.

To start the program type one of the two following commands:

- `java -jar SFS.jar`
- `java -jar SFS.jar load "Z_MDM.txt"`

More details on Z_MDM.txt are commented within the file `"Z_MDM.txt"`.