How-to Guide
SAP NetWeaver ‘04

How To...
Create XML Forms
Version 3.00 – July 2004

Applicable Releases:
SAP NetWeaver ’04 Support Package Stack 04
(Portal, Knowledge Management)
© 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.
Table of Contents

1 SCENARIO .................................................................................................................. 2

2 THE STEP BY STEP SOLUTION: NEWS ................................................................. 4
  2.1 Before You Begin .................................................................................................. 4
  2.2 Creating a Project in the XML Forms Builder .................................................... 5
  2.3 Defining a Data Schema ....................................................................................... 7
  2.4 Modeling Forms .................................................................................................. 10
      2.4.1 Edit Form .................................................................................................... 10
      2.4.2 RenderListitem Form .................................................................................. 15
      2.4.3 Show Form ................................................................................................ 17
  2.5 Creating iViews .................................................................................................. 18
      2.5.1 Creating a News Authoring iView ............................................................. 18
      2.5.2 Creating a News iView for Readers ......................................................... 19
      2.5.3 Further Steps ............................................................................................. 19

3 THE STEP BY STEP SOLUTION: COOKBOOK ...................................................... 20
  3.1 Creating the Project and Data Schema ............................................................... 20
  3.2 Edit Form ............................................................................................................ 21
      3.2.1 Mandatory Fields ....................................................................................... 23
      3.2.2 Validating Data Types ............................................................................... 25
      3.2.3 Thumbnail Browser .................................................................................... 26
  3.3 RenderListitem Form .......................................................................................... 29
      3.3.1 Creating a Layout with 2 Columns ............................................................. 31
      3.3.2 Control Selectors ....................................................................................... 32
  3.4 Show Form ............................................................................................................ 36
      3.4.1 Setting a Background Image for a Show Form ............................................ 38
  3.5 Creating iViews .................................................................................................. 39
  3.6 Alternative List Depictions .................................................................................. 40
      3.6.1 Creating a Form in the XML Forms Builder .............................................. 40
      3.6.2 Creating XML Forms Renderers ................................................................. 41
      3.6.3 Creating a Layout Set ................................................................................ 41
      3.6.4 Creating an iView ...................................................................................... 42
This How-to guide describes how to create XML forms using the XML Forms Builder and display them in SAP Enterprise Portal using iViews.

XML forms are suitable for document types that always have the same or a similar structure. For example, a news item always has a title, an author, a creation date, and body text. This How-to guide describes the creation of XML forms using the following two example scenarios:

- **News**: In this example scenario, simple forms with basic functionality are created. The fundamental work steps are explained in detail.
- **Cookbook**: In this example scenario, forms with more comprehensive functionality are created. This scenario assumes that you already know the fundamental work steps of the news scenario.

**News**

The company *Open Corp* wants to create company-internal news items in SAP Enterprise Portal. All employees of the company should see a list of current company news items on their initial page.

These news items are to be created and managed directly in SAP Enterprise Portal by employees of the Communications department. The Communications department has its own page in SAP Enterprise Portal for this purpose. This page can only be seen by employees of this particular department.

After you have carried out the work steps in this scenario, the following iViews will be available to you:

- A *News Authoring* iView in which people responsible for content (such as employees of the Communications department) can create, change, and delete news items. The elements of a news item (for example, title and text) can be created in a simple form. The authors of the news items do not need to deal with formatting the text. As well as text, images and links to further information can be integrated into the news items.
- A *News* iView that displays a list of the titles of news items to readers. This iView can be placed on the initial page for portal users. When users click on the title of one of these news items, the entire text, including any images or links, is displayed.
Cookbook
A manufacturer of kitchen utensils wants to use XML forms to publish recipes. This requires two iViews in SAP Enterprise Portal – one for creating and maintaining recipes, and another for displaying them. The list of recipes has an identical layout in both iViews, but different commands are available.

Later on, another target group needs a new layout for the list of recipes. However, the original target group wants to keep the original layout. To achieve this, you create a new form in the XML Forms Builder, in order to display the same content using a new layout. You then display the new form in a third iView.

When you have carried out the work steps in this scenario, the following iViews will be available to you:

- **An Authoring iView** that employees can use to create, change, and delete recipes. Before saving a recipe in an *Edit* form, the system checks that all mandatory fields are filled in and that data entered has a valid format. Employees can add an image from a freely defined default folder to the recipes. If no suitable image is available in the default folder, they can upload an image to this folder from their local PC.

- **A Display iView** that interested parties can use to view a list of recipes. An image displays how spicy each recipe is (in this example, this is indicated by one, two, or three chili peppers). This function can also be used to visualize properties such as the status of each recipe. When users click on the name of a dish, the recipe is displayed with a list of ingredients and cooking instructions.

- A third iView that displays the list of recipes using a different layout from the first two iViews.
2 The Step By Step Solution: News

You need the following roles in SAP Enterprise Portal in order to carry out the work steps described here:

- **Content Manager** for working with the XML Forms Builder
- **Content Administrator** for creating the iViews


2.1 Before You Begin

Make sure that the Sun Java Runtime Environment (JRE) is installed on the client host in version 1.4 or higher (or JRE version 1.3.1_02 or higher).

Only one version of the Java Runtime Environment should be installed on the client host.

To check whether the Internet Explorer is using the correct JRE version, choose **Tools** → **Internet Options** → **Advanced**.
### 2.2 Creating a Project in the XML Forms Builder

1. Start the XML Forms Builder.
   
   If you are assigned to the *Content Manager* role, the XML Forms Builder appears in the top-level navigation bar under *Content Management → XML Forms Builder*.

2. Make yourself familiar with the screen areas of the XML Forms Builder.

3. Open the project options by clicking on ![Options](image)

4. Enter a project ID and name on the *Project* tab page.
   - The project ID must be unique. It may not contain special characters.
   - The project name is displayed on the flexible user interface in the commands for creating news items.

5. Save the project by clicking on ![Save](image)
Project Files.

The system carries out a project check (can be set under Options → Project).

You may receive a warning telling you that your project allows HTML output. You should refer to the security-relevant information in the standard documentation at the following paths:

- Knowledge Management → Administration Guide → System Administration → System Configuration → Content Management Configuration → Form-Based Publishing
- Knowledge Management → Content Manager Guide → XML Forms Builder → Controls → Label Controls → HTML in Label Controls

If you want to completely forbid the interpretation of HTML code in your project, proceed as follows:

6. Choose Options → Forms.

7. From the dropdown list HTML Code (HTML Markup), choose Strip All Tags. Confirm with OK.

Tip:

By default, the system carries out a project check every time you save. If you want to deactivate this, choose Options → Project and deselect Check Project on Save.

To start the project check manually, choose Extras → Check Project.
2.3 Defining a Data Schema

A data schema depicts the content structure of an individual news item. You define data schemas in the XML Forms Builder in the left-hand area of the screen.

1. Rename the root node of the data schema.
   
   The root node of the data schema is the first node beneath DataSchema (by default, DataModel).
   
   **Tip:** If a symbol with a plus sign (⁺) is displayed next to DataSchema, double-click on it to expand the node.
   
   Use the secondary mouse button to click on the element DataModel and then choose Rename Node.

2. Enter the name News in the dialog box that appears, and confirm with OK.

3. Create a subnode for each element of a news item.
   
   Use the secondary mouse button to click on the element News, and choose Add Child.
   
   Enter the name of the element (for example, Title) in the dialog box that appears.
4. To create other elements on the same level, choose *Add Sibling Node* from the context menu of the appropriate element (such as *Title*).

5. Repeat this procedure to create the following structure:

```xml
<News>
  <Title>
  <Author>
  <Date>
  <Body>
</News>
```

**Tip:**
You can use Drag&Drop to change the sequence of the elements in the data schema.
6. Assign suitable data types to the elements.

By default, data schema elements have the data type *string*, which allows any character string to be entered. Select the required element and choose a suitable data type from the dropdown box on the *Type* tab page.

You can use data types to validate user entries in the *Edit* form. If a user enters an invalid value, the system prompts him or her to correct the entry. The user is only allowed to save the form when all invalid values have been corrected.

In the case of some data types, the system is able to complete entered data. For example, if you enter the character string `www.sap.com` into a field with the data type *URL*, the system adds the http protocol in order to complete the URL (`http://www.sap.com`).
2.4 Modeling Forms

This section tells you how to create a set of simple forms with basic functionality. The system creates the following forms for each project by default:

- **Edit:**
  This form is for creating and changing news items. One news item is created in each *Edit* form.

- **RenderListItem:**
  This form is for creating list entries. You can use this form for both the *News Authoring* view and for displaying news items. You define the layout for a list entry on the form. You define the availability of controls later on using the configuration of the flexible user interface.

- **Show:**
  This form is for displaying the content of a single news item.

2.4.1 Edit Form

The *Edit* form is for creating and changing news items.

Steps 1 to 5 show two ways of creating controls. You can choose the method according to your personal preference.

1. Click on the text field control in the function bar.
2. Hold down the shift key and select the three elements *Title*, *Author*, and *Date*.
3. Use the primary mouse button to pull the data schema elements onto the *Edit* form using Drag&Drop.
4. Now use the secondary mouse button to pull the Body element onto the form using Drag&Drop. A context menu appears from which you can select the object that you want to create.

5. Choose Label and Control ➔ Text Area.

**Note:**

You can use the property Drag&Drop of the form in question to define which objects are created by default using the primary mouse button.

Drag&Drop via the secondary mouse button always displays a context menu that you can use to create any control.
The system has entered a reference to the relevant data schema in the properties of all input fields. This reference makes sure that the content created in the form at runtime is stored in the appropriate element of the data schema in the XML file.

You can check the schema reference by selecting an input field and viewing the assignment in the table at the bottom right-hand side.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>TextField</td>
</tr>
<tr>
<td>Caption</td>
<td></td>
</tr>
<tr>
<td>Maximum L...</td>
<td></td>
</tr>
<tr>
<td>Style</td>
<td>Standard</td>
</tr>
<tr>
<td>Error Style</td>
<td>Standard</td>
</tr>
<tr>
<td>Alignment</td>
<td></td>
</tr>
<tr>
<td>Vertical Alig...</td>
<td>Top</td>
</tr>
<tr>
<td>Size</td>
<td>30</td>
</tr>
<tr>
<td>Error Text</td>
<td></td>
</tr>
<tr>
<td>Validation ...</td>
<td>Append</td>
</tr>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Mandatory</td>
<td>No</td>
</tr>
<tr>
<td>Schema R...</td>
<td>/DataSchema/DataModel/Title</td>
</tr>
</tbody>
</table>

6. Select the input field *Date* and enter the value `$locDate` in the *Caption* field.

This causes the field to be filled with a suggested value (that is, the current date) at runtime.

The alias `$locDate` returns the current date at the locale of the current user.
7. Choose **Extras → Create Buttons**.

8. Make your settings and then choose **OK**. The system generates the buttons for the *Edit* form.

9. Start the preview by clicking on the green arrow (▶). The system generates the transformations for the preview in a temporary folder. The preview is opened in a new browser window. The **Date** field is already filled, but you can overwrite the suggested value if you want.
10. To improve the layout, change the width of the controls and align them to each other.

Use the secondary mouse button to open a list of commands for aligning control.

You can set the font, font size, and so on, in the properties of labels.

Tip:
You do not have to close the browser window containing the preview each time you make changes. Instead, choose (Rebuild Preview) and then refresh the content of the browser window containing the preview.

Tip:
You can choose to use a browser control (input field with input help) for the Date field instead of a simple text field.

Delete the input field for the date, and create a browser control as described in steps 3 and 4.

Enter the following data in the properties of the browser control:

- **Button Caption**: Text to be displayed on the button (for example, Select Date)
- **Action**: Choose Date.
2.4.2 RenderListItem Form

This form is for creating list entries. You can use this form for both the News Authoring view and for displaying news items. You define the availability of controls later on using the configuration of the flexible user interface.

You define a single entry of a list on this form. If multiple news items are available, the system repeats this layout.

1. Pull three label controls (A3) from the function bar onto the form.

2. Establish references to the elements Title, Author, and Date.

To establish the references, use Drag&Drop to pull each element from the data schema onto the respective label control in the form.

The system enters the data schema element and its complete path into the SchemaReference property of the label control.

The reference makes sure that the content that is stored in the corresponding element of the data XML file is displayed in the label control at runtime.
Tip:
Instead of carrying out steps 1 and 2, you can simply select the three elements while holding down the shift key and use Drag&Drop to place them on the ListItem form. This allows you to create three label controls with preset data schema references in one step.

3. Choose the value Display for the title in the Hyperlink property of the label control.

This means that the user can click on the title in order to display the entire news item whose depiction you defined in the Show form.
2.4.3 Show Form

1. Create label controls as on the RenderListItem form.
   On this form, you also create a label control for displaying the body text.

2. Generate the project by clicking on 🎨.
   After generating the project you can use the forms in the portal.
   A command for creating news items appears in iViews with the flexible user interface.

Result:
You have created the forms and can now create iViews in which you will be able to create and display news items.

Note:
If you cannot see the command, check the settings for form availability in the configuration (System Administration → System Configuration → Knowledge Management → Content Management → Form-Based Publishing → Forms Availability).
If you want to temporarily prevent your project from being visible in the portal, choose 📦 (Options) in the XML Forms Builder and select Hidden on the Project tab page.
2.5 Creating iViews

This example tells you how to create the following two iViews:

- **News Authoring** iView for authors (allows you to create and manage news items)
- **News** iView for end users (only allows you to display news items)

Both iViews use the layout of the `RenderListItem` form. You define which commands are available in the iViews using the KM configuration.

### 2.5.1 Creating a News Authoring iView

1. **Choose Content Administration ➔ Portal Content** from the top-level navigation bar.

2. Create an iView based on the **KM Navigation iView** in the Portal Content Directory.

3. Create the following iView properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layout Set</td>
<td>NewsExplorer</td>
</tr>
<tr>
<td>Layout Set Mode</td>
<td>Exclusive</td>
</tr>
<tr>
<td>Path to Initially Displayed Folder</td>
<td>/documents/myNews</td>
</tr>
<tr>
<td>Path to Root Folder for Navigation</td>
<td>/documents/myNews</td>
</tr>
</tbody>
</table>

4. Save your entries.

**Result:**

You have created a **News Authoring** iView.

Click on **New <Project_Name>** or on **Edit** to open the **Edit** form.

News items that you create are displayed in the list using the layout that you defined in the `RenderListItem` form.
2.5.2 Creating a News iView for Readers

1. Repeat steps 1 to 3 of the News Authoring iView procedure and create the following iView properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layout Set</td>
<td>NewsBrowser</td>
</tr>
<tr>
<td>Layout Set Mode</td>
<td>Exclusive</td>
</tr>
<tr>
<td>Path to Initially Displayed Folder</td>
<td>/documents/myNews</td>
</tr>
<tr>
<td>Path to Root Folder for Navigation</td>
<td>/documents/myNews</td>
</tr>
</tbody>
</table>

2. Save your entries.

**Result:**

You have created a News iView for readers. The news items are displayed in the list with the layout that you defined in the RenderListItem form. Click on the title of a news item to display the entire news item in the layout that you defined in the Show form. In contrast to the News Authoring iView, no commands are available here for creating or editing the news items. The commands that are available are defined by the flexible user interface.

You can choose a different path for storing news items in both iViews.

2.5.3 Further Steps

You carry out the following work steps so that the iViews are displayed in the portal:

1. Place the News iView on a page that is available to all employees.

2. Place the News Authoring iView on a page that is only available to people who are supposed to create news items.
3 The Step By Step Solution: Cookbook

This section contains instructions for more advanced functions. The functions described here do not need to be integrated in the order in which they are described.

3.1 Creating the Project and Data Schema

1. Choose (New Project) in the XML Forms Builder and create a project with the ID Cookbook and the name Recipe.

2. Create a data schema like the one in the screenshot.
3.2 Edit Form

1. The completed Edit form should look like the one in this screenshot.

2. The completed Edit form appears as follows in the XML Forms Builder.
   The following steps show you how to construct it.

3. Create a one-line input field (text field control) for the following elements:
   - Title
   - Preparation_Time
   - Cooking_time
   - Servings
   - Date
   - Author
4. In the control property of the fields Date and Author, enter the prescribed values $locDate and $sap_user.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>Text Field</td>
</tr>
<tr>
<td>Caption</td>
<td>$locDate</td>
</tr>
<tr>
<td>Maximum L.</td>
<td></td>
</tr>
<tr>
<td>Style</td>
<td>Standard</td>
</tr>
<tr>
<td>Error Style</td>
<td>Standard</td>
</tr>
<tr>
<td>Alignment</td>
<td></td>
</tr>
</tbody>
</table>

5. Create a multi-line input field (text area control) for the element Description.

6. Create an HTML editor control for the elements Ingredients and Directions.

This is a ‘what you see is what you get’ editor that allows simple text formatting.

Tip:
The references are preset if you use Drag&Drop to create the controls from the data model (see 2.4.1 Edit Form).

Choose Extras → Create Buttons and create two buttons for saving and canceling a save.
3.2.1 Mandatory Fields

The title of the recipe must be a mandatory field.

1. Choose the value Yes in the Mandatory property.

2. Enter an error message into the Error Text field. This error message is displayed on the screen if the author tries to save the Edit form without entering a value into the input field.

3. The system also generates error messages at validation. Use the property Validation Error Text to define whether or not they should be displayed.

The label must be linked to an input field control in order for an asterisk to be displayed next to mandatory fields.

If you created the input field using Drag&Drop from the data model (see 2.4.1 Edit Form), the link is already predefined.

To find out whether the link already exists, click on the field label. The linked input field is highlighted.
If there is no link, link the controls as follows:

4. Use the secondary mouse button to click on the Title label and choose Link to Control.

5. Click on the field label.
   The linked input field is now highlighted.

**Result:**

When the user tries to save but has left the field empty, a border appears around the field and an error message is displayed.
3.2.2 Validating Data Types

Only numerical values should be entered into the fields Preparation Time, Cooking Time and Servings. The system should also check whether the author has entered a valid date into the Date field.

1. Select the data schema element Preparation_Time.

2. Go to the tab page Type.

3. Choose the value Integer from the dropdown box.

4. Enter an error message into the Error Text property of the input field (for example, Please type in an integer).

   This error message is displayed if the author tries to save when there is an invalid value in the field.

5. Make sure that the field Preparation Time in Minutes is linked with the input field (see 3.2.1 Mandatory Fields).

   This link ensures that the field label is bordered in red if the author tries to save when there is an invalid value in the field.

6. Repeat steps 1 to 5 for the elements Cooking_time und Servings.

7. Repeat steps 1 to 5 for the element Date, but choose the data type Date from the dropdown box.
3.2.3 Thumbnail Browser

The author of a recipe should be able to insert an image of the prepared dish. All images are stored in a folder in a KM repository (/documents/Images). If no suitable image is available there, the author should be able to upload an image from a local PC to the folder /documents/Images.

1. Set the data type *Image* for the data schema element *Picture*.

2. Create a browser control with a reference to the data scheme element *Picture*.
3. Choose the value *Thumbnail* in the control property *Action.*
   This defines the type of input help.

4. Use the field *Start In* to specify the path to the folder to be opened (for example, */documents/Images*).
   If you leave this field empty, the root path to the project is opened. You defined this in the options of the XML Forms Builder under *Options → Paths*.

5. Choose the value *Image* in the control property *Button Type*, and then select an image in the field *Button Image*.
   This image is displayed to the right of the input field instead of the button.
   You can also use a button of the type *Button* here. This example is meant to show you how you can include your own buttons in forms.

**Result:**
When an author clicks on the icon next to the input field, a dialog box for selecting an image appears.

To upload a new image, click on the popup indicator (⟩) next to the breadcrumb, and then choose *New → Upload.*
You have now made the Edit form available. Only the combobox Spicevalue needs to be added. You create this in the section 3.3.2 Control Selectors.
3.3 RenderListItem Form

You want to display a list of recipes in an iView. You want to display the name of the dish (title), a description, and an image of the prepared dish.

1. Create a label control for the title and the description, and an image control for the image in the bottom left-hand corner.
   Make sure that the controls have references to the appropriate data schema elements.

2. Choose the value Display in the Hyperlink property of the Title control.
   This means that users can display the entire recipe that you defined in the Show form, simply by clicking on its title.

3. Set the required font and size in the properties of the label control.

4. Generate the project.
   You can now test the creation of recipes in the News Authoring iView (Content Management → Explorer → News).
   The chili pepper images are not yet displayed.
Tip:

Choose *Defined Size* in the *Display* property of the image control and then set the values for *Width* and *Height* if you want all the images displayed to be the same size.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image URL</td>
<td>Schema Reference</td>
</tr>
<tr>
<td>Hyperlink</td>
<td></td>
</tr>
<tr>
<td>Condition</td>
<td></td>
</tr>
<tr>
<td>Display</td>
<td>Defined Size</td>
</tr>
<tr>
<td>Alignment</td>
<td></td>
</tr>
<tr>
<td>Vertical Align</td>
<td>Top</td>
</tr>
<tr>
<td>Empty</td>
<td>Empty Column</td>
</tr>
<tr>
<td>Alternative</td>
<td></td>
</tr>
<tr>
<td>Link Desci...</td>
<td></td>
</tr>
<tr>
<td>Border</td>
<td>0</td>
</tr>
<tr>
<td>Schema R...</td>
<td>JDataStreampicturePic...</td>
</tr>
<tr>
<td>Property R...</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>24</td>
</tr>
<tr>
<td>Y</td>
<td>24</td>
</tr>
<tr>
<td>Width</td>
<td>80</td>
</tr>
<tr>
<td>Height</td>
<td>80</td>
</tr>
</tbody>
</table>
3.3.1 Creating a Layout with 2 Columns

You want to display two recipes next to each other, as shown in the screenshot to the right. You change the collection renderer to do this.

1. Choose System Administration → System Configuration → Knowledge Management → Configuration.

2. In the KM configuration iView, choose Content Management → User Interface → Settings → CollectionGridRendererSettings.

3. Duplicate the NewsCollectionRenderer. Rename the new collection renderer, for example, as NewsCollectionRenderer_2cols.

4. Set the parameter *Number of Columns* to 2, and save your entry.

5. In the KM configuration iView, choose Content Management → User Interface → Settings → Layout Sets, and include the NewsCollectionRenderer_2cols in the layout set that you are using. Create a new layout set if necessary.
3.3.2 Control Selectors

You want to display images with one, two, or three chili peppers in the list in order to indicate how spicy each dish is.

The author of the recipe should be able to choose the spiciness of the dish from a dropdown list. The system then shows one, two, or three chili peppers in the list depending on the entry chosen by the author.

1. In the configuration of Content Management, create a property group Recipe and a document property Spicevalue with the permitted values hot, very hot, and extremely hot (System Administration → System Configuration → Knowledge Management → Content Management → Global Services → Property Metadata).
2. To reload the properties from the configuration in the XML Forms Builder, use the secondary mouse button to click on Properties on the tab page Data Model, and then choose Reload Properties. The property appears under the name of the group.

3. Create a combobox control with a reference to the property on the Edit form. The system enters the document property with its complete path into the property Property Reference of the combobox.
Note:

Comboboxes with references to a document property are dynamic. The system reads the permitted values from the configuration at runtime (see step 1) and displays them in the dropdown list.

You do not have to define any list entries in the XML Forms Builder.

4. Go to the RenderListItem form and create a control selector with reference to the property.

5. Use the secondary mouse button to click on the control selector and then add three images.
6. Select the images in the tree view of the control.

7. In the field Image URL, choose the command Select and then select the image in question.

8. Enter one of the permitted values for the property Spicevalue into the property Condition.

9. Generate the project.
3.4 Show Form

The completed *Show* form should appear as shown in the screenshot to the right.

1. Create label controls for the following elements:
   - Title
   - Description
   - Preparation Time
   - Cooking Time
   - Servings
   - Ingredients
   - Directions
   - Author
   - Date

**Note:**

The content of the label controls *Ingredients* and *Directions* is created in the HTML editor. You should permit HTML display for these two label controls (and only these two controls). Refer to 2.2, step 7.
2. Note that you need two label controls for most elements (such as *Preparation Time*) in this example:

- One label control that is a field label (font color = brown in the screenshot above)
- One label control that is an output field for the content of the data schema element, and must therefore have a reference (font color = black in the screenshot above)

If the author of a recipe leaves a field blank in the *Edit* form, the corresponding field label in the *Show* form should be hidden.

3. Use the secondary mouse button to click on the field label, and choose *Link Visibility*.

4. Click on the output field. The linked output field is now highlighted.

**Result:**

If the author leaves the field *Servings* blank, the field label is hidden in the *Show* form.
3.4.1 Setting a Background Image for a Show Form

You want to set a background image for the Show form as in the screenshot to the right.

1. Select the Show form by clicking on the root node of the tree view or on an empty space in the form.

2. Click on the button next to the Background Image field and select an image from a KM repository.

3. Choose the value Repeat for the Background Repeat Mode so that the image is tiled across the entire background.
3.5 Creating iViews

Create two iViews. The procedure is the same as for the News project (see 2.5 Creating iViews).

The layout sets NewsExplorer (creating and managing news items) and NewsBrowser (display only) can be used for all types of objects that can be created using XML forms – not only for news items.

The path where you store the recipes could be /documents/Recipes.
You want to display the list in a new iView using a different layout to that used in the first two iViews.
You want the image to be larger, and you do not want the description to be displayed.

3.6.1 Creating a Form in the XML Forms Builder

1. Create a new form (for example, General) in the XML Forms Builder.
   To do this, use the secondary mouse button to click on the bar above the forms, and choose Add Form.

2. Create controls with references to the data schema.
3.6.2 Creating XML Forms Renderers

1. In SAP Enterprise Portal, choose System Administration → System Configuration → Knowledge Management → Configuration.

2. In the configuration iView, choose Content Management → User Interface → Settings → Resource Renderer Settings → XML Forms Renderer Settings → New.

3. Enter data as displayed in the screenshot and save your entries.
   In the field XML Forms Transformation, enter the name of the form that you created in step 3.6.1.

3.6.3 Creating a Layout Set

1. In the configuration iView, choose Content Management → User Interface → Settings → Layout Set.

2. Duplicate the layout set NewsBrowser.

3. Deselect the resource renderer used in the News Browser and select the XML forms renderer that you created in step 3.6.3 (for example, General).

4. Save your changes.
3.6.4 Creating an iView

1. Choose **Content Administration → Portal Content**.

2. Create an iView in an appropriate folder of the portal catalog. Use the iView template **KM Navigation iView**.

3. In the iView property **Layout Set**, enter the name of the layout set that you created in step 3.6.3.

4. In the iView properties **Path to Initially Displayed Folder** and **Path to Root Folder for Navigation** enter the name of the folder in which you store your recipes.