

**How-to Guide
SAP NetWeaver '04**



How To handle Data Modeling in MDM 5.5

Version 1.01 – Mai 2005

**Applicable Releases:
SAP NetWeaver '04
Master Data Management 5.5 (MDM 5.5)**

© Copyright 2004 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, and Informix are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data

contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided "as is" without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

1	(Business) Scenario.....	1
2	Introduction.....	1
3	The Step By Step Solution.....	3
3.1	Analyze the source information (see data file above).....	4
3.2	Create the catalog.....	5
3.3	General Remarks.....	7
3.4	MDM 5.5 Modeling general tips	11
3.5	MDM 5.5 Taxonomy tips	13

1 (Business) Scenario

How to design an MDM 5.5 catalog?

This script tries to describe the process of planning of an MDM 5.5 catalog's structure:

1. based on the knowing of all the information available in the product data base of the customer and
2. based on the customer requirements

2 Introduction

A catalog is more than a large collection of data regarding the products. A usable and multifunctional catalog demands a good and rich structured product content to support both the company's sales goals and the customer's purchasing requirements.

This guide describes general guidelines for designing an MDM 5.5 catalog. There are no hard and unique rules which should be applied by the design of a new catalog, and good judgment is involved mostly in every step of designing. This good judgment increases with the experience in designing.

A good designed catalog should have these features:

1. it is easy to be maintained (for the owner of the catalog),
2. it is easy to be searched (for the customer)

The information available in the product data base of the customer (with other words the source information for the designer of the catalog) needs to be separated into two parts:

1. Information which applies to all product records in the catalog, independent from the category of the product.
Information common to all product records would be candidate for **fields** in the main table of the catalog.
2. Information which applies only to certain categories of products and are qualities that exist only for some products.

For example, in a catalog which contains both furniture and electronics, some types of information from the data base of the customer will apply to all products of the catalog to be created such as product identifier, price, description, and so on. But information about electronics products will not apply to furniture products and vice-versa. The category-specific information will be stored as **attributes to the specific category**.

To make clearer where and when to create a field and where and when to create an attribute in a catalog, a comparison between fields and attributes is given:

Field	Attribute
Is created in the MDM 5.5-Console	Is created in the MDM 5.5-Client in the Taxonomy mode
By the creation of a field, the catalog should be unloaded (creation of a field represents a changing of the data modeling)	By the creation of an attribute, the catalog should be loaded (creation of an attribute doesn't represent a changing in the data modeling). This makes possible to link an attribute to the corresponding category.
A field applies to all the product records. It is category independent.	An attribute is category-specific. It applies only to products which are linked to the specific category. So, an attribute cannot be linked to the taxonomy root but only to a category of the taxonomy.
A field is displayed as well in the Fields-grid as in the Field-Detail-grid.	An attribute is not displayed in the records-grid. It is displayed in the Record-Detail grid only for the selected record, if the record is linked to the category to which the attribute is linked.
A drill-down search and a free-form search can be done based on a field.	A drill-down search can be done based on an attribute. No free-form search is supported.

Tab. 1 Comparison Fields vs. Attributes

The typical steps of designing a catalog include the following:

- To list all the fields to be included in the catalog.
- Decide the type of each field of catalog. If a field is designed to be a lookup field (= a field which takes its legal values for data entry and search from a corresponding lookup table) then decide if the corresponding lookup table should be flat, hierarchical, taxonomy or qualified.

Note: each lookup field in the main table becomes a searchable dimension of the catalog and appears automatically in the MDM 5.5 Client as a search tab in the Search Parameters pane in Record mode.

- Define for each field of the catalog how the data should be stored in the catalog – as a numeric value like integers or real numbers, as currency, as measurement, as text or images, whether the field should be a single-valued or multi-valued and whether it should be a lookup field.
- Sort out what information from the product data base of the customer should be used to define the categorization of all products and to define the attributes associated to which categories.
The categories, the category specific attributes (and priority) and one or more classification hierarchies will be stored in the taxonomy. A rough taxonomy must be defined at the beginning of the design process to support the subsequent phases. The taxonomy can be refined continuously during the course of the implementation and also after deployment.
- Define the set of additional 'support' tables like flat, hierarchy and qualified lookup tables, object lookup tables and special tables.

3 The Step By Step Solution

To make this guide more understandable let's consider the example of a company who sells furniture and therefore it wants to create a catalog for persisting the data and for publishing. The information available in the data base of the customer, which represents the source information for the designer of a catalog, is contained in a file with the following structure:

Prod	Page	Descr	Manuf. Color, Nr	Manuf. Finish color, Nr	Model	Width	Depth/Height	Price	Delivery	Manufacturer
1001	Office<211	5	Grey; 123456		PC desk	90 cm	55/60.5 cm	169,00 E	Business, 24-hour service	Furniture<Modular<Hecks
1002	Office<974	11	Brown; 456765		PC stand	83 cm	33/88 cm	399,00 E	Business	Furniture<Solid<Schmitt
1003	My room<311	388	Brown; 87321A	Natural; 801251	Bureau	80.5 cm	33/99 cm	210,00 E	Business	Furniture<Modular<Brandt
1004	Kitchen<1024	11	White; 54124D		Bar	39 cm	43,5/85 cm	99,90 E	Business, 24-hour-service	Furniture<Solid<Ronda u
1005	Bedroom<51	33	Antique; 48127A	Natural; 891233	Double bed	180 cm	200/50 cm	505,30 E	Business	Furniture<Modular<Hecks
1006	My room<88	21	Brown; 76561D	Rustic, 712341	Bed	100 cm	180/50 cm	99,90 E	Business, 24-hour-service	Furniture<Modular<Brandt
1007	Kitchen<101	44	Green; 21212A		Table Comb i	225 cm	64/225 cm	499,00 E	Business, 24-hour-service	Furniture<Modular<Brandt
1008	Bedroom<111	312	White; 123123	Matte; 54314Y	Cabinet with doors	220 cm	180/90 cm	399,00 E	Business	Furniture<Solid<Schmitt

Note:

We will not consider in this script the aspect of publishing and printing of a catalog.

3.1 Analyze the source information (see data file above)

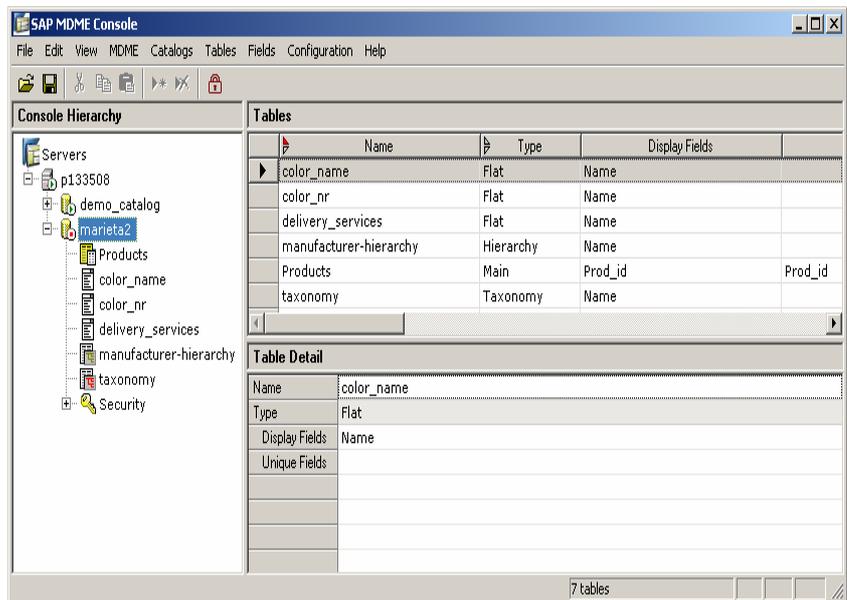
1. The column **Prod** contains unique values, so the value of this field will be used in the catalog to be created as the key field (unique field) of a product record.
2. The column **Page** contains two values separated during the delimiter '<'. The first part of the content of this column gives information about the category of the product (Office, My room, Kitchen and Bedroom). The content of the second part of this column looks like a page number in an old catalog. The consequence is to define in the catalog two fields corresponding to these two parts: a field Category and a field Old_catalog.
3. The column **Descr** looks to be somehow connected with the information contained in the second part of the column Page. The content of the column Descr could be the number of the paragraph on the page number from above. (to be clarified with the customer). If so, the consequence is to concatenate the already created field Old_catalog (see point 2) with the paragraph number as the second part of the field.
4. The column **Manuf. Color, Nr** contains two values separated by ';': a color and a color number. Because every product in the source file has a manufacturer color and color number the consequence is to define two correspondent fields in the catalog.
5. The column **Manuf. Finish color, Nr** is not filled for every product in the source file. If we analyze which products have a finished manufacturer color and number we detect that only the products of the categories 'My room' and 'Bedroom' are connected with this information. So the consequence is to think about attributes (see Tab. 1). We will create in the client the attributes 'finish color' and 'finish color nr' and we will link these two attributes to the categories 'My room' and 'Bedroom'.
6. By analyzing of the content of the column **Model** we can see that it could be considered as a refinement of the category of the product (see point 2 part one). The consequence is to design the model as the second level of the of the category hierarchy. So, we will concatenate the category (part one of the column Page) with the Model to become the category field as a hierarchical relationship between the two fields.
7. The column **Width** contains the width of the product. It is available for all records in the source file and therefore a correspondent field will be created in the catalog.
8. The column **Depth/Height** contains two values separated by '/'. We will create two fields in the catalog, one corresponding to the depth and another one corresponding to the height. Notice that one record uses a ',' as a decimal point (43,5 instead of 43.5). MDM 5.5 does not support this. Think about a correction of this value by importing of data.
9. The column **Price** contains the price of the product in Euro. It is available for all records in the source file and therefore a correspondent field will be created in the catalog. Take notice about the problem of the decimal point like by point 8.
10. The column **Delivery** contains one or two values, but they are similar from the point of view of the semantic. So, a field with multiple values will be created in the catalog for this source field.

11. The column **Manufacturer** contains three values separated during the delimiter '<'. The three values belong together and they describe the manufacturer of the product. So, a field of type Lookup [Hierarchy] will be created in the catalog corresponding to this source field.

3.2 Create the catalog

The evaluation of the data base of the customer and his requirements has been completed. The data model was designed and agreed (see paragraph 3.1).

- Start the MDM 5.5-Console.
- Create a new empty catalog. Note that the main table **Products** of the catalog was automatically created.
- Create the taxonomy table **taxonomy** using the functionality of adding a new table of type Taxonomy to the new created catalog.
- Create in the same way the following tables:
 - color_name** - as lookup table for the manufacturer colors
 - color_nr** - as lookup table for the manufacturer color number
 - delivery_services** – as lookup table multi-valued for the delivery
 - manufacturer-hierarchy** – of type Hierarchy for the hierarchy of the manufacturer
- Create the fields for the created tables above



- Create the fields of the main table **Products**:

Prod_Id - type text, unique field = yes

Old_catalog – type text

Color - type lookup [Flat] based on the Lookup table color_name

Color_nr – type lookup [Flat] based on the Lookup table color_nr

Width – type Measurement, Dimension = Length, default Unit = cm, Decimal Places = 1

Depth - type Measurement, Dimension = Length, default Unit = cm, Decimal Places = 1

Height - type Measurement, Dimension = Length, default Unit = cm, Decimal Places = 1

List_price – type Currency, symbol = €, Decimal Places = 2

Delivery_services – type lookup [Flat] based on the Lookup table delivery_services, multi-Valued = yes

Manufacturer_hierarchy – type lookup [Hierarchy] based on the Lookup table manufacturer-hierarchy

Category_hierarchy – type lookup [Taxonomy] based on the Lookup table taxonomy

Color_nr – type lookup [Flat] based on the Lookup table color_nr

The screenshot shows the SAP MDM Console interface. On the left, the 'Console Hierarchy' tree is expanded to show the 'Products' table under the 'marieta2' catalog. The main area displays a table of fields for 'Products' with columns for Position, Name, Type, Keyword, Display Field (DF), and Unique Field (UF).

Pos.	Name	Type	Keyword	DF	UF
[1]	Prod_id	Text [50]	Yes	[1]	[1]
[2]	old_catalog	Text [50]	No		
[3]	color	Lookup [Flat]	No		
[4]	width	Measurement [Length]			
[5]	depth	Measurement [Length]			
[6]	height	Measurement [Length]			
[7]	list_price	Currency			
[8]	delivery_service	Lookup [Flat] (multi-valued)	No		
[9]	manufacturer_hierarchy	Lookup [Hierarchy] (multi-valued)	No		
[10]	category_hierarchy	Lookup [Taxonomy]	No		
[11]	color_nr	Lookup [Flat]	No		

Below the table, the 'Field Detail' for 'Prod_id' is shown:

Name	Prod_id
Type	Text
Keyword	Yes
Display Field	Yes
Unique Field	Yes
Width	50

The status bar at the bottom indicates '11 fields'.

Note:

To get more details how to create tables and fields for a repository in the MDME Console, please consult the Manuals using the following web page:

<http://Service.sap.com/nw04installation>, under *Installation & Upgrade Guides - Sap NetWeaver - Release 04 - Operations*.

3.3 General Remarks

Creation of Taxonomy

Product categories and subcategories are represented in an MDM 5.5 catalog as a hierarchy in the taxonomy table. The taxonomy table is a special kind of lookup table that provide support for the hierarchy of categories and also for the category-specific attributes that can be assigned to each category.

Every product should belong to a category. A product can belong to at most one leaf-node category. The taxonomy table itself and its fields are defined and created in the MDM 5.5 Console.

The hierarchy of categories and their associated category-specific attributes can be created (populated) and managed either during the importing of the data using the MDM 5.5 Import Manager (= on the fly) or they can be created manually in the MDM 5.5 Content Manager (Client) in the Taxonomy mode, after the catalog was created and loaded.

- If the information concerning the categories in the source file is well structured and consolidated, (they should not be modified after importing) it is recommended to create the categories on the fly.
- If the source file doesn't deliver enough good information to be able to define the taxonomy, it is recommended to create the categories manually in the client.

Taxonomy analysis includes the following:

- Source data analysis to identify the key areas to which the products belong (for example electronics, furniture, computer, office...).
- Decide which data of the source file are fields, category-specific attributes or hierarchies.
- Prioritization of attributes on category basis.

Flat and Hierarchy Lookup Tables

A lookup table is used to store values that are shared by many records in other tables with the intension to assure a valid set of legal values for the corresponding lookup field. This set of legal values will be used for data entry and for searching activities (in the MDM 5.5 Client) connected to the lookup field. The storing of the shared values in a lookup table brings two advantages:

- A change of a value in the lookup table causes an immediately update of all records that reference it (user friendly).

- The maintenance of a lookup field is constraint to the set of the legal values from the corresponding lookup table (data consistency).

Note: A lookup table from type Hierarchy is useful if there is a hierarchical relationship between the values of the correspondent lookup field. A hierarchical lookup table is useful also when it has only leaf nodes below the root, because it allows you to change the default alphabetical sequence of values (what is not allowed in a flat table).

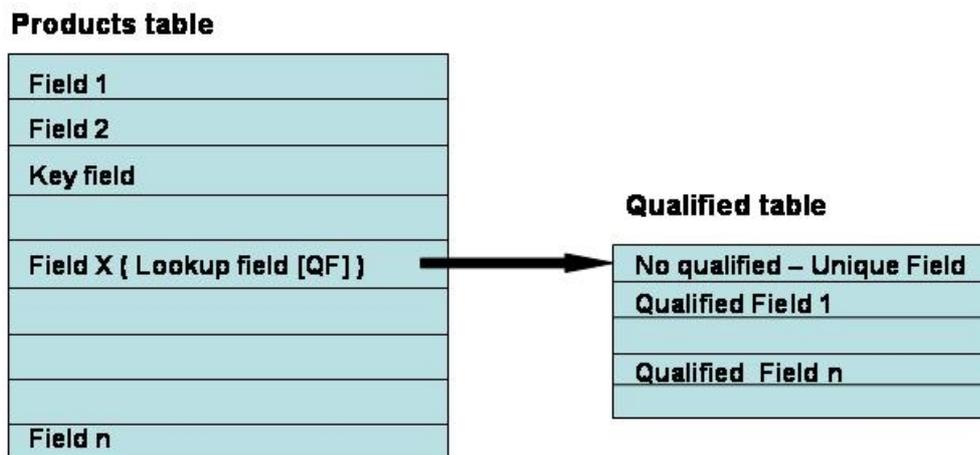
Qualified Lookup Tables

A qualified table is a special kind of table that is extremely flexible. It is normally used to store in an efficiently way complex relationships between a product record of the main table and one or more lookup table entries that contain various additional information concerning this product.

Using qualified tables and qualifiers allow you to store a large amount of potentially sparse data. Doing so, n fields from the main table can be replaced with a single qualified lookup field in the main table. This qualified lookup field takes its values from a qualified table linked to it, table which has n records, corresponding to the n replaced fields of the main table and one or more qualifiers.

See the next picture about the relation between the main table and the qualified table:

Definition of the Products and qualified tables



As example let's consider the following source file that contains sparse price data based on different quantities as normal fields:

Prod_ID	Name	Qt.1 - 10	Qt.10 - 25	Qt. 26 -50	Qt. 51 - 100	Qt. > 100
1001	White chalk	2.52 \$			28 \$	40 \$
1002	Paper		1.80 \$			85.90 \$

Because the pricing data are not available for each type of quantity, we can use a qualified lookup field in the main table instead of the quantity based price fields to store the quantity based pricing data. The qualified table associated to this qualified lookup field will have the only single field Quantity as non qualifier and the Price as qualifier. In our example the qualified table Pricing will contain the following quantity records:

Pricing:

Quantity	Price
1 - 10	
10 - 25	
26 - 50	
51 - 100	
> 100	

A link between the qualified lookup field of each record of the main table and the qualified table records will be created.

Based on this design, the structure of the catalog concerning the quantity based pricing likes as shown below:

Prod_ID	Name	Lookup[Pricing]
1001	White chalk	1 - 10; 2.52 \$ 51 - 100; 28 \$ > 100; 40 \$
1002	Paper	10 - 25; 1.80 \$ > 100; 45.80 \$

Note: A link between the main table and the qualified table is created only for those product/quantity combinations for which price values actually exist.

Using the concept of qualified lookup fields and qualified lookup tables, the source data can be restructured so that an efficient storage of data is assured and also an efficient way of searching in the catalog.

For loading, the data to the qualified table it necessary to consider:

The source information should have the required values for the qualified table and the corresponding values for the key field in the main table. This means that there should be a link value between the source data in the qualified table and the main table.

For a better understanding, see next picture:

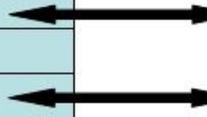
Relation between the source data to be loaded in the qualified table and the Products table

Products table

Field 1
Field 2
Key field
Lookup field (QF)
Field n

Source data required by the qualified table

Key field – (Main Table)
No qualified – Unique Field
Qualified Field 1
Qualified Field n



3.4 MDM 5.5 Modeling general tips

- For prices use qualified table → always available for search / publishing
- Use text blocks for describing text – not for engineering data
- Lookup(hierarchy) can be multi-value, lookup(taxonomy) only single value
- One version of the truth – avoid redundancy wherever it is possible.
- Attribute values are lookups that are a closed list – list should be entered before loading product data.
- The unique key [UF] can be spread over more than one field.
- Keyword setting in the console stands for marking the field for keyword search, e.g. “25 ft Meas. Tape” → MDM 5.5 searches for each part of the string separated by blank – in all fields that are flagged with the Keyword word.
MDM 5.5 stores the fields and their words for keyword search in a special cache. There is XEROX soundex used for stemming (i.e. to cover minor grammatical differences). In standard MDM 5.5 only structured content is searchable (lookup fields and attributes), by using Keyword setting also unstructured content is searchable.
You find keyword search in the free-form search tab.
e.g. for Region that is used as No Qualifier the keyword setting shouldn't be used, because search is already available in product lookup
- For predefined texts use text blocks – order by data groups, for different semantics use a different field (also a different group in data groups) e.g. one lookup for Additional Sales Texts and one lookup for Additional Technical Restriction.
Separate text blocks into separate fields if you want to apply different rules to the fields.
Use text blocks if you only have text, lookups if there are additional images.
- Take care of naming when defining relationship definition. Use names that show the direction of the relation
- Attribute description is not available in layout mode, so don't use attribute description to detail attribute values.
- Take in account that in layout you can only display direct relations, not more than one level away
- To get rid of field NAME in a lookup table – necessary if main field is not text - you first have to define the new field, flag it as DF (display field) and then remove the DF flag from NAME and remove NAME
- Don't put products on sub tables, In sub tables there are no qualified tables, no taxonomies and no masks available
- Model prices in qualified tables as qualifiers → advantages in search and layout.

- Wherever you use lookups put all lookup specific functionality in the lookup table.
- Field options: Cache yes/no – use it if you expect to have less than 100 distinct values – Cache means is the value range cached or not
- Qualified Tables: Use Qualified tables always if you have different values under different conditions. Qualified tables can be used to group different features together. There is no taxonomy search for data in the qualified tables.
- If values repeat quite often, use lookup tables.
- The more you bring data into pieces, the better it is to articulate their behavior for delivery.
- Use Sibling Relationship for example for cross sales issues.
- Relationships with sub tables are available, but access in layout mode is very limited and the content of the sub tables itself is rather restricted (no qualified tables, no taxonomy)
- If you sell an item (even as a spare part) put it into the main table.
For example: If you have a motor built-in in your product, but not sold separately, put it in a lookup sub table.
- If the outcome of the modeling is that the main table has >50 fields, try to restrict access with user specific user interface (web).

3.5 MDM 5.5 Taxonomy tips

- Use alias for attributes when is possible, e.g. to provide different definitions for different departments.
- By definition, all category specific attributes are handled as lookups.
- Often the engineering way of thinking will dictate the taxonomy.
- Recommendation: Having 1 Taxonomy, but n hierarchies – e.g. for selection
- Setting up the taxonomy hierarchy is very important:
 1. Make it consistent – Taxonomy is not good, if you can put one product into different categories.
 2. Avoid having too many levels
 3. Organize it in a more engineering driven way.
- Categories with attributes attached are displayed in bold.
- Find attributes that way: an attribute is always a question to a product.
- Classify attributes for importance – use customer priority – enable in configuration. Priorities are used for display the attributes in record detail but can also be used for limiting attributes shown in the web.
- By default, coupled numeric attributes are multi-value.
- Advantage of numeric attributes: sort by values (instead of lexicographic search) and usage of dimensions.
- You can group attributes with using the Alias parameter → API.
- Some taxonomy design rules:
 1. Guarantee consistency – one record should fit only to one category.
 2. A category defines the purpose of an item – The category answers questions about the products.
 3. Keep it as simple as possible and avoid too many levels.
 4. For naming of leaf nodes use Nouns.
- Use as much as you can numeric attributes, measurements and ranges.
- Put the common attributes at the top of the table and the different ones below.
- If you search for a specific measurement/dimension for your new attribute, just use the Menu Attributes→Unit and Dimension Finder.
- Matching Sets can relate to attributes that are inherited.
- Consolidate children - Only possible if they have the same set of attributes.

- Split categories only around single value text attributes, if you have numeric attributes first convert the values to text attributes.

www.sap.com/netweaver

THE BEST-RUN BUSINESSES RUN SAP

