

IBM Software Demos WebSphere Dashboard Framework

IBM WebSphere Dashboard Framework is a powerful and flexible tool for rapidly building standards-based, active dashboards. It helps reduce development and maintenance costs by simplifying and shortening the entire dashboard development cycle.

<0:16>

Maureen LeClaire, an application developer for Renovations Corporation, uses **WebSphere Dashboard Framework** to build a sales dashboard. **WebSphere Dashboard Framework** plugs into **WebSphere Portlet Factory**, and adds dashboard-specific features and design components.

Maureen starts by building an “Opportunities by Stage” portlet. To do this, she first creates a new portlet container called a Model.

<0:47>

She then builds the portlet functionality by snapping together a set of reusable software automation components called Builders. Builders automate the creation of code through intuitive, wizard-like interfaces. As such, they help to dramatically speed dashboard development.

<1:05>

Since the sales data for this portlet comes from **Microsoft Excel**, Maureen selects the Enhanced **Excel** Import Builder. She could also easily create portlets that source data from a wide variety of systems such as **SAP** or **Siebel**.

<1:20>

Once the Builder is selected, she uses the interface to fill out the Builder inputs.

Maureen now selects the spreadsheet file that contains the sales data, and uses the Builder to find the content within the file. She selects the one sheet found, and specifies that the data in that sheet contains a header row.

<1:44>

When the Builder is saved, it generates the code to access the spreadsheet, including the schema that describes the data, a variable that will hold the data, and a service that can be invoked to retrieve the data and place it in the variable.

<1:59>

Using **WebSphere Dashboard Framework**, developers can easily create portlets that adhere to a service-oriented architecture. With just a few extra

IBM Software Demos

WebSphere Dashboard Framework

clicks, Maureen can expose this **Excel** data as a REST service or as a WSDL-described web service, for use by external parties and other developers within her company.

<2:21>

Next, Maureen uses Dashboard-specific Builders included in **WebSphere Dashboard Framework** to create her portlet interface. She selects the “Summary Drill Down” Builder, which automates the creation of summary charts and tables.

<2:35>

Again, she fills out the Builder inputs, and uses a selection window to choose an initialization action from the data service created by the **Excel** Import Builder.

She then gives her portlet a title.

<2:50>

Maureen decides to display the data in both a summary chart and a summary table.

These components will be located on separate pages, with tabs to navigate between them.

<3:06>

In the chart properties, she selects a bar chart, with Stage Name displayed on the x-axis, and Average Deal Size and Total on the y-axis.

<3:22>

She also increases the chart width.

<3:31>

Before testing her portlet, Maureen adds another feature, an action menu that enables users to print the portlet and export the data to **Excel**. She adds these capabilities simply by checking the appropriate boxes.

<3:47>

She then saves her work and tests the portlet.

<3:52>

Both the summary chart and summary table are available as well as the ability to export the data.

<4:02>

IBM Software Demos WebSphere Dashboard Framework

Maureen still wants to add a few additional features, including the ability to sort on column headers, and a summary row that displays totals or averages for each of the columns.

{TRANSITION}

<4:18>

Later, we see that Maureen has added all of the additional features she needs, in just minutes.

To do this, she first added localization, so that column names now come from resource bundles.

<4:31>

She also added the summary row to the table, and a Status Indicator that will turn the “Average Days at Stage” field red, if the value in the field is greater than the current threshold of 50.

<4:47>

Maureen used a powerful Builder called “Rich Data Definition” to add formatting and sorting to the table.

<4:53>

And finally, she added a Builder called “Portlet Adapter” that enables her to automatically deploy this Model as a portlet.

Once again, Maureen tests the portlet to validate these changes.

<5:08>

Seeing that all changes have been added successfully, she is very pleased with the results.

<5:15>

IBM WebSphere Dashboard Framework enables companies to quickly deploy highly tailored business dashboards that consolidate data and processes from multiple sources. It also helps companies to reduce development costs and speed time-to-value for their dashboard initiatives.