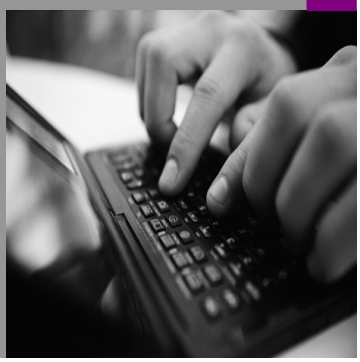


How-to Guide
SAP NetWeaver '04

How To... Troubleshoot MDM Import Manager

Version 1.00 – November 2005

Applicable Releases:
SAP NetWeaver '04
MDM 5.5 SP2

1 Scenario

MDM 5.5 is a very flexible framework that enable customers to load, consolidate and harmonize any kind of object thank to its very flexible data model. In all scenarios, it's necessary to run mass load of data from different data sources. This job is done through the Import manager tool which provides a very efficient user interface to map, match and load any kind of data source.
This document is intended to explain most of common issues encountered during the Import Manager use and understand how to solve typical problems.

2 Introduction

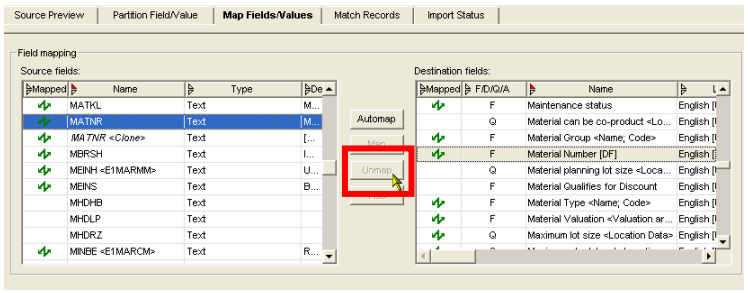
When using the Import Manager, the user sometimes come to a situation where the system does not give any tip to solve the problem. For example, you cannot unmap a field, or you don't know why you can't fill qualified table content. We will here explain why and give instruction to solve that kind of problem based on the import of a XML file. We will not see other kind of data source, at least in this first release of the document. A general recommendation is to use the latest release and build available in the SAP Marketplace as you may encounter some errors corrected in the latest build.
Here is the list of troubleshoot explained:

- Unable to unmap some already mapped field (under some conditions)
- Unable to import a hierarchy together with the import in main table
- Why sometimes the "map" flag option can be checked for displayed text fields
- Unable to import data in qualified table with multi non-qualifier fields displayed
- How to recreate automatically missing fields in XI for qualified tables with multi non-qualifier fields
- How to update qualified content of records by matching option for qualified table content to add records or merge informations in existing qualified content
- Missing source elements as table when opening an XML file without XSD schema
- Incorrect XML regarding the XSD schema selected (how to solve it)

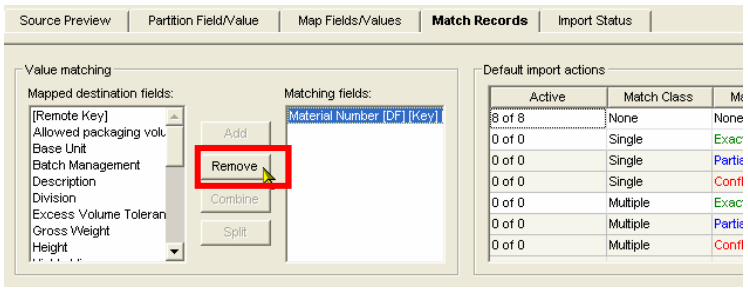
3 The Step By Step Solutions…

3.1 Unable to unmap a field in the matching tab

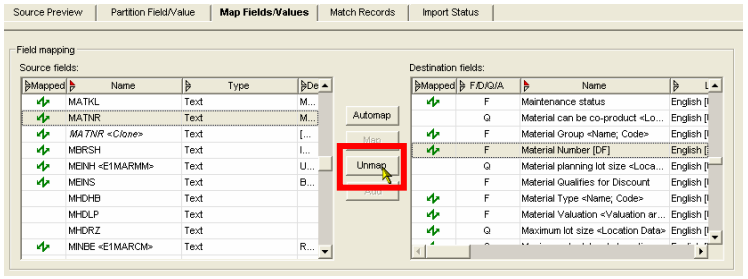
1. As shown here, the “unmap” function is disabled: you are not allowed to change this field mapping. There are two possible reasons :
- the target field is used in the “match records” tab to find matching records in the repository. (see § 2)
- the target field is used to recreate a compound field for a qualified table mapped (see § 4 in this section)



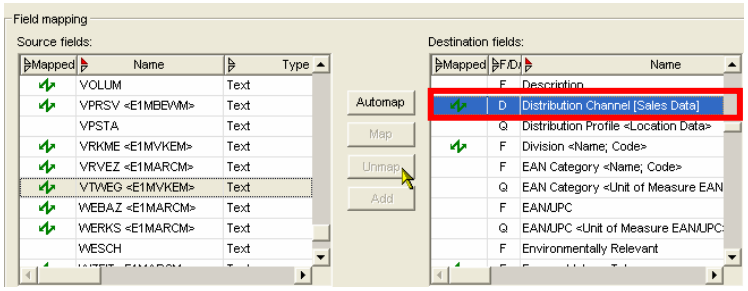
2. If the target field is used in the “match records”, you must first remove the matching field from the list in the “Match records” tab to enable the unmap function.



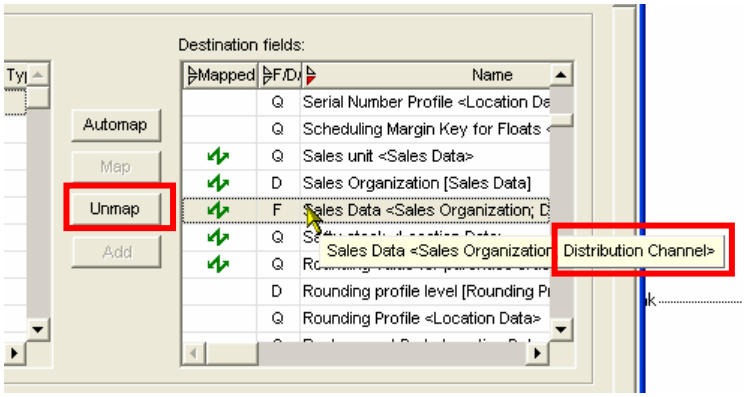
3. Then the option to unmap is enabled again: you can change your mapping, in this case the source field for the material number. Don’t forget to update your matching fields list again after.



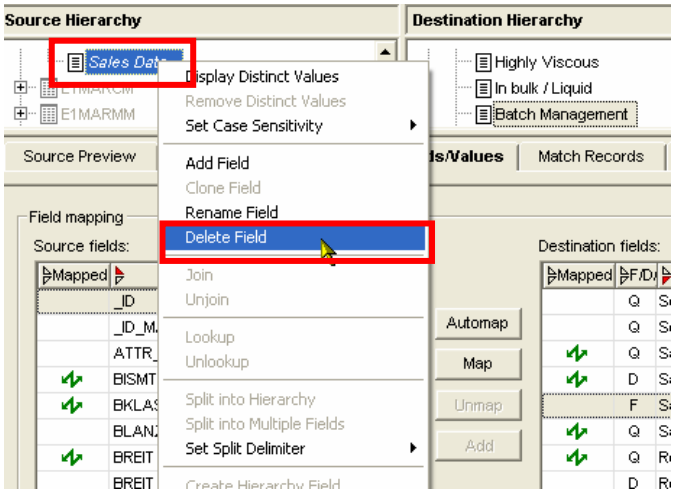
4. If the target field (here VTWEG) is used in a recreated compound field for qualified content, then you can’t unmap it. Please notice the target field to find the related field in main table you have to unmap (here “Distribution Channel”)



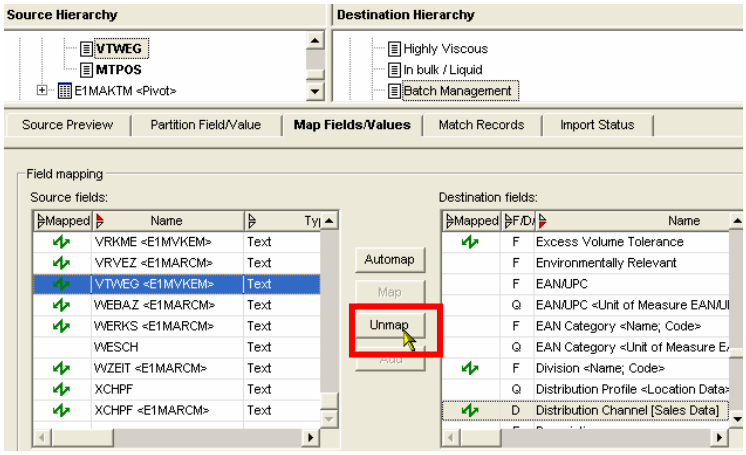
5. To unmap it, you must first find the right compound field relying on it and unmap it first. Just look for the F field mapped to the locked field in the previous step, and unmap it. This step is need in order to delete the compound field in the next step



6. Then look for the compound field in the source hierarchy which should have the name of the previously unmapped qualified table (here “Sales data”), right click the source hierarchy structure and choose the delete field option :



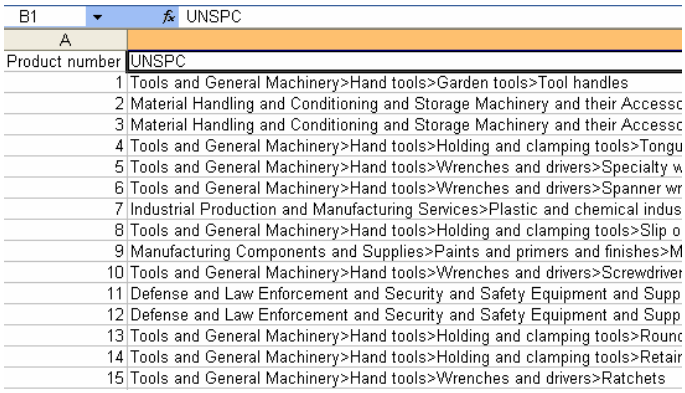
7. Now the VTWEG field is no more used in a compound field, so you can go back to the previously mapped field you wanted to unmap : the unmap action is now enabled.



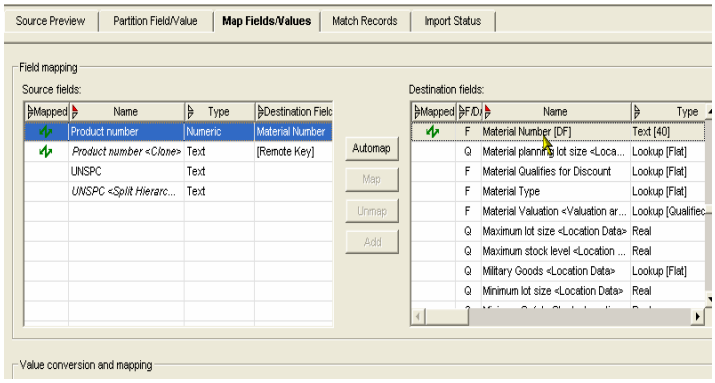
3.2 How to import a hierarchy within main table import

Importing assignment of products to hierarchy can be easily done once the hierarchy has been already loaded into the repository. In this case, the load of hierarchy sub-table is performed separately before with hierarchy as target structure in the import manager. Nevertheless, it's also possible to enrich the hierarchy sub-table during the import of main table records:

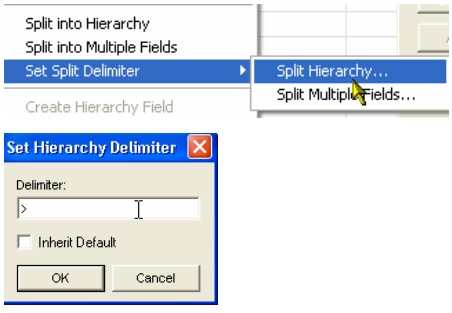
1. Please be careful with file format prerequisites: the category (leaf node of the hierarchy) must be concatenated to the full hierarchy path to access it. It can be difficult or impossible to recombine it otherwise within the import manager. Pay attention to the used separator which may be customized in the import manager (here ">")



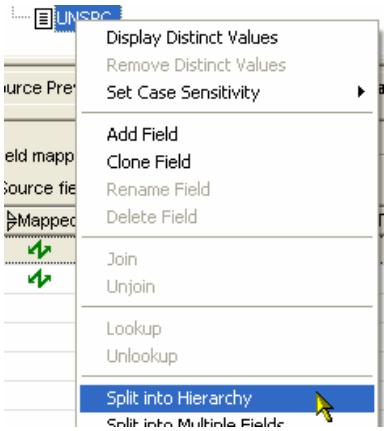
2. Map at least the display field(s) of the main table to be able to match records for the match record step



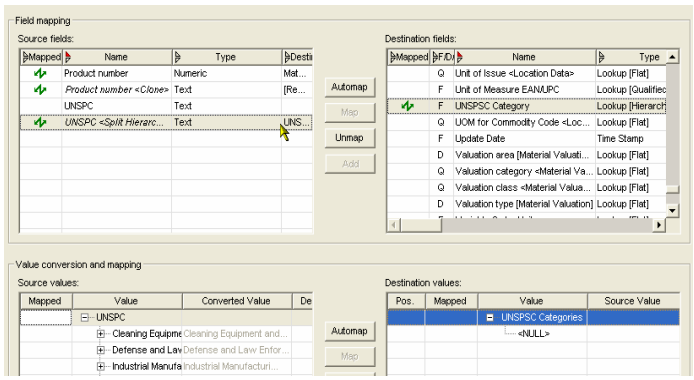
3. Right click on the field containing the hierarchy info and select the split hierarchy option : the delimiter should be the one used in the source, in our case ">".



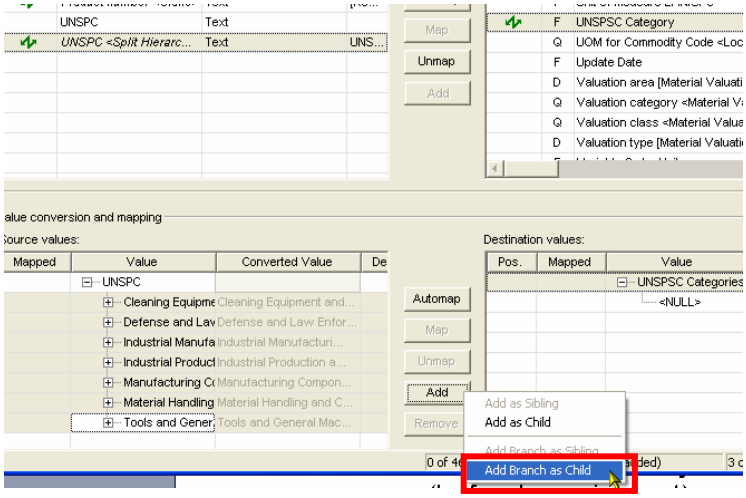
4. Right click again on the field containing the hierarchy info and select the split into hierarchy. A new field with the hierarchy well represented is then created.



5. Map the created hierarchy source field to the main table target field storing record's hierarchy assignment



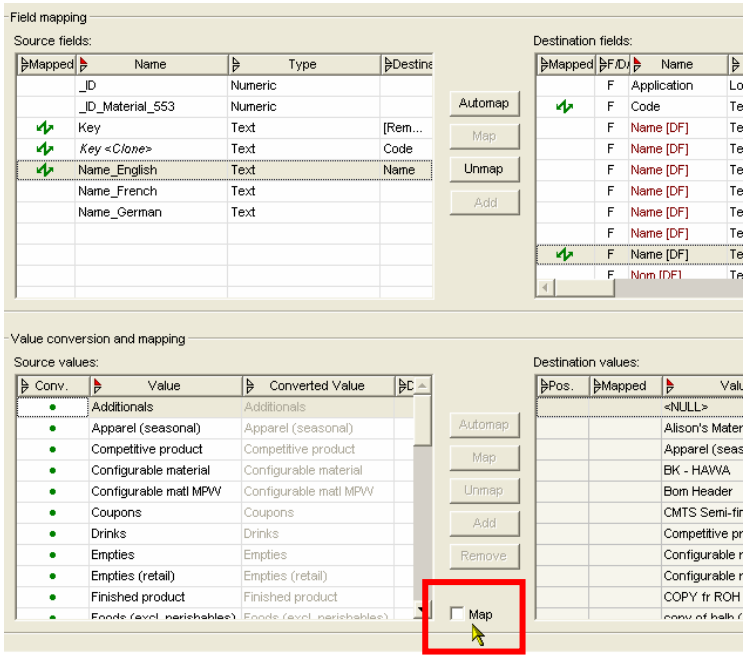
6. To add the full hierarchy and values (leaf nodes assignment), **select all nodes except the root in the source values list**, click on the "add" button, and choose "Add Branch as child" if you have selected the root in the destination value. Note that only leaf nodes have a blue arrow to show value mapping assignment exist, but the import manager will create the full hierarchy during the import run of main table. You can then go to the matching tab to choose the matching fields you want, and even use the hierarchy assigned field as a matching criteria. Then run the import : records will be created in main table, the hierarchy will also be created including the assignment of main records to the correct leaf nodes.



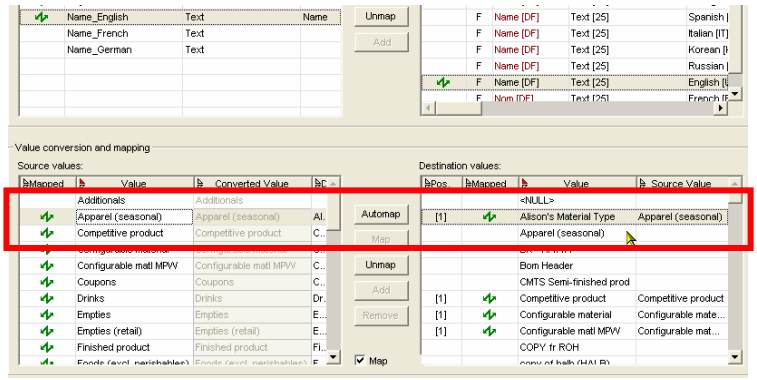
3.3 Map flag option : when to activate and consequences

1. In some cases, it's possible to activate manually the value mapping like in this screen copy. It happened only in very few conditions :
- text display field on lookup table
- promoted text lookup on display field [DF].

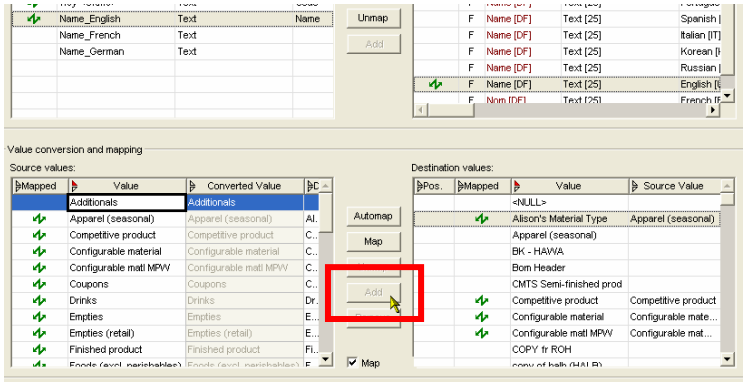
Furthermore, ensure that target field is not yet used as matching field, otherwise the checkbox will be grey out and change not possible.



2. Activating this flag enable you to choose manually the mapping of values (done automatically otherwise). In this screen copy, we see that we can map to a different value even if the same value already exist in the target field list.



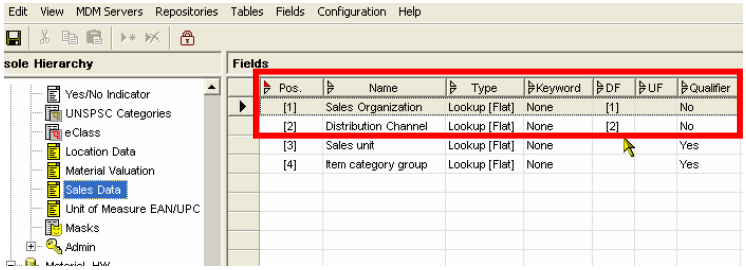
3. Nevertheless, it's not possible to add values in the destination value if it corresponds to the target table. As it is mandatory to match all sources values of mapped fields, it can be a problem if new values are to be created. The only workaround is to map to an other existing value, to Null, or to create the value manually with the data manager



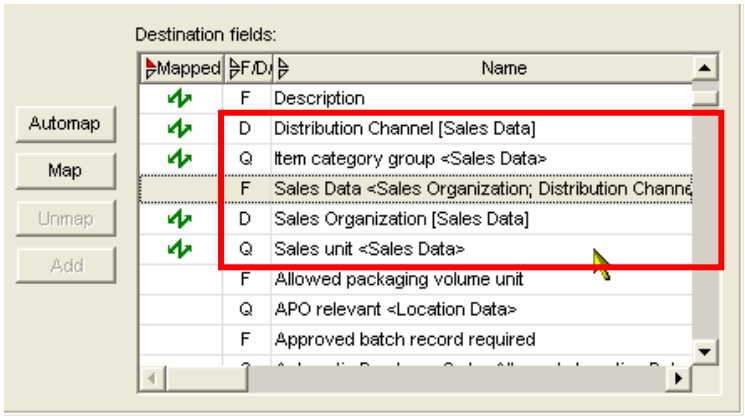
3.4 Importing data in qualified table with multiple non qualifiers fields

In more complexes data models, qualified tables can be defined with more than one non qualifier field also with “display field” property enabled: for example, sales data in standard material repository have 2 non qualifier fields, sales area and distribution channel which are also display fields. We see here how to fill up this table during the import of main table records.

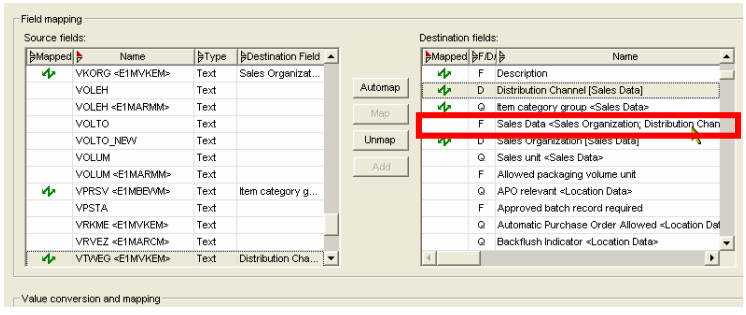
1. To be able to create the right record in qualified table with the right combination the system must be able to build up the complete key according to display field. Therefore, you can check what fields are needed in the console : these fields will be displayed in the import manager when importing in the main table because they are display field.



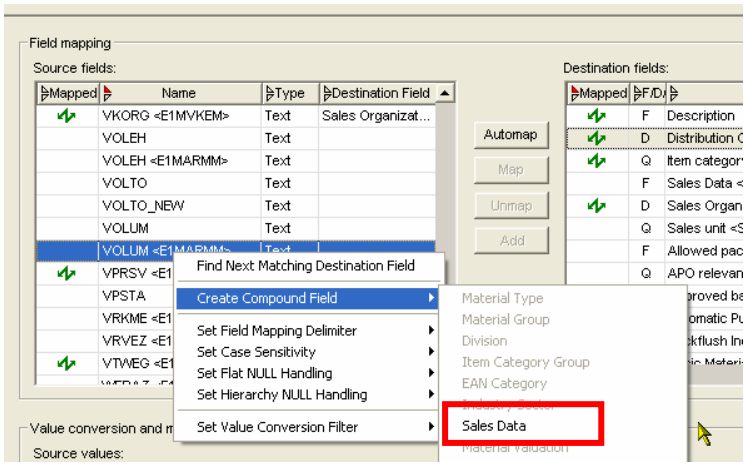
2. You can retrieve the same information in the import manager thanks to the property of fields :
- “D” stands for display fields, including in lookup tables
- “Q” stands for qualified fields, i.e. fields in qualified table with property “qualifier=Yes”
- “F” is considered as a normal field of target table, here the main table. It also shows the key (display fields) of the referenced qualified table if this field is a qualified flat in the repository definition



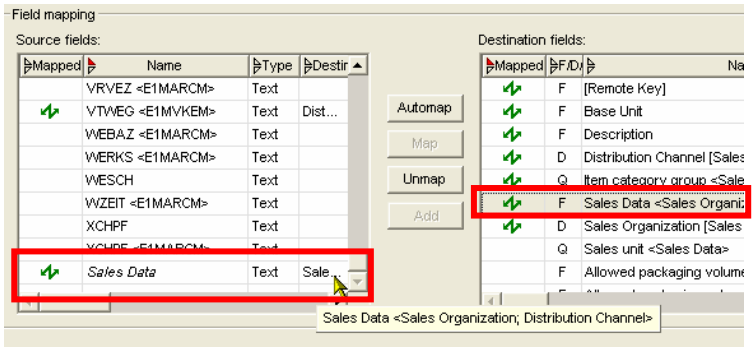
3. Minimum required field mapping : you must map at least on field of the main table and all display fields (D) of the qualified lookup table. Qualifier field (Q) are optional. Nevertheless, it is not enough as you have to recreate the combined fields of the qualified table (here Sales Data)



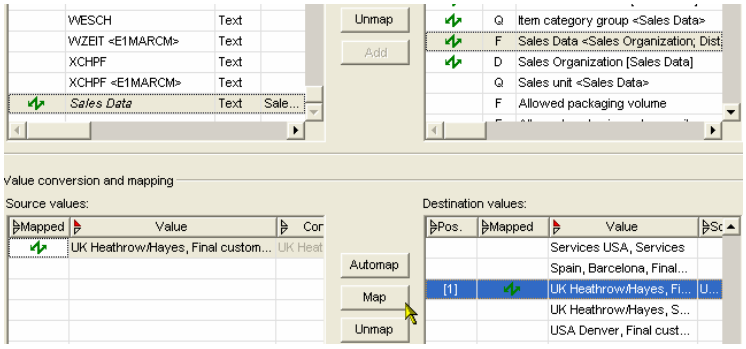
4. Once this prerequisite is done, you can recreate the compound field with a right click in the right side (source field). The “Sales Data” entry is the only entry you can select because it’s the only one which has all its related display fields already mapped. If one of the display fields of the table is not yet mapped, you are not allowed to recreate this compound field.



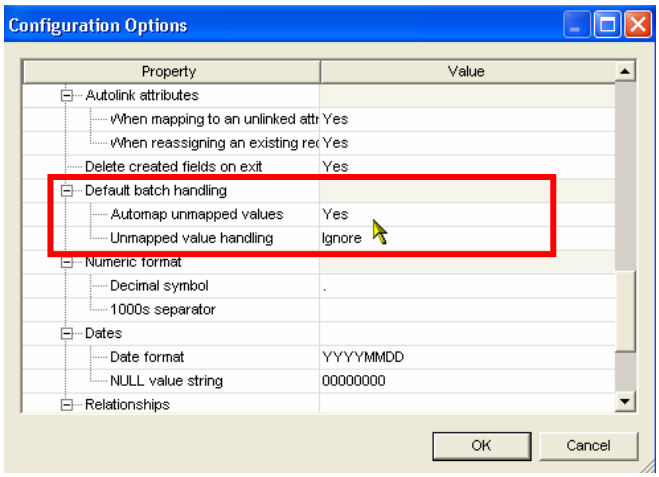
5. The import manager has then automatically created a new source field with the name of the target qualified table (“Sales data”) and mapped it automatically to the “Sales Data” target field (F)



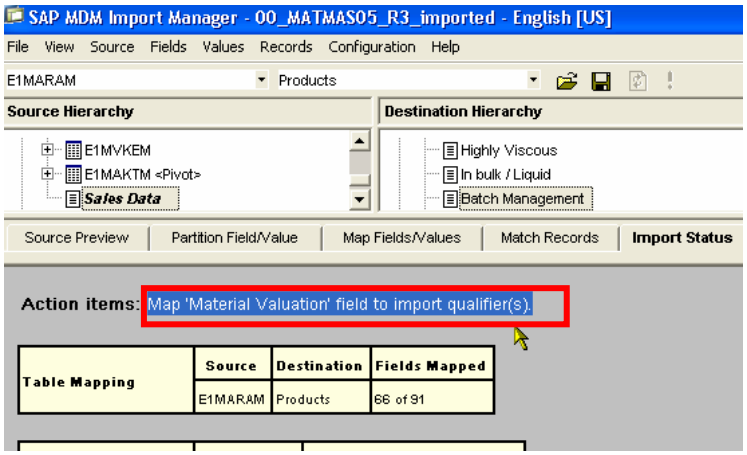
6. You can use the Automap function or map manually the values, combinations of displayed fields reused to create the compound field. Please note that up to and including release 5.5 SP3, Automap action is not saved in the mapping : you have to perform it manually for this kind of qualified table with multi non qualifiers-displayed fields every time you perform an interactive import.



7. For batch handling, you can check and activate the default batch handling option to avoid missing mapping in such cases.



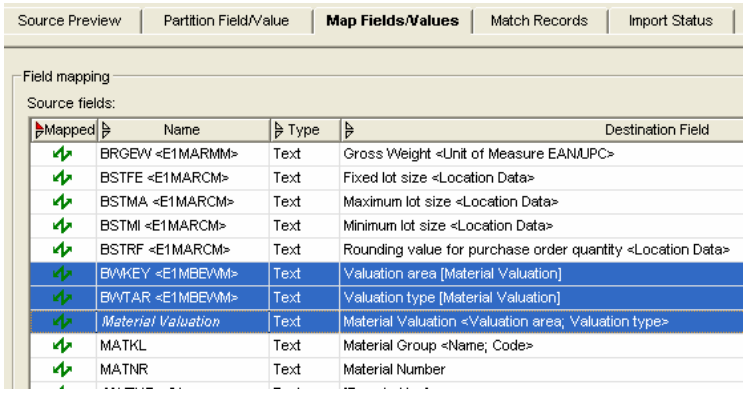
8. Further note if incoming message with one source field mapped to qualified table display field is missing: in this case, even if the mapping had saved all the mapping rules, the import manager will not be able to recreate the compound field as showed here. You must then ensure that all display fields are available (even as empty tag for XML) in the source file. Please check part 3.6 of this document to know how to recreate the field in XI mapping.



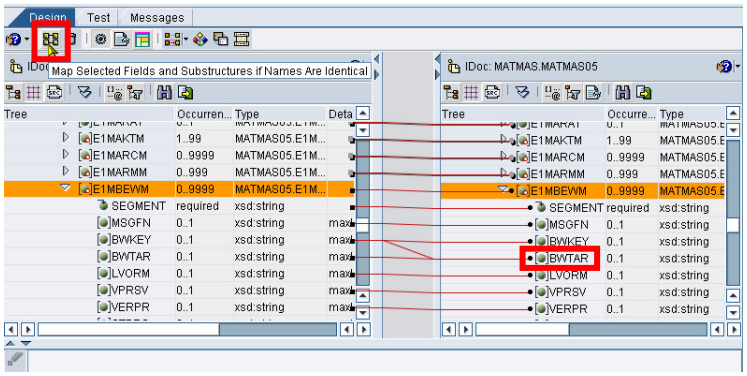
3.5 How to recreate missing empty fields for import in qualified tables with multi non qualifier fields

As show in part 3.4, if the qualified table get more than one non-qualifier field, you must map all of them to be able to recreate the compound field. It happens that one of these fields can be empty. In that case, following the source generator, the generated XML file often contains only tags which are not empty : you then have to recreate the missing tag if the other field used as non qualifier in the target table is existing in the source file. Basic knowledge in XI mapping is here required to define the mapping, but XSLT transformations or any EAI tool should be able to execute the same task.

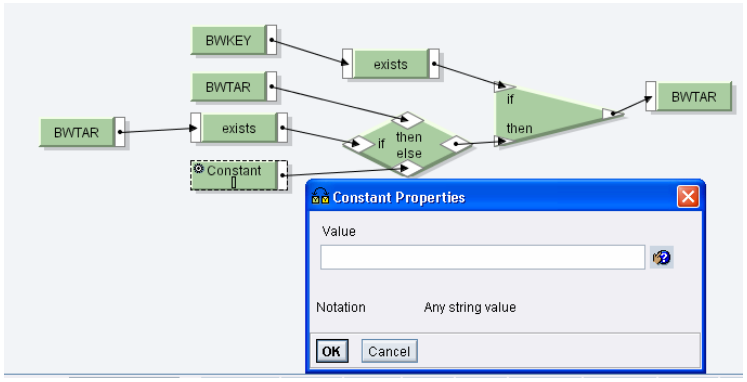
1. First of all, check in a normal import (with all fields filled) which fields in the XML file are used as source field for the non qualifier fields in the target qualified table. In this example, the field BWTAR (material valuation type) can be empty, so we must ensure in mapping that if BWKEY is provided, BWTAR must also exist or being recreated as empty tag otherwise.



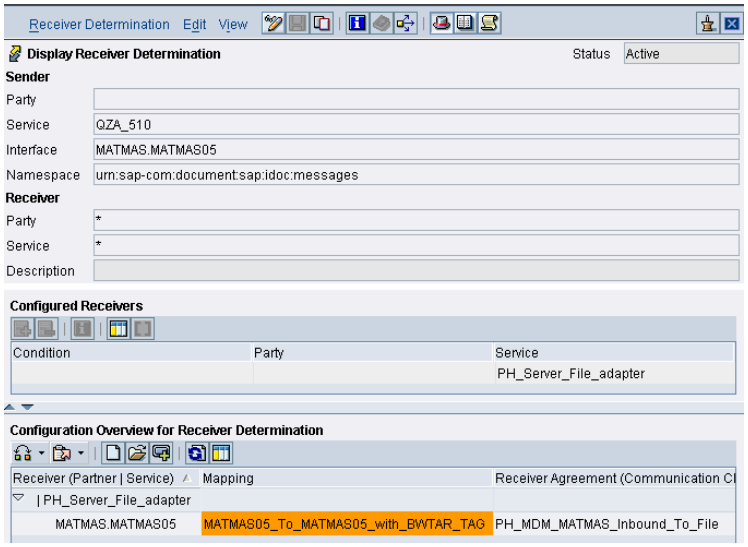
2. Open the Integration Builder to create a new message mapping in XI for the used interface (here MATMAS05). Use the same imported XSD schema as source and target interface. Use the Automap feature to automatically map all elements in XI. Then double left click on the BWTAR target element.



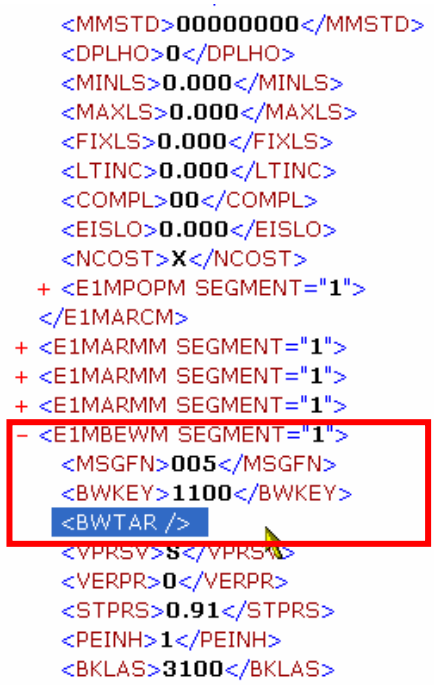
3. Use the graphical mapping tool to define this similar mapping according to the fields implied in your scenario. It ensures in that example that if BWKEY tag exists, it check that BWTAR also exist or recreate the tag with the designed constant (here empty). In any cases, this mapping ensure that you'll always get both tags or none (because we know that only BWTAR can be optional in the source system)



4. Configure then XI to use this mapping by defining an interface mapping. Open the XI Integration Directory to use this interface mapping in the business process definition (interface determination step)



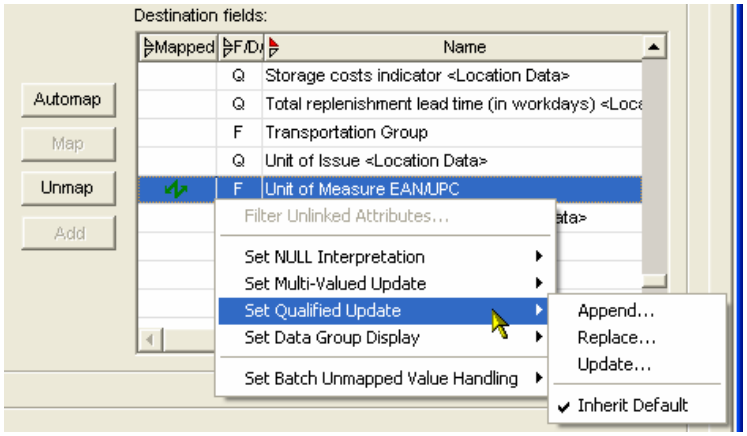
5. As a result the XML file includes now all the time all needed fields for the qualified content: the compound field can be recreated for all incoming messages. Nevertheless, you have to ensure that combinations with one empty field value also exist in MDM repository, meaning you may have to add in the qualified table the combination of BWKEY=1100 and BWTAR=Null…



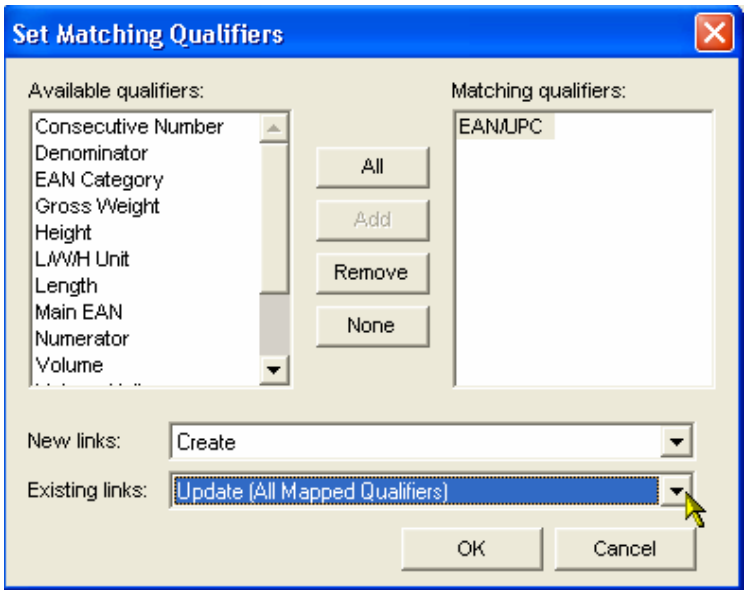
3.6 Updating existing qualified records

When the repository already contains some data with qualified content, it's needed to match not only main table records during import, but also qualified table related content to add or update existing qualified data. It may also happen that you have different update strategy for different qualified table during the same import to main table in the same repository. We assume here that the field mapping is already done and will show how to configure the update method for qualified table.

1. Navigates to the target field mapped (Type F) representing the display fields combination of the qualified table (in particular if more than one non qualified field). Right click on it: In the "Set qualified Update" option, if it is a qualified field, you can overwrite for this qualified table the default update method defined in the Import manager configuration options.



2. Choosing "Update…" method enable you to decide how the system react on the defined matching qualifiers. Matching qualifier list allows you to extend the standard matching based on normal non-qualifier display fields: you can add other fields (which have property qualifier=Yes) to get a different result. According to the matching fields you choose, the system will then create new qualified content, or update existing qualified content.
Each qualified table can then have a different update strategy for the same import run : replace, append, or update.

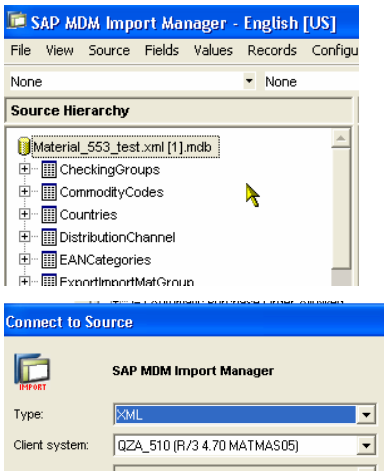


3.7 Missing source structure when opening XML file without XSD specification

When you do not specify any XSD definition for the opened XML file, the system will not recognize structures with only one occurrence as shown in the first step. We then illustrate how to create a simple XSD file to ensure that the structure can be recognized even if only one occurrence of element exist in the file to import

1. As shown here, despite BasicMaterials tag occurrence, it is not displayed as a source field when using simple XML without schema.

```
<LotSize>
  <Key>ZX</Key>
  <Name_German>Exakte Losgrößenberechnung mit Splittung</Name_German>
  <Name_English>Lot-for-lot order quantity w. splitting</Name_English>
  <Name_French>Calcul exact des lots avec fractionnem.</Name_French>
</LotSize>
<BasicMaterials>
  <Key>SPHAEROGUSS</Key>
</BasicMaterials>
</Material_553>
```



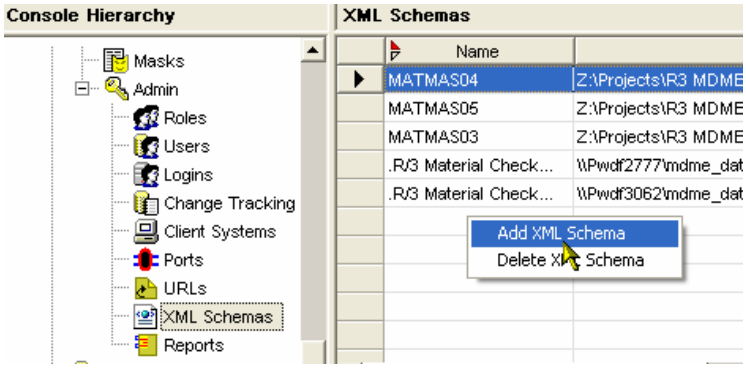
2. To avoid this kind of behavior, you must define a simple XSD file: this definition will inform the Import manager XML parser that this element have sub-elements and can occurs more than once. Import manager will then consider it as a source table even if only one entry is present. Here, the maxOccurs attributes must be greater or equal 2, or can be unbounded (most cases…)

Important note : You must define all possible tags that may be present in the XML file. You can purchase third party tools to generate a XSD from an XML instance, or use Xsd.exe command line from Microsoft .Net Framework

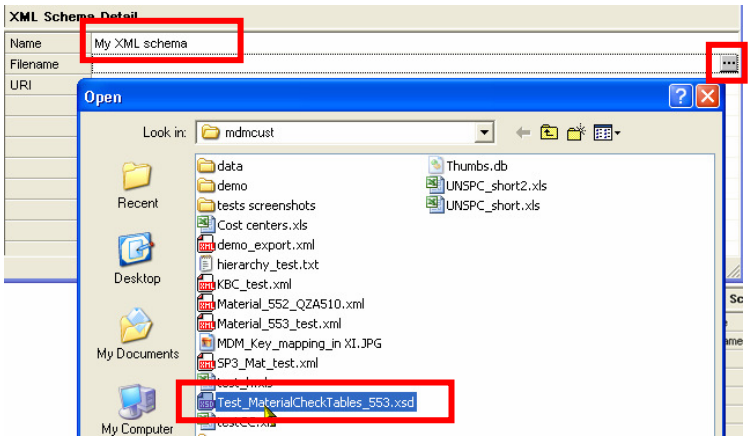
If you are not familiar with XSD, a short complete sample is provided in the annexes of this document

```
<xsd:element name="BasicMaterials" minOccurs="0" maxOccurs="unbounded">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="Key" minOccurs="0"/>
    </xsd:all>
  </xsd:complexType>
</xsd:element>
```

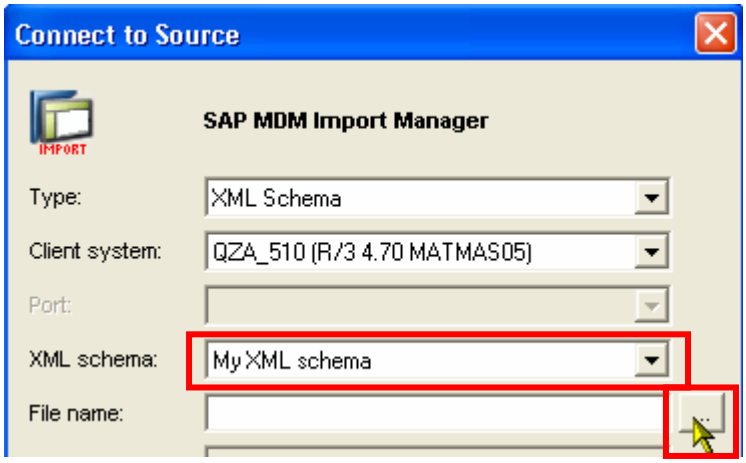
3. Then import this XSD into the repository using the console interface : navigate to the admin tree to select the XML Schemas, and add with the right click context menu
Note: the repository must be unloaded to perform this operation



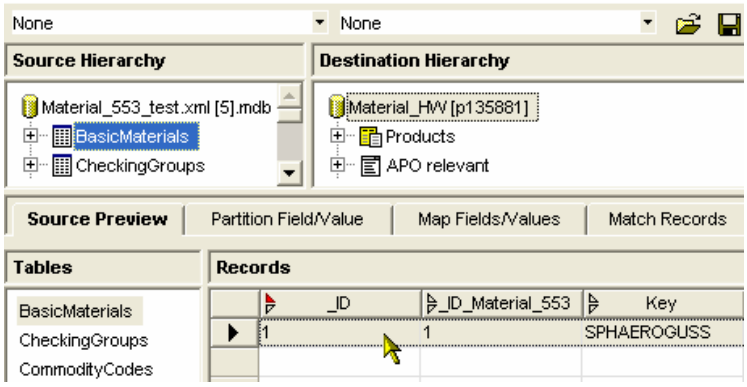
4. Input name (Will be displayed in the Import manager UI), and browse to the created XSD file to load it into the repository. By loaded, we mean that file have been copied into the repository and the MDM server no longer needs to access the file system to get this XSD..
Important note : if you change the XSD definition content, change the filename too: the console cannot detect a change in the configuration if you do not change the file to load, so the XSD will not be uploaded again in the repository.



5. Now Load the repository again with the console, and restart the import manager using the newly created XML schema, and browse to the previously opened XML file



6. As presented here, the BasicMaterials shows up as a possible source structure, even if there is only one record in the source file



4 Appendix

1. XSD sample template for XML file with multiple table structures.

Note : you must ensure the sequence of tags in XML file is in the same order as described in the XSD file : records for MyTable1 are before records for MyTable2.
To check and help you on your XSD creation and XML validation against this XSD, you can download free tools like Altova XMLSpy Home edition, or Cooktop, a very tin and free product available at http://www.xmlcooktop.com/

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
	<xsd:element name="My_Full_data_group">
		<xsd:complexType>
			<xsd:sequence>
				<xsd:element name="MyTable1" minOccurs="0" maxOccurs="unbounded">
					<xsd:complexType>
						<xsd:all>
							<xsd:element name="MyFieldT1_1"/>
							<xsd:element name="MyFieldT1_2"/>
							<xsd:element name="MyFieldT1_3"/>
						</xsd:all>
					</xsd:complexType>
				</xsd:element>
				<xsd:element name="MyTable2" minOccurs="0" maxOccurs="unbounded">
					<xsd:complexType>
						<xsd:all>
							<xsd:element name="MyFieldT2_1"/>
							<xsd:element name="MyFieldT2_2"/>
							<xsd:element name="MyFieldT2_3"/>
						</xsd:all>
					</xsd:complexType>
				</xsd:element>
			</xsd:sequence>
		</xsd:complexType>
	</xsd:element>
<xsd:schema>
```

2. Free Xsd.exe generator tool from Microsoft to generate XSD from XML instance

You can download from Microsoft SDN http://msdn.microsoft.com/downloads/ the .Net Framework and the .Net Framework SDK (Software development kit) to get this executable file. Then the full documentation of Xsd.exe use is available on http://msdn.microsoft.com/library/en-us/cptools/html/cpconxmlschemadefinitiontoolxsdexe.asp

This tool can help you to generate most of the XSD content, so that you just have to update the required tags to ensure the MaxOccurs attribute is unbounded or greater than one.

3. Free XML and XSD documentation

http://www.w3.org/XML/

http://www.w3.org/XML/Schema

www.sdn.sap.com/irj/sdn/howtoguides