

Using the Category View Builder with IBM WebSphere Portlet Factory

August 31, 2006

© Copyright International Business Machines Corporation 2006. All rights reserved.

This article with the accompanying sample shows you how to use IBM® WebSphere® Portlet Factory Version 6 (hereafter called Portlet Factory) to categorize tabular data. It is intended for portlet developers using Page Automation.

This article is one in a collection of articles and samples that illustrate techniques for developing with Portlet Factory. See the [Portlet Factory Product Documentation](#) page for a complete list of these. For an introduction to developing with Portlet Factory, complete the introductory tutorials that are available both in the product help and on that web site.

Prerequisites

You should have a basic familiarity with Portlet Factory and be able to create and run Portlet Factory models.

Introduction

Data displayed using tables is often more useful if application end users can sort and group the data. A Data Column Modifier builder is used to add sorting while a Category View builder provides grouping of data. These data groups are displayed in a table using expandable / collapsible sections. A Category View builder can also categorize data on more than one column to create categories and sub-categories.

Sample description

Categorizing on a Single Column. By adding a Category View builder call to a model that uses Page Automation, a flat table of data can be displayed with expandable / collapsible sections. The data inside each section is grouped by the values in the categorized column. It is important to also use a Data Column Modifier (DCM) to move the categorized column to the extreme left of the table of data. Without the DCM, the `STATUS` column displays its categories in the middle of the table instead of at the far left. It is easy to see that the table looks better when a DCM is used.

Adding Sorting to Categorized Columns. Data Column Modifier builders can be used to add sorting to column headings in a table. It is a good practice to add sorting to the column headings for all categorized columns so that the categories can also be sorted in reverse order.

Ensuring the Data get Categorized. The Data Categorization section of the Category View builder is very important. A method that categorizes the data is invoked or not based upon the settings chosen here. Since the data is fairly simple, choosing `Auto` works best. By default, categorization is set to `None`, therefore, it is important to remember to set this input. The included models have this input set to `Auto`. If the data is already known to be sorted by the column being used for categorization, however, `None` is an acceptable choice.

Categorizing on a more than one Column. When categorizing on two columns, the idea is that the leftmost column is a category and the column to its right is a sub-category. The `category_view2` model categorizes on both `Status` and `Billing`. This arrangement allows the view to show `Billing` as a sub-category of `Status`.

Table 1. Sample package contents

Filename and location	Description
WEB-INF/models/samples/category_view/category_view.model	Model that demonstrates categorizing on a single column.
WEB-INF/models/samples/category_view/category_view2.model	Model that demonstrates categorizing on two columns.
WEB-INF/models/samples/category_view/data/OrdersServiceProvider.model	Service Provider model containing more data than the standard sample.

Figure 1. Sample model running in a web browser

The screenshot shows a web browser window with the address bar containing the URL: `http://localhost:10038/FirstTwoSamples_b/webengine/samples/category_view/category...`. The browser interface includes a menu bar (File, Edit, View, Favorites, Tools, Help), a toolbar with navigation buttons (Back, Forward, Home, Stop, Refresh), a search bar, and a 'Go' button. The main content area displays a table with the following columns: Status, Order ID, Date Ordered, Date Shipped, Quantity, Amount, Billing, Shipped, and State. The table is organized into expandable/collapsible sections: 'In Process' (collapsed), 'Out of Stock' (expanded), 'Returned' (expanded), and 'Shipped' (collapsed). The 'Out of Stock' and 'Returned' sections contain several rows of data, each with a unique Order ID and associated details.

Status	Order ID	Date Ordered	Date Shipped	Quantity	Amount	Billing	Shipped	State
▶ In Process								
▼ Out of Stock								
	000715	2000-02-12		4	457.72	Credit	false	California
	009116	2002-09-16		32	279.63	Credit	true	California
	006555	2002-10-06		75	711.18	PO	false	Texas
	007309	2001-04-07		71	639.16	Credit	true	California
	002398	2001-08-18		46	179.31	PO	false	New Hampshire
	005193	2002-08-12		15	254.51	PO	false	California
▼ Returned								
	009532	2001-09-02	2001-09-09	8	306.34	Credit	false	Massachusetts
	006258	2002-02-12	2002-02-13	78	176.99	Credit	false	Massachusetts
	009961	2001-09-02	2001-09-08	54	58.59	Credit	false	Massachusetts
	008156	2000-08-02	2000-08-06	59	179.89	PO	false	Texas
▶ Shipped								

Instructions for running the sample

To run the sample application:

1. Create a project or use an existing project that contains the `Tutorials` and `Samples \ Applications` Feature Set
2. Download the sample ZIP file and import it into the project using `File > Import > WebSphere Portlet Factory Archive` command.
3. Open the `WEB-INF\models\samples\category_view\category_view.model` and run it. This model demonstrates categorizing on a single column.
 - Experiment with the expandable / collapsible sections.
 - Temporarily disable the Data Column Modifier (DCM) builder call and run the model again. You will see why it is important to use a DCM to move the categorized column to the left. Enable the DCM builder call when finished.
 - Expand all of the `Status` categories.
4. Open the `WEB-INF\models\samples\category_view\category_view2.model` and run it. This model demonstrates categorizing on two columns.
 - Experiment with the categories and subcategories: `Status` and `Billing`.

Additional considerations

- This builder also works well with the Domino View and Form builder. It can use a categorized view in a Domino database and rely on Domino to implement the categorization. It can also categorize a flat Domino view.

Resources

WebSphere Portlet Factory product documentation

<http://www.ibm.com/developerworks/websphere/zones/portal/portletfactory/proddoc.html>

WebSphere Portlet Factory support

<http://www.ibm.com/software/genservers/portletfactory/support/>

developerWorks forums

http://www.ibm.com/developerworks/forums/wsdd_forums.jsp

Trademarks

- DB2, IBM, Lotus, Tivoli, Rational, and WebSphere are trademarks or registered trademarks of IBM Corporation in the United States, other countries, or both.
- Windows and Windows NT are registered trademarks of Microsoft Corporation in the United States, other countries, or both.
- Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Other company, product, and service names may be trademarks or service marks of others.