

SYNDICATING HIERARCHIES EFFECTIVELY

Applies to:

SAP MDM 5.5

Summary

This document introduces hierarchy tables and a method of effectively sending out data stored in hierarchy tables.

Created on: 18 May 2006

Author Bio



Anand Chandrashekar

Anand is an associate consultant employed with Wipro Technologies. He has over 6 years of experience as an ABAP consultant. He has worked on master data distribution projects in the R/3 world using Idoc technology. He has recently been working on MDM related projects.

Table of Contents

The Hierarchy Table	3
Maintaining the hierarchy table	3
Our example-Cities	3
Record Mode.....	3
Hierarchy mode	4
Exporting Hierarchies	5
Data Manager	5
Syndicator	6
Syndicating to XML.....	7
The Schema.....	7
Launching and Configuring the Syndicator	7
Data Manager Export File	7
Creating a Merge Item	8
Mapping Merge Item Country	8
Appendix.....	9
Regions Schema Definition.....	9
Related Content.....	10
Disclaimer and Liability Notice.....	10

The Hierarchy Table

The hierarchy table is standard table type provided within SAP MDM. As the name suggests, these tables store data and the hierarchical relationship within. Contents of the hierarchy table can be created online via Data Manager or be imported by Import Manager. For exporting, options are available in the Data Manager and Syndicator.

Exporting and importing hierarchies are complex activities as the applications involved need to take care of passing not just information about the data elements in a hierarchy, but also the relationship linkages.

Let's first have a look at maintaining a hierarchy table and then move towards exports.

Maintaining the hierarchy table

MDM provides two modes to view and manipulate data within the hierarchy table. These modes, 'Record' and 'Hierarchical', are available in the MDM Data Manager.

Our example-Cities

Let's take an example of cities. This is a triplet relation of country-state/province-city. For example, 'Germany-Hesse-Frankfurt' is one such triplet. In order to maintain these kinds of hierarchies, a hierarchy table 'Cities' needs to be created. It will have just one field in it, 'Name'.

Field Detail	
Name	Name
Type	Text
Required	No
Writable Once	No
Matrix	Normal
Multilingual	Yes
Keyword	Normal
Display Field	Yes
Unique Field	No
Calculated	
Calculation	
Width	50
Sort Type	Case Insensitive

Fig 1 – Field Detail of the 'Cities' Table

All required entities; country, state/province and city need to be first directly loaded into this table. Thus, the entities 'Germany', 'Hesse' and 'Frankfurt' become individual entries in this table.

Record Mode

The record mode is primarily concerned with the storing and viewing of individual elements of the hierarchy. It does not cover the relationship within the elements. Data manager will display the individual elements in a linear form as shown below. Addition, modification and deletion of individual records are possible in this mode.

Records	
	Name
	Australia
	Australian Capital Territory
	Canberra
	Cape Town
	Frankfurt
	Germany
	Hesse
▶	India
	Maharashtra
	Mumbai
	New South Wales
	Pune
	South Africa
	Sydney
	Western Cape

Fig 2 – Record Mode

Hierarchy mode

In the hierarchy mode, the hierarchical relations can be specified. This can be achieved by simple drag-and-drop actions on individual data elements. While drag-and-drop operations, options for adding a node as child or sibling will be provided. Once constructed, the hierarchy looks as follows.

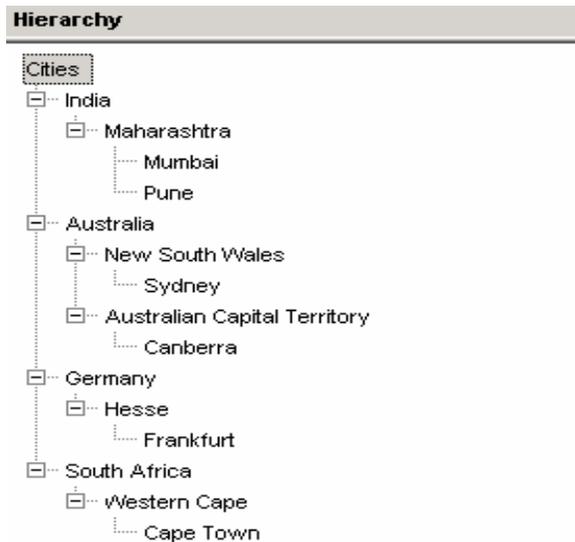


Fig 3 – Hierarchy Mode

Exporting Hierarchies

Once hierarchies have been created within MDM, it will be useful for external systems to have the same hierarchical data. Here are the options available to export data from MDM.

Data Manager

In the record mode, the Data Manager provides an export action in the file menu. It is possible to export to MS Excel, MS Access or text formats. Here are the options when an Excel output is chosen.

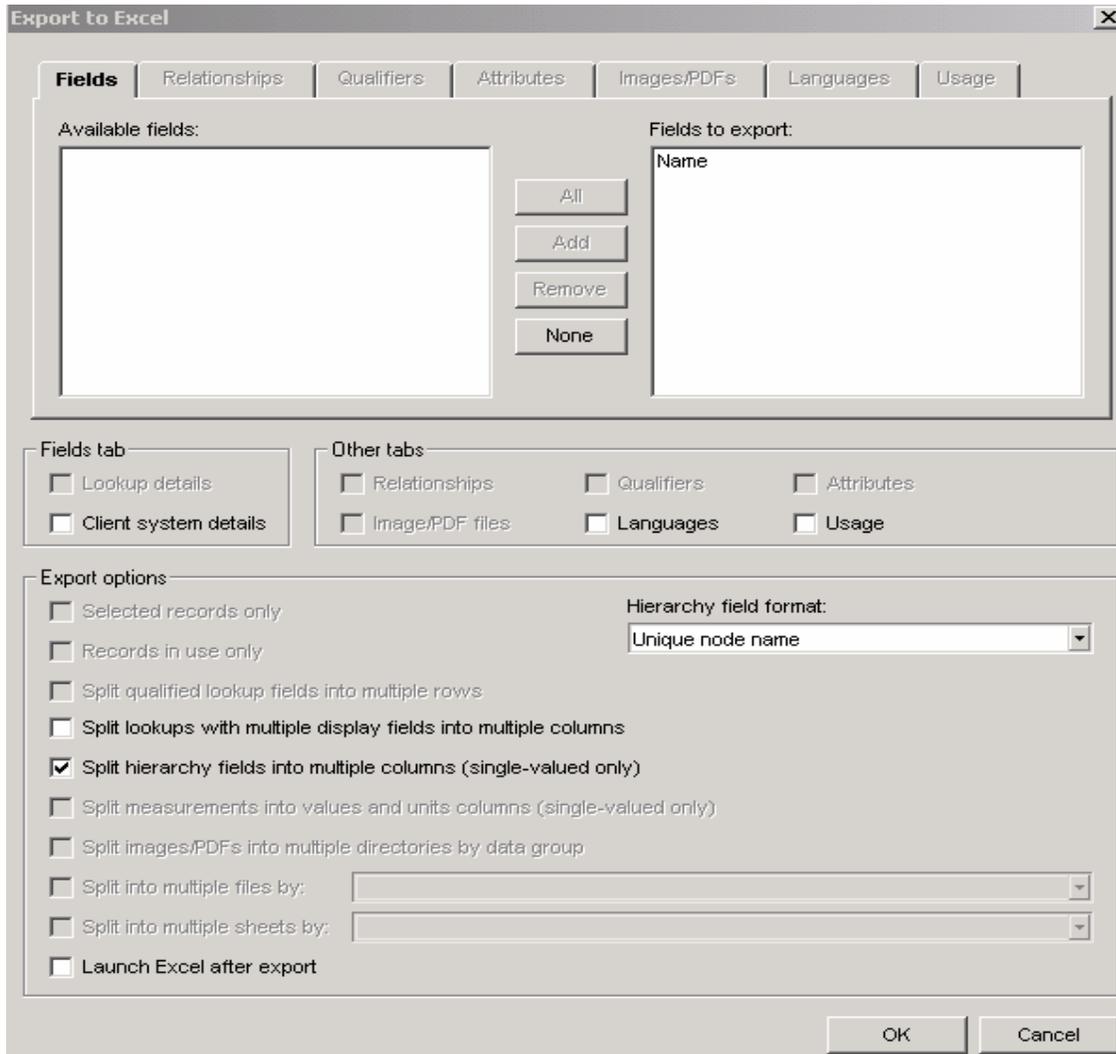


Fig 4 – Export Options in Data Manager

By default, the export will only send the field contents and not the hierarchical relationship. If the 'Split hierarchy into multiple columns' option is not selected, you will find the output as in Fig 5.

1	Name
2	Australia
3	Australian Capital Territory
4	Canberra
5	Cape Town
6	Frankfurt
7	Germany
8	Hesse
9	India
10	Maharashtra
11	Mumbai
12	New South Wales
13	Pune
14	South Africa
15	Sydney
16	Western Cape
17	

Fig 5 – Plain Download

	A	B	C
1	Name [1]	Name [2]	Name [3]
2	India	Maharashtra	Mumbai
3	India	Maharashtra	Pune
4	Australia	New South Wales	Sydney
5	Australia	Australian Capital Territory	Canberra
6	Germany	Hesse	Frankfurt
7	South Africa	Western Cape	Cape Town
8			
9			

Fig 6 – ‘Split Hierarchy’ Download

By selecting the option mentioned above, it is possible to get a file which is more sophisticated. Such an output will contain not just data but also an indication of the relations within (Fig 6).

Syndicator

Contents of the Cities table can also be syndicated using SAP MDM Syndicator. However, Syndicator does not have the capability to syndicate relationships between elements in the Cities table. A typical text output from Syndicator will look as below.

```

cities.txt - Notepad
File Edit Format View Help
Name
Australia
Australian Capital Territory
Canberra
Cape Town
Frankfurt
Germany
Hesse
India
Maharashtra
Mumbai
New South Wales
Pune
South Africa
Sydney
Western Cape

```

Fig 7 – Syndicated Output

Note that the option for splitting hierarchy from root or leaf is available for a hierarchical field in a main or sub table, not for the hierarchy table itself.

Syndicating to XML

As seen above, it is possible to export hierarchies; both data and relationships. For exporting relationships from Data Manager, the output media can only be one of the following types: text, MS Excel or MS Access. Data Manager will not be able to export to a port or XML format and so on. The Syndicator, on the other hand has the capability to export to ports and XML files, but doesn't have the native capability of exporting relationships. As we have seen, it can only export data.

In order for Syndicator to export to an XML, we will have to mix-and-match the output possibilities available in MDM. Here is an example of syndicating hierarchies to an XML file.

The Schema

Let's create a region schema which stores the triplet. Here is the visual representation.

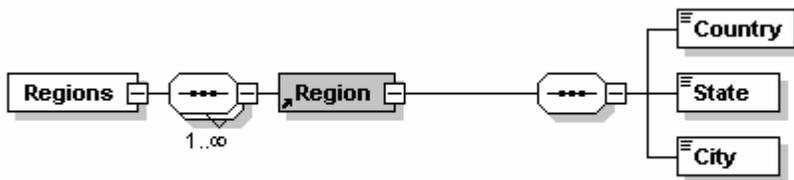


Fig 8 – The Output Schema

The full schema definition file is available in the appendix.

Launching and Configuring the Syndicator

Launch the Syndicator and select the destination properties as below.

The screenshot shows the 'Destination Properties' dialog box with the following fields:

- Type: XML Schema File
- Client system: MDM
- Port: (empty)
- XML schema: (empty)
- XML file name: D:\MDM\XML\Regions\Regions.xsd
- Root: Regions

Buttons for 'OK' and 'Cancel' are visible at the bottom.

Fig 9 – Destination Properties

The source table for Syndication needs to be set as the 'Cities' table. By marking the 'Region' node as a repeatable node, we can map the high level 'Cities' table to the 'Region' node. This will make sure that all the contents of the source table flow into a single XML file.

Data Manager Export File

To go further, we will need a file exported from data manager which contains both data and relationships. We have seen this earlier in the form of an MS Excel file. The same option needs to be used again; this time to create a text file. Let's name this file as 'Cities.txt'. We will be using this file as a source for 'Merge Items'.

This file will be similar to the MS Excel file we created previously. It will have 3 columns denoting the triplet relation. 'Name[1]', 'Name[2]' and 'Name[3]' correspond to country, state/province and city respectively.

Creating a Merge Item

Create a merge item 'Country' targeting the Cities table as follows. The 'Name[1]' corresponds to the Country field and this needs to be mapped to the 'Name' field in the Cities table. All fields thus merged are selected. Here is how the merge item looks.

Merge Items	Properties												
Country	<table border="1"> <thead> <tr> <th>Name</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Country</td> </tr> <tr> <td>Table</td> <td>Cities</td> </tr> <tr> <td>Source File</td> <td>C:\Documents and Settings\chandrd\Desktop\Cities.txt</td> </tr> <tr> <td>Matching Fields</td> <td>Name [1]-Name [DF]</td> </tr> <tr> <td>Source Fields to Merge</td> <td>Name [1]; Name [2]; Name [3]</td> </tr> </tbody> </table>	Name	Value	Name	Country	Table	Cities	Source File	C:\Documents and Settings\chandrd\Desktop\Cities.txt	Matching Fields	Name [1]-Name [DF]	Source Fields to Merge	Name [1]; Name [2]; Name [3]
Name	Value												
Name	Country												
Table	Cities												
Source File	C:\Documents and Settings\chandrd\Desktop\Cities.txt												
Matching Fields	Name [1]-Name [DF]												
Source Fields to Merge	Name [1]; Name [2]; Name [3]												

Fig 10 – The Merge Item Country

Mapping Merge Item Country

Map the Name fields to country, state and city nodes of the XML tree. Here is how the map looks.

Source items:				Destination items:	
Name	Type	Mapped		Mapped	Name
Cities	Hierarchy		Region		Regions
Name [DF]	Text [50]				Region
Country	Merged Source				Country
Name [1]	Merged Field		Region		State
Name [2]	Merged Field		Region		City
Name [3]	Merged Field		Region		

Fig 11 – Mapping the Merge Item

We are now ready to Syndicate. Press the syndication button. The output XML file will contain all the triplets. Here is a snapshot of the output.

```

- <Regions>
  - <Region>
    <Country>Australia</Country>
    <State>New South Wales</State>
    <City>Sydney</City>
  </Region>
- <Region>

```

Here are the steps performed by merging. Each record is picked up from the Cities table and compared with individual records in the file. Only the 'Name[1]' field i.e. country field in the file is used for comparison. Wherever a match is found, the corresponding state/province and city fields in the file is picked up and sent to the XML output.

It is seen that there are some empty nodes in the output XML file. This is because the non-country entities in the Cities table do not match with any records in the merge file. The receiver of this file will need to accommodate that as this cannot be avoided using this method.

Appendix

Regions Schema Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2006 sp2 U (http://www.altova.com) by Anand
Chandrashekar (Wipro) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="Region">
    <xs:annotation>
      <xs:documentation>Comment describing your root
element</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Country" type="xs:string"/>
        <xs:element name="State" type="xs:string"/>
        <xs:element name="City" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Regions">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element ref="Region"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Related Content

Syndicating Qualified Master Data Fields

<https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/792f57b7-0a01-0010-f3b6-881269136a83>

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.