

# Script Portlet Samples for IBM Script Portlet

## *Overview*

This is a set of examples that illustrate some basic techniques for using the IBM Script Portlet for WebSphere Portal. The samples show how to use some Portal features such as Public Render Parameters, and there are examples of using open source libraries.

To use these examples you will need to have WebSphere Portal 8.0.0.1 or later with the IBM Script Portlet installed. These samples were built and tested with the IBM Script Portlet release 1.1 from July 2014.

See the References section below for information on obtaining IBM Script Portlet

The samples include:

- Hello, World
- jQuery Eventing (two cooperating portlets)
- jqPlot Chart
- jQuery DataTables
- Load WCM Content
- Launch Script
- Public Render Parameters (two cooperating portlets)
- WCM tag samples
- Imported content files
- Portlet Preferences

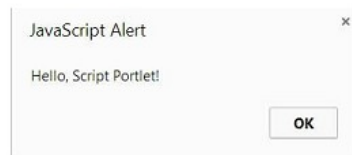
There are instructions for using the sample code near the end of this document.

## *Hello, world*

This is an extremely simple HTML snippet and JS function. Clicking the button shows a JS alert popup.

Hello, world

Click Me



## *jQuery Eventing*

This sample uses jQuery eventing, with `bind()` and `trigger()` functions. A list of customers is defined in the Customer List portlet, and a click event on each customer is used to fire the event by calling `trigger()` with a queue object. In the details portlet, the `bind()` function is called using the same queue object. When the event is received, it looks up a customer ID from the `customersData` array and updates the display.

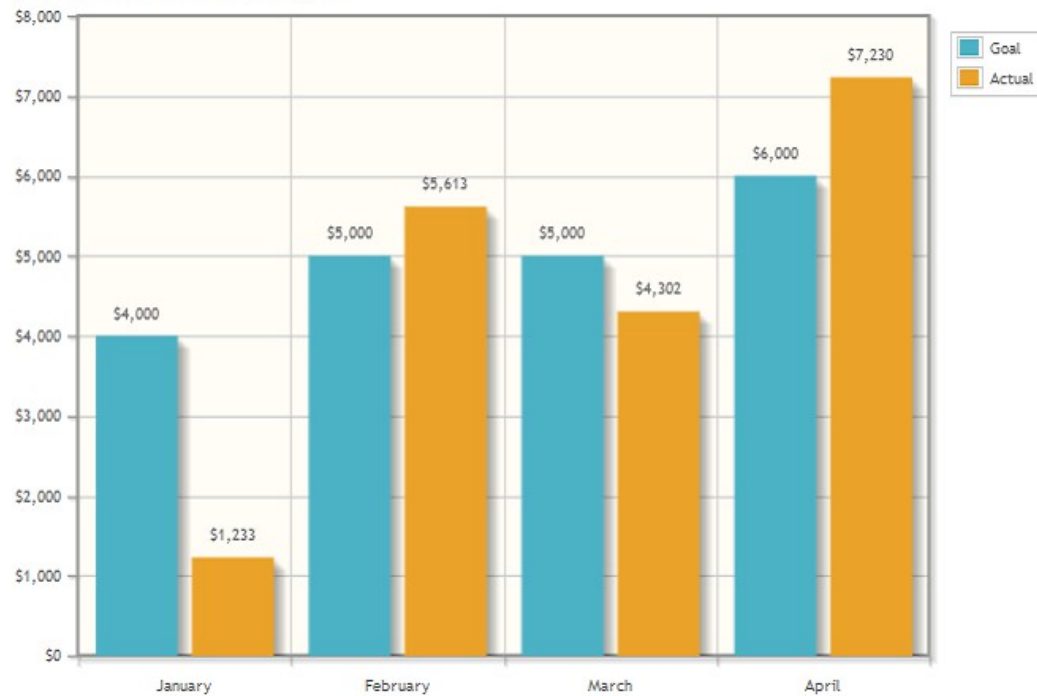
To use this sample, put two Script Portlets on the page side-by-side. In the left-hand portlet use the event\_source code, and in the right-hand portlet use the event\_target code.

Customer List		Customer Details	
Samantha Daryn	2143	Lucille Suarez	2144
Status: Standard Updated: 07/01/2012		\$1,050.00	Preferred Updated: 11/21/2012
\$2,350.00		303-555-2435 123 Main St Concord	<a href="#">Print Documents</a>
Lucille Suarez	2144		
Status: Preferred Updated: 11/21/2012			
\$1,050.00			
Amar Srivastava	2145		
Status: Inner Circle Updated: 08/12/2012			
\$7,235.00			
Ted Amado	2146		
Status: Standard Updated: 02/14/2012			
\$1,030.00			

### *jqPlot Chart*

This example show a chart created with the jqPlot open source charting library. The data for this chart in this example is defined right in the JS code. In a real world implementation the data would come from a REST service.

## jqPlot Chart Sample



## jQuery DataTables

This sample shows the jQuery DataTables library. The Customers data in this case comes from a JS variable. See this web site for more information on the jQuery DataTables library: <https://datatables.net/>

## jQuery DataTables

Show  entries

Search:

ID	Name	Balance	City
2143	Samantha Daryn	\$2,350.00	Salisbury
2144	Lucille Suarez	\$1,050.00	Concord
2145	Amar Srivastava	\$7,235.00	Sherman Oaks
2146	Ted Amado	\$1,030.00	Sydney
2147	Dan Misawa	\$1,300.00	Canterbury
2148	Frank Adams	\$3,440.00	Denver
2149	Steve Williams	\$1,110.00	San Jose
2150	Ron Espinosa	\$7,670.00	Dublin
2151	Ed El-Amon	\$1,800.00	Beijing
2152	Pierre Dumont	\$2,350.00	Pittsburgh

1 to 10 of 16 entries

◀ Previous Next ▶

### ***Load WCM Content***

This example shows how to load any WCM content via Ajax. In this example, the url points to the web content library and the “articles” site area. You can type any content in that site area and this script will load it and display it in a div.

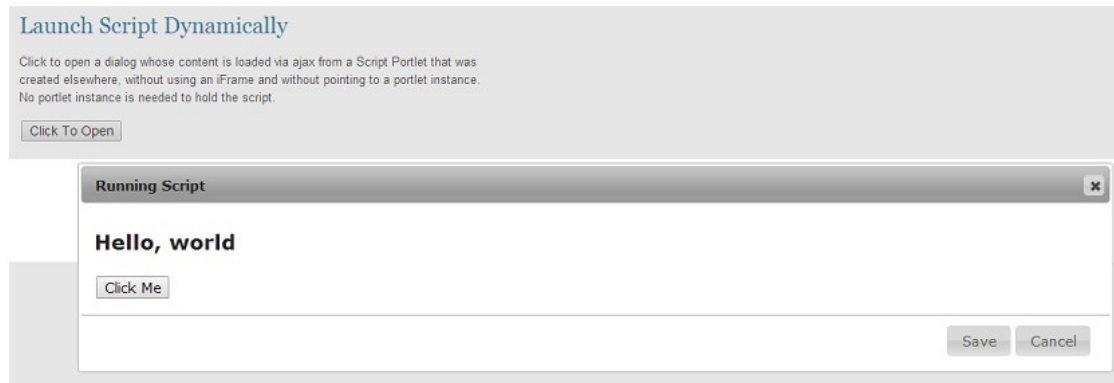


### ***Launch Script***

This example shows how to get the contents of a script via Ajax, and how to take that content and render it in a dynamically-created modal dialog. This can be useful when you want to launch dialogs/applications programmatically and you don't want to use iFrames, for example if you need to support mobile devices. You could also use the WCM Menu Component to pick which scripts should be displayed based on a Personalization rule.

NOTE: To run this sample you must update the URL in the JavaScript to the URL to a script content item in your WCM library. For example, the screenshot shows the “Hello, world” sample Script Portlet launched in a popup dialog. There are a few ways of finding the URL for a Script Portlet Content Item. One way is to use a browser tool to get the URL from the “src” attribute of the iFrame which is used by the preview window of the

Script Portlet Editor. Another way is to use the public WCM URL tag, or you could use the public WCM servlet rendering URL format.



### ***Public Render Parameters***

This sample shows how to set and retrieve a Public Render Parameter (PRP). The parameter that's used is the pre-defined "CUSTOM\_CONTEXT" parameter defined by the WCM Rendering Portlet. The top portlet shows a list of customer names that are defined in a JS variable and used to populate a SELECT list. When the form is submitted it sets the PRP value. The bottom portlet then retrieves the PRP value and uses it to select the customer from the list.

To use this sample, put two Script Portlets on the page, one above the other. In the top portlet use the prp\_set code, and in the bottom portlet use the prp\_retrieve code.

### **Set Public Render Parameter**

Select customer:

### **Retrieve Public Render Parameter**

Ted Amado

\$1,030.00

303-555-1234

44 Center St.

Sydney

2146

Standard

Updated: 02/14/2012

[Print Documents](#)

### ***WCM Tag Samples***

This sample shows how to use some of the WCM tags that are available. For example, you can retrieve the current user's name, or get a "namespace" value that will be unique for this instance on a page.

WCM Tags Sample

Tag	Example Value	Notes
User display name	Jonathan Booth	Uses a combination of two EL beans
LDAP cn	Jonathan Booth	Common Name from LDAP
LDAP sn	Booth	Surname from LDAP
LDAP uid	jbooth	UID from LDAP
LDAP preferredLanguage	en_US	Preferred language from LDAP
Authors	wpsadmin	Authors of this Script Content Item
Title	Script Portlet Content Item_3-WG	Title of this Script Content Item
Parent Title	Portal Context Sample	Title of the parent of this Script Content Item
Portlet Namespace	ns_Z7_N0LA11O0KO22F0AS8JOEJK3876_	A unique identifier for this portlet/page
Locale	en-US	Current rendering locale
Display only in page edit mode	Edit mode is ON	The value column is only shown when this page is in edit mode.

### ***Imported Content Files***

This sample shows the use of imported content: an image, a JSON data file, and an HTML fragment. All are loaded via Ajax calls. Each of these files is referenced with a relative link in the index.html page, and each relative link is replaced by a WCM Element tag when imported. The WCM element name is the same as the relative pathname. The JSON file and the HTML fragment are loaded by the application JavaScript code. The URLs are read from the hidden "a" tags that are in the index.html file.

The Javascript that loads the HTML fragment and the JSON file may not run standalone (from the file system) from all browsers due to security restriction.

## Imported Content Files

### Imported Image



### HTML Fragment Loaded by JS

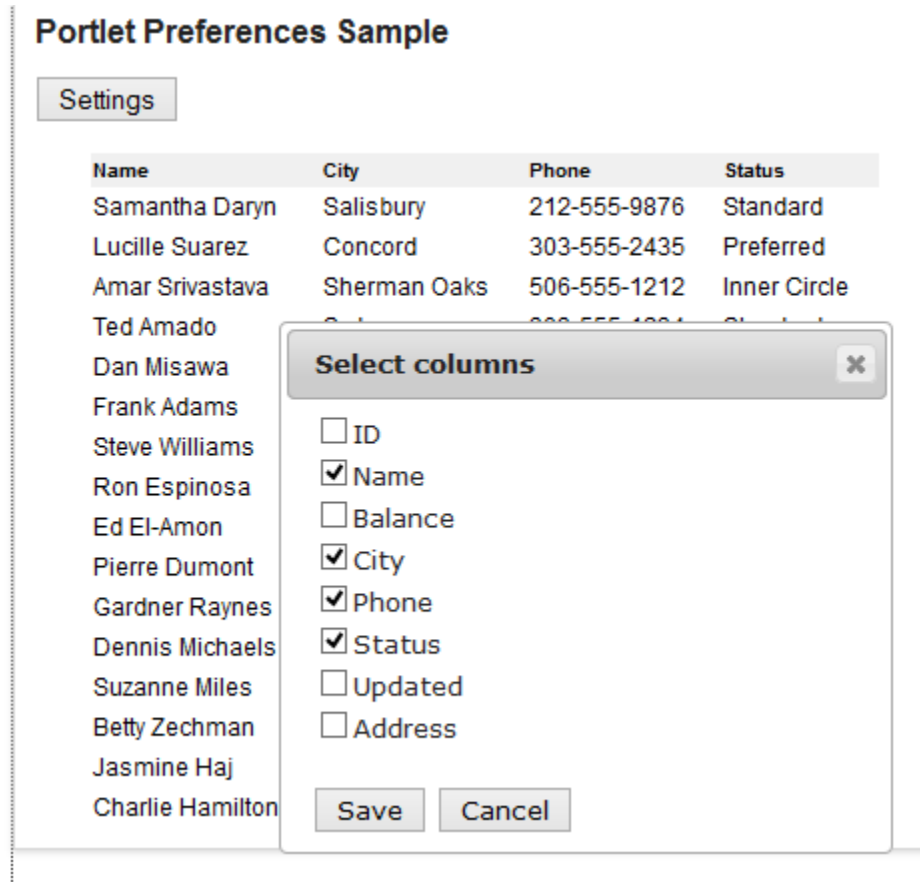
This text comes from a separate HTML fragment file, dynamically loaded into the main page by Javascript running in the browser.

### JSON Data Loaded by JS

Name	Address	City
Samantha Daryn	2001 West Rd.	Salisbury
Lucille Suarez	123 Main St	Concord
Amar Srivastava	South Mill Pond Ave.	Sherman Oaks
Ted Amado	44 Center St.	Sydney
Dan Misawa	1204 Mountain Boulevard	Canterbury
Frank Adams	133 Rock Boulevard	Denver
Steve Williams	1 Circle Lane	San Jose
Ron Espinosa	2425 Mountain Rd.	Dublin
Ed El-Amon	26B Seiching Road	Beijing

## ***Portlet Preference Files***

This sample shows the use of preference js api. The data that is read from a JSON file is displayed as a table using JQuery. The columns that are displayed in the table are controlled by preference values that are stored in the portlets preferences. This allows a user to use the same portlet to display different data depending on the preferences that are set for that instance of the portlet. For this sample the preferences are an array of strings that indicate weather a column is visible or not. The preference api supports the setting different kinds of preferences from simple strings, js arrays or JSON objects, it is up to you to identify the type of data your portlets need. Every script portlet can have preferences as needed and the preference definition can be customized to the specific instance.



### ***How to use the sample code***

First, unzip the script\_portlet\_samples.zip file onto your computer. Each folder contain the files for one Script Portlet. For each portlet there is an index.html file, a JavaScript file, sometimes a CSS file and a zip for all these individual files.

For the samples that have no dependency on WebSphere Portal, you can actually test the application on its own simply by using a web browser to view the index.html file on your file system. For example, the jqPlot Chart and jQuery DataTables samples can be tested in this way. Then you can create Script Portlets for each sample as described next.

Here are the steps for using the code for these samples:

1. Go to the Portal page where the portlet will be and go into Edit mode for the page. Add the Script Portlet to the page by selecting it from the Content tab in Edit mode. You will then see the Import button in the portlet.
2. Click on the import button and select the zip file from the respective samples folder.
3. Click the "Import" button.

You should then see the sample running in the Portlet.



### ***References***

The Script Portlet is available as a download on the Portal Catalog here: [IBM Script Portlet for WebSphere Portal](#)