

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



# **How to Configure Predefined Properties with Dependent Values**

**Version 1.00 – December 2004**

**PUBLIC**

**Applicable Releases:  
SAP NetWeaver '04 SPS 09**

© 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<sup>®</sup>, 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 Scenario

*SAP NetWeaver's Knowledge Management* offers you the possibility of using properties with interdependent values. When you determine the value of a property, the values of other properties can vary according to the value you choose.

This How-to guide shows you how to proceed if you want to integrate a new tab listing the properties *Country* and *City* in the *Details* dialog box of a document. These properties are to be used to describe the origin of documents created in a company with several European outposts. The values for the property *City* are to vary depending on the value that a user selects for the property *Country*. In our example, the countries used are Germany, France, and Spain. There are several cities available for each country.

## Values for the properties *Country* and *City*

Country	Germany	France	Spain
City	Walldorf Berlin Frankfurt Hamburg	Paris Nice Bordeaux	Madrid Barcelona Sevilla

Advantages for your users:

- By prescribing correct selection options you support your users in the selection of properties. Spelling errors and invalid combinations of values are avoided.
- You can offer the properties and interdependent values in the languages used by your users in the portal. This is realized by using bundle files.

Advantages for administrators:

- The dependencies between the values of different properties are evaluated by a special property renderer that is included in the standard delivery. You do not have to provide your own Java classes.
- The properties and their dependencies are maintained in the portal in the *Configuration* iView as normal.

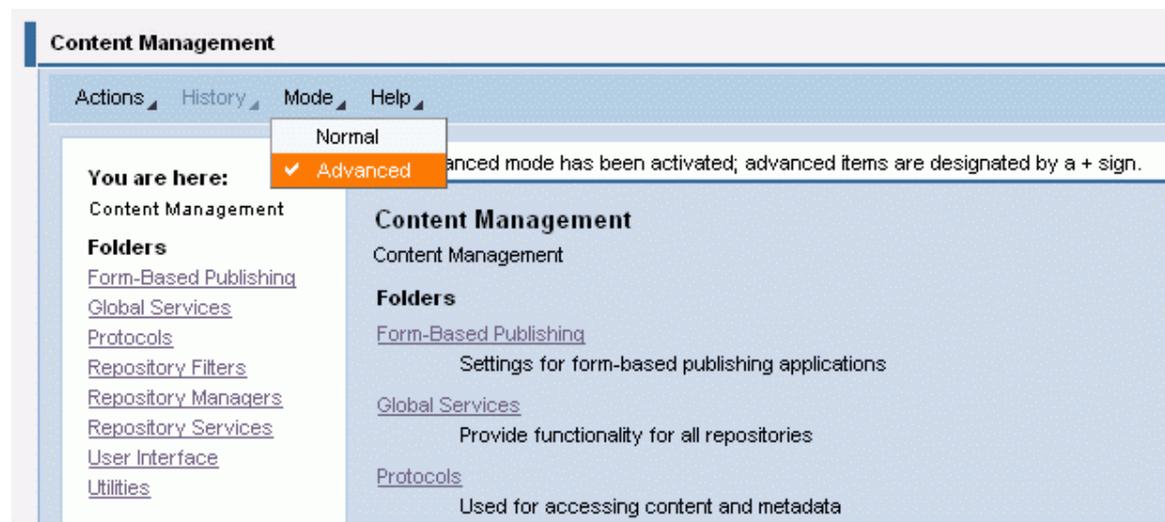
## 2 Prerequisites

### 2.1 Assigned Roles

You need the role of a super administrator or system administrator in order to carry out the tasks described in this guide.

### 2.2 Calling Up the Configuration iView and Activating Advanced Mode

Call up the *Configuration* iView as follows: Choose *System Administration* → *System Configuration* in the portal. Then choose *Knowledge Management* → *Content Management* in the detailed navigation pane. This screen is the point of access for the configuration steps below. Make sure that the advanced view is activated in the *Configuration* iView. To do so, proceed as follows: To activate the advanced view, choose *Mode* → *Advanced* in the menu bar. The advanced view displays all configuration classes and parameters. This means that you do not have to search for the individual parameters that you have to call up for the steps described in this guide.



## 3 The Step By Step Solution

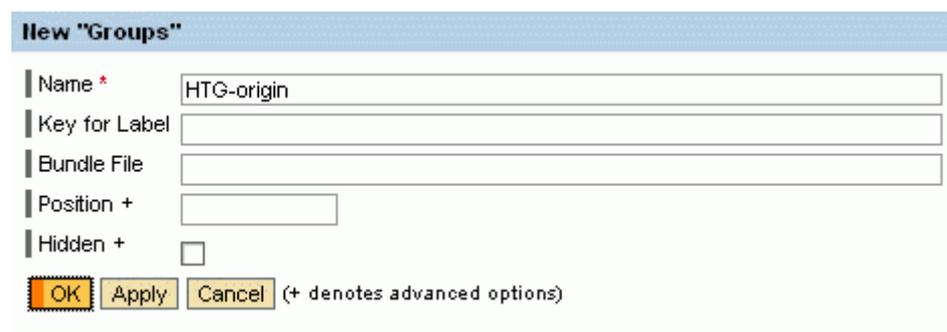
### 3.1 Overview

While working through this section, you will create several properties, configure interdependent property values, and modify the property structure of the *Details* dialog box. In connection with this, we shall show you how the configuration affects the *Details* dialog box of a document. You then find a selection of optional configuration steps that you can carry out to enhance your scenario.

### 3.2 Creating a Group

The properties used in this How-to guide (*Country* and *City*) need to be placed in a separate group because of their usage (describing the origin of documents). Proceed as follows:

1. To create a group, choose *Global Services* → *Property Metadata* → *Groups*.
2. Choose *New*.
3. Specify the parameters. In our example you only have to specify a name for the group.
4. To save your entries, choose *OK*.



**New "Groups"**

Name \*

Key for Label

Bundle File

Position +

Hidden +

(+ denotes advanced options)

### 3.3 Creating the Base Property *Country*

You now create the property *Country*. You want the value of the property *City*, which you are going to create later on, to vary depending on the value of this property. We can therefore refer to the property *Country* as the *base property*.

1. Choose *Global Services* → *Property Metadata* → *Properties* in the *Configuration* iView.
2. Choose *New*.
3. Specify the following parameters:

```
Unique ID      = HTG-Country
Property ID    = Country
Type           = String
Group          = HTG-origin
Maintainable   = activated
Dependencies   = activated
Allowed Values = Germany, France, Spain
```

All other parameters are not necessary for this example.

4. To save your entries, choose *OK*.

**View "HTG-Country"**

Description	<input type="text"/>
Property ID *	Country
Namespace Alias *	default
Type *	String
Group *	HTG-origin
Mandatory	<input type="checkbox"/>
Multi-Valued	<input type="checkbox"/>
Read Only	<input type="checkbox"/>
Maintainable	<input checked="" type="checkbox"/>
Indexable	<input type="checkbox"/>
Default Value	<input type="text"/>
Allowed Values (csv)	Germany,France,Spain
Key for Label	<input type="text"/>
Meta Data Extension	Not set
Folder Validity Patterns (csv)	/
Document Validity Patterns (csv)	/
Resource Types (csv)	<input type="text"/>
Mime Types (csv) +	<input type="text"/>
Default Sorting * +	Ascending
Label Icon +	<input type="text"/>
Hidden +	<input type="checkbox"/>
Dependencies +	<input checked="" type="checkbox"/>
Additional Metadata (csv) +	<input type="text"/>
Property Renderer +	Not set
Virtual +	<input type="checkbox"/>
Composed of +	<div style="border: 1px solid #ccc; padding: 2px;"> <div style="background-color: #f0f0f0; padding: 2px;">Unique ID</div> <div style="padding: 2px;">Not set</div> <div style="display: flex; justify-content: space-between; align-items: center; padding: 2px;"> <span>⏪ ⏩ ⏴ ⏵</span> <span>Page 1 / 1</span> </div> </div>
Comparator Class +	<input type="text"/>

(+ denotes advanced options)

### 3.4 Creating the Holding Property *City*

You now create the property *City*. You do not define values within this property. It serves as a holder for sub-properties whose possible values are loaded dynamically according to the value of the property *Country*.

Instead of properties, the dependencies between the properties *Country* and *City* are stored in the parameter `Additional Metadata`. The property renderer *multi\_dependence*, which is included in the standard delivery of KM, is used to evaluate these dependencies.

The following syntax is used in the parameter `Additional Metadata`:

```
dependOn=<property ID of the base property>,  
<value 1 of the base property>=<property ID of the assigned sub-property>,  
<value 2 of the base property>=<property ID of the assigned sub-property>,  
<value n of the base property>=<property ID of the assigned sub-property>
```

If you have your own namespace and are not using the namespace *default*, you must also specify the namespace alias, as shown below:

```
dependOn=<namespace alias:property ID of the base property>,  
<value 1 of the base property>=<namespace alias:property ID of the assigned sub-property>,  
<value 2 of the base property>=<namespace alias:property ID of the assigned sub-property>,  
<value n of the base property>=<namespace alias:property ID of the assigned sub-property>
```

Enter the property ID of the assigned sub-property for each value that the property *Country* can adopt. The configuration of the sub-properties is explained in the next step. You should therefore make a note of the property IDs that you specify in this step.

1. You are in the Configuration iView under *Global Services* → *Property Metadata* → *Properties*.
2. Choose *New*.
3. Specify the following parameters:

```
Unique ID          = HTG-City  
Property ID       = City  
Type              = String  
Group             = HTG-origin  
Maintainable     = activated  
Additional Metadata = dependOn=Country,  
                   Germany=GermanCity, France=FrenchCity,  
                   Spain=SpanishCity  
Property Renderer = multi_dependence
```

All other parameters are not necessary for this example.

4. To save your entries, choose *OK*.

View "HTG-City"	
Description	
Property ID *	City
Namespace Alias *	default
Type *	String
Group *	HTG-origin
Mandatory	<input type="checkbox"/>
Multi-Valued	<input type="checkbox"/>
Read Only	<input type="checkbox"/>
Maintainable	<input checked="" type="checkbox"/>
Indexable	<input type="checkbox"/>
Default Value	
Allowed Values (csv)	
Key for Label	
Meta Data Extension	Not set
Folder Validity Patterns (csv)	/
Document Validity Patterns (csv)	/
Resource Types (csv)	
Mime Types (csv) +	
Default Sorting * +	Ascending
Label Icon +	
Hidden +	<input type="checkbox"/>
Dependencies +	<input type="checkbox"/>
Additional Metadata (csv) +	dependOn=Country,Germany=GermanCity,France=FrenchCity,Spain=Spanish
Property Renderer +	multi dependence
Virtual +	<input type="checkbox"/>
Composed of +	<div style="border: 1px solid gray; padding: 5px;"> <div style="background-color: #cccccc; padding: 2px;">Unique ID</div> <div style="padding: 2px;">Not set</div> <div style="display: flex; justify-content: space-between; align-items: center; padding: 2px;"> <span>⏪</span> <span>⏩</span> <span>⏴</span> <span>⏵</span> <span>Page 1 / 1</span> </div> </div>
Comparator Class +	
<div style="display: flex; gap: 10px;"> <span>Edit</span> <span>Close</span> <span>(+ denotes advanced options)</span> </div>	

### 3.5 Creating Sub-Properties

You now create a sub-property for each value of the base property *Country*. Each of these sub-properties contains several towns in the relevant country. The values of these sub-properties are later loaded to the value set of the holding property *City* in accordance to the value of the property *Country*.

Firstly, create the property for the German towns. The values of this property are to be loaded if the value *Germany* is selected for the base property *Country*. The most important parameters should be filled as follows:

1. You are in the Configuration iView under *Global Services* → *Property Metadata* → *Properties*.
2. Choose *New*.
3. Specify the following parameters:

Unique ID = HTG-GermanCity  
 Property ID = **GermanCity**  
 Type = String  
 Group = HTG-origin  
 Maintainable = **deactivated**  
 Allowed Values = Walldorf, Berlin, Frankfurt, Hamburg

All other parameters are not necessary for our example.

4. To save your entries, choose *OK*.

**View "HTG-GermanCity"**

Description					
Property ID *	GermanCity				
Namespace Alias *	default				
Type *	String				
Group *	HTG-origin				
Mandatory	<input type="checkbox"/>				
Multi-Valued	<input type="checkbox"/>				
Read Only	<input type="checkbox"/>				
Maintainable	<input type="checkbox"/>				
Indexable	<input type="checkbox"/>				
Default Value					
Allowed Values (csv)	Walldorf,Berlin,Frankfurt,Hamburg				
Key for Label					
Meta Data Extension	Not set				
Folder Validity Patterns (csv)	/				
Document Validity Patterns (csv)	/				
Resource Types (csv)					
Mime Types (csv) +					
Default Sorting * +	Ascending				
Label Icon +					
Hidden +	<input type="checkbox"/>				
Dependencies +	<input type="checkbox"/>				
Additional Metadata (csv) +					
Property Renderer +	Not set				
Virtual +	<input type="checkbox"/>				
Composed of +	<table border="1"> <tr> <td>Unique ID</td> </tr> <tr> <td>Not set</td> </tr> <tr> <td>  </td> </tr> <tr> <td>Page 1 / 1</td> </tr> </table>	Unique ID	Not set		Page 1 / 1
Unique ID					
Not set					
					
Page 1 / 1					
Comparator Class +					

**Edit** **Close** (+ denotes advanced options)

Now create the property for the French towns (for when Country = France).  
Specify the parameters as follows:

Unique ID = HTG-FrenchCity  
 Property ID = *FrenchCity*  
 Type = String  
 Group = HTG-origin  
 Maintainable = **deactivated**  
 Allowed Values = Paris,Nice,Bordeaux

**View "HTG-FrenchCity"**

Description					
Property ID *	FrenchCity				
Namespace Alias *	default				
Type *	String				
Group *	HTG-origin				
Mandatory	<input type="checkbox"/>				
Multi-Valued	<input type="checkbox"/>				
Read Only	<input type="checkbox"/>				
Maintainable	<input type="checkbox"/>				
Indexable	<input type="checkbox"/>				
Default Value					
Allowed Values (csv)	Paris,Nice,Bordeaux				
Key for Label					
Meta Data Extension	Not set				
Folder Validity Patterns (csv)	/				
Document Validity Patterns (csv)	/				
Resource Types (csv)					
Mime Types (csv) +					
Default Sorting * +	Ascending				
Label Icon +					
Hidden +	<input type="checkbox"/>				
Dependencies +	<input type="checkbox"/>				
Additional Metadata (csv) +					
Property Renderer +	Not set				
Virtual +	<input type="checkbox"/>				
Composed of +	<table border="1"> <tr> <td>Unique ID</td> </tr> <tr> <td>Not set</td> </tr> <tr> <td> <input type="button" value="&lt;"/> <input type="button" value="&gt;"/> <input type="button" value="&lt;="/> <input type="button" value="&gt;="/> </td> </tr> <tr> <td>Page 1 / 1</td> </tr> </table>	Unique ID	Not set	<input type="button" value="&lt;"/> <input type="button" value="&gt;"/> <input type="button" value="&lt;="/> <input type="button" value="&gt;="/>	Page 1 / 1
Unique ID					
Not set					
<input type="button" value="&lt;"/> <input type="button" value="&gt;"/> <input type="button" value="&lt;="/> <input type="button" value="&gt;="/>					
Page 1 / 1					
Comparator Class +					

(+ denotes advanced options)

Now create the property for the Spanish towns (for when `Country = Spain`).  
Specify the parameters as follows:

```

Unique ID       = HTG-SpanishCity
Property ID    = SpanishCity
Type           = String
Group          = HTG-origin
Maintainable   = deactivated
Allowed Values = Madrid,Barcelona,Sevilla

```

**View "HTG-SpanishCity"**

Description	
Property ID *	SpanishCity
Namespace Alias *	default
Type *	String
Group *	HTG-origin
Mandatory	<input type="checkbox"/>
Multi-Valued	<input type="checkbox"/>
Read Only	<input type="checkbox"/>
Maintainable	<input type="checkbox"/>
Indexable	<input type="checkbox"/>
Default Value	
Allowed Values (csv)	Madrid,Barcelona,Sevilla
Key for Label	
Meta Data Extension	Not set
Folder Validity Patterns (csv)	/
Document Validity Patterns (csv)	/
Resource Types (csv)	
Mime Types (csv) +	
Default Sorting * +	Ascending
Label Icon +	
Hidden +	<input type="checkbox"/>
Dependencies +	<input type="checkbox"/>
Additional Metadata (csv) +	
Property Renderer +	Not set
Virtual +	<input type="checkbox"/>
Composed of +	<div style="border: 1px solid gray; padding: 5px;"> <div style="background-color: #cccccc; padding: 2px;">Unique ID</div> <div style="padding: 2px;">Not set</div> <div style="display: flex; justify-content: space-between; align-items: center; padding: 2px;"> <span>⏪</span> <span>⏩</span> <span>⏴</span> <span>⏵</span> <span>Page 1 / 1</span> </div> </div>
Comparator Class +	

Edit Close (+ denotes advanced options)

You have now created all necessary properties.

### 3.6 Creating Property Groups for a Property Structure

You have to create a new property group so that the properties *Country* and *City* are displayed on a separate tab in the *Details* dialog box. You enter both properties as elements there. To do this, proceed as follows:

1. Choose *Global Services* → *Property Structures* → *Property Groups*.
2. Choose *New*.
3. Specify the following parameters:

```
Name           = Origin
Group Items    = Country, City
```



Use the property ID or combined specification namespace alias:property ID in the parameter *Group Items*.

4. To save your entries, choose *OK*.

Description	
Group Items	Country, City
Key for Label	
Bundle File	
Property Metadata Group +	Not set
Group Renderer * +	standard
Single Focus +	<input type="checkbox"/>

Edit Close (+ denotes advanced options)

### 3.7 Updating the Property Group *all\_groups*

You have to update the elements of the property group *all\_groups* so that the property group that you just created is displayed in the *Details* dialog box. The property group *all\_groups* is used by the property structure *DefaultPropertiesStructure*, which is used by default for the display of the properties in the *Details* dialog box.

1. You are in the Configuration iView under *Global Services* → *Property Structures* → *Property Groups*.
2. Select the property group *all\_groups* from the table and choose *Edit*.
3. Enter the property group *Origin*, which you just created, into the parameter *Group Items* in addition to the existing entries.
4. To save your entries, choose *OK*.

**Edit "all\_groups"**

Object remains locked until you click OK or Cancel

Description

Group Items

Key for Label

Bundle File

Property Metadata Group +

Group Renderer \* +

Single Focus +

(+ denotes advanced options)

You have now carried out all necessary configuration steps.

### 3.8 Displaying the Result

Use the Explorer iView to navigate to a document. The Explorer iView is located at *Content Administration* → *KM Content* in the portal. Call up the *Details* dialog box for the document. Choose *Settings* → *Properties* in the *Details* dialog box.

**How to use KM properties.pdf**

View Actions Collaboration Settings

Size: 51,5 KB  
Created: 17/12/04 4:45 PM  
Administrator, Super

Modified: 17/12/04 4:45 PM  
Administrator, Super

Read: Not Read

Close

Feedback  
Permissions  
Subscription

Reviews  
Reviews allow users to collaboratively determine the quality of a document. There are no reviews for this item.  
[Write First Review...](#)

The new tab *Origin* is located to the right with the properties *Country* and *City*.

**How to use KM properties.pdf**

View Actions Collaboration Settings

Size: 51,5 KB  
Created: 17/12/04 4:45 PM  
Administrator, Super

Modified: 17/12/04 4:45 PM  
Administrator, Super

Read: Not Read

Close

**Properties**

Name \*

Description

System Custom Rendering Miscellaneous Access Links **Origin**

Country

City

Save

Open the dropdown box for the property *Country*. Choose the value *Germany*.

The screenshot shows a web interface for editing a document's properties. On the left, there is a metadata section for 'How to use KM properties.pdf' with fields for Size (51,5 KB), Created (17/12/04 4:45 PM), Modified (17/12/04 4:45 PM), and Read status (Not Read). The main 'Properties' section has tabs for System, Custom, Rendering, Miscellaneous, Access Links, and Origin. The 'Country' dropdown is open, displaying a list of countries: Germany, France, and Spain. The 'City' field is currently empty.

The window is updated automatically. The dropdown box for the property *City* now contains the towns that are valid for the value *Germany*.

This screenshot shows the same 'Properties' window after the 'Country' has been set to 'Germany'. The 'City' dropdown is now open, displaying a list of German cities: Waldorf, Berlin, Frankfurt, and Hamburg. The 'Country' dropdown is now closed and shows 'Germany'.

Choose *Save* to save your chosen values.

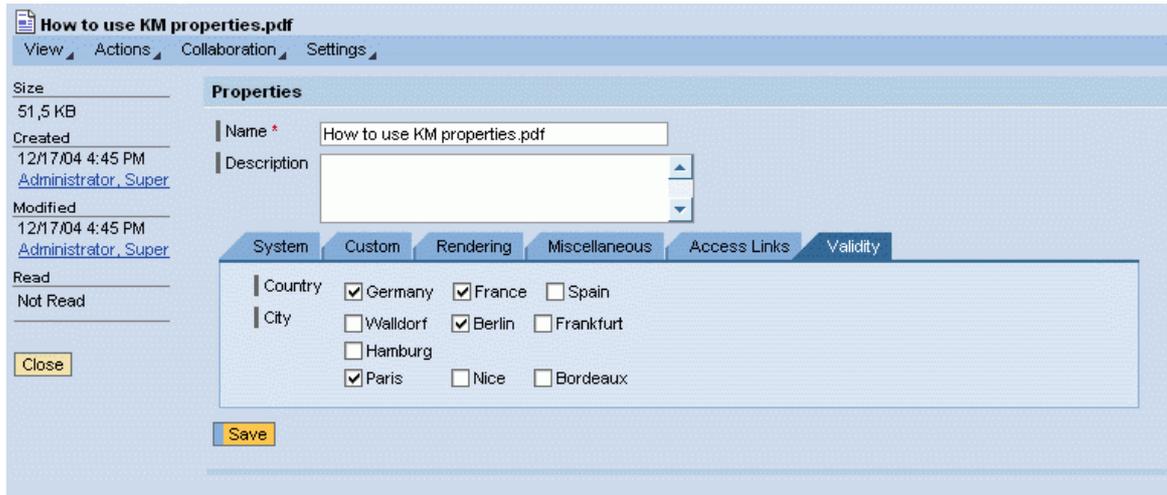
Every time that you change the base property, the window is updated and the towns relevant for the country chosen are uploaded.

This screenshot shows the 'Properties' window after the 'Country' has been changed to 'Spain'. The 'City' dropdown is open, displaying a list of Spanish cities: Madrid, Barcelona, and Sevilla. The 'Country' dropdown is now closed and shows 'Spain'.

## Alternative Usage

In order to permit multiple values, you have to activate the parameter *Multi-Valued* in the configuration of the properties *Country* and *City*. All values of the properties are then depicted as checkboxes in the *Details* dialog box.

You can use these to restrict the validity of a document to particular countries or towns, for example. You should remember to change the tab header according to the usage (for example, to *Validity*). The tab header corresponds to the name of the property group that you created under *Global Services* → *Property Structures* → *Property Groups*. If you use a portal in several languages, you can also use a bundle file to store the names of tab headers in different languages (see also [3.9.2](#)).



## 3.9 Optional Configuration Steps

### 3.9.1 Extension of Example on Several Levels

The use of predefined properties with interdependent properties is not restricted to two levels. You can configure three or more levels if you wish. Let us assume that you want to use the property *Department* as a third level. Proceed as follows to achieve this:

1. For additional property levels, the parameter `Dependencies` must be activated for the property on the previous level (here, the property *City*).
2. Create the additional holding property *Department*.  
You have to specify a dependency in the parameter `Additional` for every value that the property *City* can adopt.

The parameters of the property should be filled as follows:

```
Unique ID      = HTG-Department
Property ID    = Department
Type           = String
Group          = HTG-origin
Maintainable   = activated
Additional Metadata = dependOn=City,
                        Walldorf=Walldorf_Departments,
                        Berlin=Berlin_Departments,
                        Frankfurt=Frankfurt_Departments,
                        Hamburg=Hamburg_Departments,
                        Paris=Paris_Departments,
                        Nice=Nice_Departments,
                        Bordeaux=Bordeaux_Departments,
                        Madrid=Madrid_Departments,
                        Barcelona=Barcelona_Departments,
                        Sevilla=Sevilla_Departments
Property Renderer = multi_dependence
```

3. Firstly, create a corresponding property for each value that the property *City* can adopt. The values of this corresponding property are to be loaded dynamically to the range of values for the property *Department*.

Example of parameters for the property *Walldorf\_Departments*:

```
Unique ID      = HTG-Walldorf_Departments
Property ID    = Walldorf_Departments
Type           = String
Group          = HTG-origin
Maintainable   = deactivated
Allowed Values = PM, QM, Dev, Support
```

4. Create properties with the following property IDs according to the same principle:

```
Berlin_Departments
Frankfurt_Departments
Hamburg_Departments
Paris_Departments
Nice_Departments
Bordeaux_Departments
Madrid_Departments
Barcelona_Departments
Sevilla_Departments
```

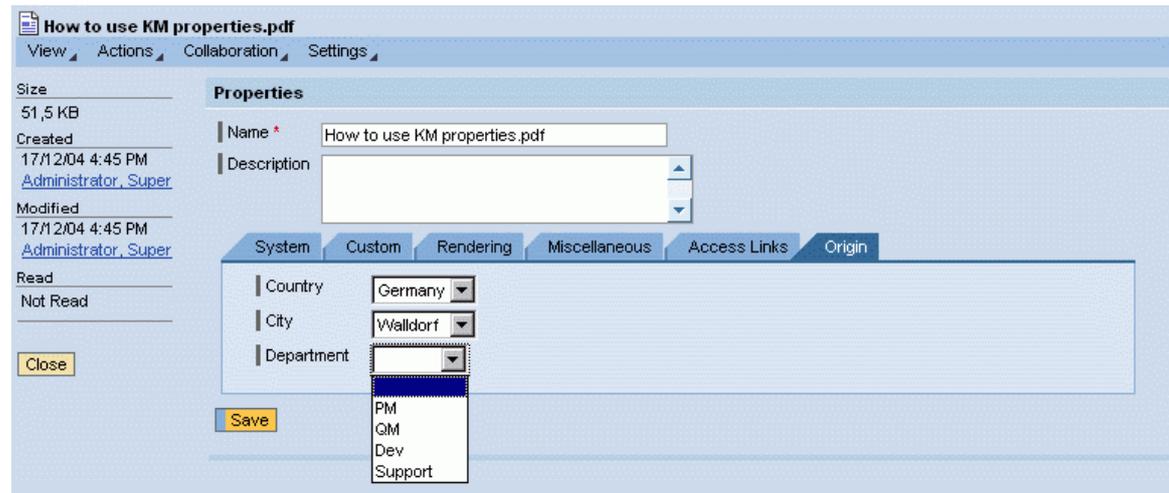
5. Proceed as follows in order to include the property *Department* so that it is displayed alongside *Country* and *City* on the tab *Origin*:

- a. Choose *Global Services* → *Property Structures* → *Property Groups*.
- b. Select the entry *Origin*, and then choose *Edit*.
- c. Modify the following parameters:

```
Group Items = Country, City, Department
```

- d. To save your entries, choose *OK*.

This completes the configuration. You can check the result in the *Details* dialog box.



If you want to use further levels, repeat steps 1 to 5 accordingly.

### 3.9.2 Using Bundle Files

You can use bundle files to offer labels and values in various languages. To do this, create a bundle file, specify it in a metadata extension, and assign the metadata extension to the properties.

At the end of this How-to guide is a link to a detailed procedure for changing the labels for properties.

### 3.9.3 Using Interdependent Property Values for Searching

You can also use predefined properties with interdependent values for the search. To do this, you have to activate the parameter `Indexable` in the configuration of all properties. You have to specify the base property in the search options set you use in order to be able to specify the properties in the search options. You do this in the parameter `Allowed Predefined Properties`. To modify the search options using the configuration *iView*, choose *Content Management* → *User Interface* → *Search* → *Search Options Set*.



The base property is then displayed in the search dialog box as soon as a document with this property is indexed.

[www.sdn.sap.com/irj/sdn/howtoguides](http://www.sdn.sap.com/irj/sdn/howtoguides)

