Working with Business Console Gadgets Help

Version 10.1

October 2017

WEBMETHODS

This document applies to Software AG Designer Version 10.1 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: SBP-GADGETS-OLH-101-20171114

Table of Contents

About this Guide

This guide provides information about creating user interfaces with Composite Application Framework (CAF) and OpenUI.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies storage locations for services on webMethods Integration Server, using the convention *folder.subfolder:service* .
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
Italic	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
`Monospace font`	Identifies text you must type or messages displayed by the system.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at http://documentation.softwareag.com. The site requires Empower credentials. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

You can find product information on the Software AG Empower Product Support website at https://empower.softwareag.com.

To submit feature/enhancement requests, get information about product availability, and download products, go to Products.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the Knowledge Center.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at http://techcommunity.softwareag.com. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

1 Working with Business Console Gadgets

About Business Console Gadgets

Business Console gadgets are re-usable components that enable customization and personalization of business monitoring AppSpaces in Business Console.

Business Console offers built-in gadgets for creating AppSpaces. However, you can create your own gadgets by using wizards in Designer, deploy these new gadgets to a My webMethods Server, and re-use these gadgets for customizing Business Console.

When you design a gadget in Designer, you can specify:

- Whether or not to use AngularJS framework to create the gadget
- User interface and functionality of the gadget
- Custom parameters for configuring gadgets in Business Console
- JavaScript events for communicating between gadgets
- Remote server connections for the gadget
- REST invocation method for communicating with remote servers (whether to use Cross-origin resource sharing (CORS) support on remote servers or Business Console's proxy REST service)
- My webMethods Server on which the gadget must be deployed

You can manage access permissions for a gadget in My webMethods Server. For more information about assigning permissions, see *Administering My webMethods Server*.

For information about re-using gadgets and customizing Business Console, see *Working with webMethods Business Console*.

Creating a New Business Console Gadget

In the UI Development perspective, you can create gadgets in a portlet application project or a web application project in one of the following views:

- Solutions view
- Navigator view
- Package Explorer view

After you complete the gadget design, publish the web and portlet application projects to My webMethods Server to make it available in Business Console AppSpaces.

For information about creating and publishing a portlet or a web application project, see *webMethods CAF and OpenUI Development Help*.

Note: You need JavaScript knowledge for programming gadgets.

To create a new gadget

1. Select the UI Development perspective.
2. Select the web or portlet application project where you want to create a new gadget, and start the New Business Console Gadget wizard in one of the following ways:
   - In the Solutions view, expand **User Interfaces**, right-click on the project where you want to create a new gadget, and select **New Business Console Gadget**.
   - In the Navigator view or Package Explorer view, right-click on the project where you want to create a new gadget, and select **New > Other > Software AG > UI Development > New Business Console Gadget**.
3. In the New Business Console Gadget wizard, provide the specification for the new gadget. For information about what values to specify for the new gadget, see "New Business Console Gadget Wizard" on page 10.

   The New Business Console Gadget wizard creates the configuration files and definition file for the new gadget. For information about the configuration files and definition file for a gadget, see "Gadget Configuration Files" on page 13 and "Gadget Definition File" on page 15.
4. Configure and customize the gadget.
   a. Program the gadget depending on the gadget type by doing one of the following:
      - To create a gadget without using any pre-defined templates, edit the custom.js file.
      - To create a gadget using the AngularJS framework, edit the controller.js and custom.js files.
   b. In the Scripts and Styles folders for the gadget, create a script file and style file, respectively.
   c. Provide the gadget details on the **Gadget Definition Editor** tab. To view the **Gadget Definition Editor** tab, do one of the following:
      - In the Solutions view, double-click the gadget.
      - In the Navigator view or Package Explorer view, double-click the gadgetDefinition.xml file corresponding to the gadget in the web_or_portlet_application/WebContent/WEB-INF/gadgets/gadgetName_gadgetID folder.

      For information about the details to specify, see "Gadget Definition Editor" on page 23.

New Business Console Gadget Wizard

General Gadget Specification

In the New Business Console Gadget wizard, provide the project and gadget details as described in the table below:

Field / Control	Description
Project	Specifies the web or portlet application project selected for gadget creation.
Gadget Type	Specify one of the following type: ■ **AngularJS** if you want to use AngularJS framework for the new gadget . For example, the AngularJS framework will contain a template for invoking REST services. ■ **Default with empty stubs** if you want to design the new gadget from scratch without using any pre-defined templates.
Gadget Root Directory Name	(Optional) Specify a name for the folder in which the new gadget should be stored under project's **WebContent** node. If you do not specify a folder name, the new gadget will be stored directly under project's **WebContent** node.
Gadget Name	Specify a name for the new gadget. The gadget name: ■ Must have minimum 5 characters and maximum 50 characters. ■ Should not have any special characters. ■ Should be a single word.
Gadget Title	Specify the title to be displayed on the gadget.

Field / Control	Description
Preview Image	Click **Browse** and select an icon in .png or .jpg format for the gadget. The image size should not be more than 50KB, and the preferred size of the image is 70 X 70.
Settings Title	Specify a name for the gadget setting dialog.
Description	Provide a description for the new gadget.
Gadget ID	Specifies the auto-generated ID for the new gadget. This field is not editable.
Gadget Group Name	Select one of the options for the gadget group name: ■ **Use project name** if you want to use the project name as the gadget group name. ■ **Use custom name** if you want to enter the name for the gadget group in the input field. The group name provided here will be used to categorize gadgets in the Add New Gadget dialog in Business Console.
Finish	Click if you selected **Default with empty stubs** as the gadget type. A new gadget will be created under the project selected for the gadget.
Next	Click if you have to specify the AngularJS gadget settings. For details, see "AngularJS Gadget Specification" on page 12.

AngularJS Gadget Specification

The table below describes what you can specify when you create a new AngularJS gadget.

Element	Description

Hosts	For each remote server that you want to configure with the gadget , specify the following host details:
	■ **Name** Type the alias name of the server.
	■ **Host name** Type the host name of the server.
	■ **Port** Type the port used by the server.
	■ **Server type** Select one of the server types from the list.
	  ■ AgileApps Cloud (Uses SAML 2.0 authentication)
	  ■ Integration Server (Uses SAML 2.0 authentication)
	  ■ My webMethods Server (Uses SAML 2.0 authentication)
	  ■ Others (Does not use SAML 2.0 authentication)
	■ **Secure** Select this option if you want to use HTTPS protocol for secure communication.

JavaScript Event Bus	Select this option if you want to specify publish and subscribe events for the gadget. To define publish events 1. Click **Add** next to Publish. 2. Specify the publish event name. 3. (Optional) Provide the payload for the publish event in a text window. Specify a payload only if the event has to send a fixed payload each time the event is fired. To define subscribe events 1. Click **Add** next to Subscribe. 2. Specify the subscribe event name.

Element	Description
REST Invocation	■ Select the **Use Cross-origin resource sharing (CORS) support on host servers** option if you want the gadget to communicate with host servers that support CORS. ■ Select the **Use Business Console proxy service** option if you want the gadget to use Business Console's inbuilt Proxy REST service to communicate with the host servers.
Finish	Click this option to create a new gadget as specified in the wizard.

Gadget Configuration Files

You can find the folders and files that the New Business Console Gadget wizard creates in the following locations:

- In the Solutions view, the folders and files are in the User Interfaces/<*gadget_folder_if_provided*>/*gadget_name* folder.
- In the Navigator view or Package Explorer view, the folders and files are in the *web_or_portlet_application* /WebContent/<*gadget_folder_if_provided*>/*gadget_name* folder.

The following table describes the folders and files.

Folder	Files	Description
Images	*gadget_icon_image.png* or *gadget_icon_image.jpg*	This image file will be used as the icon for the new gadget.
Scripts	■ config.js	This file will contain the gadget configuration details. Do not edit this auto-generated file.
	■ controller.js	This file is created only if the gadget type is AngularJS. This file

Folder	Files	Description
		specifies how the gadget will work. You need to know JavaScript and AngularJS to edit this file and program the gadget.
	■ custom.js	Optionally, you can use this file to programmatically customize the gadget.
Styles	*style* .scss	This folder will contain the default style sheet for the gadget. You can add your own style sheets (.scss files) to this folder.
Views	settings.xhtml	This file will be empty initially. In this file, you can build a custom form to capture the values of the custom parameters specified when the gadget is configured in a Business Console AppSpace. The controller for an AngularJS gadget decides the business logic based on the values it receives from the custom form. For non AngularJS applications (for example JQuery), you can use selectors to directly capture the form values and then apply the business logic.

Folder	Files	Description
	view.xhtml	You can edit this file and specify how the gadget should be displayed on the web. For gadget of type AngularJS , each view is tied to its respective controller. Controller decides the content for display.

Gadget Definition File

In the Navigator view or Package Explorer view, the gadget definition file is listed in the *web_or_portlet_application* /WebContent/WEB-INF/gadgets/*gadgetName_gadgetID* folder.

Folder	Files	Description
gadgetName_gadgetID	gadgetDefinition.xml	This file will contain gadget details such as the location of the files pertaining to gadget styles, scripts, and run-time parameters. Use the gadget definition editor to edit the gadget definitions. See "Gadget Definition Editor" on page 23 .

Importing a Gradle or Apache Maven Based Web Project to Create Gadgets

You can import a Gradle or Apache Maven based web project into Designer to create gadgets. After importing, the Gradle or Apache Maven based web project must be modified prior to creating gadgets.

To import and modify a Gradle or Apache Maven based web project

1. In Designer, click **File > Import > General > Existing Projects into Workspace**, and then click **Next**.
2. In the Import dialog box, perform one of the following:

- In **Select root directory**, click **Browse** to choose the directory location of your Gradle or Apache Maven based web project.
- In **Select archive file**, click **Browse** to select a zip file of your Gradle or Apache Maven based web project.

3. Click **Finish**.

   The Project Explorer and Navigator views display the imported project.

4. To modify the project, perform the following:

   a. Right-click the project and click **Properties**.

      The Properties dialog box for the project appears.

   b. Click **Project Facets** and select **Convert to faceted form**.

   c. In the **Configuration** drop-down list, select **CAF Web Application**.

   d. Click the **Runtimes** tab and select the My webMethods Server runtime environment.

      If a runtime environment is not listed, click **New** to add the My webMethods Server runtime environment.

   e. Click **Further configuration required**.

   f. In the **Content directory** field, type the location of the WEB-INF parent folder. For example, src/main/webapp.

   g. Click **Next** and select CAF libraries in the **Type** drop-down list, and then click **OK**.

   h. In the Properties dialog box, click **Apply**, and then click **OK**.

   i. In the Project Explorer or Navigator view, right-click the project and click **CAF tools > Repair CAF Project**.

   You can now create gadgets in the project. For information about creating new gadgets, see "Creating a New Business Console Gadget" on page 8.

Creating Gadget User Interfaces for a REST Resource

You can use both REST and REST V2 resource descriptors to generate custom webMethods Business Console user interfaces, and add those user interfaces to new or existing gadgets.

The high-level steps for creating user interfaces based on REST API Descriptors are as follows:

1. Create a REST or REST V2 resource and generate a REST API descriptor for that resource, using the Package Navigation section on the UI Development perspective in Designer. For more information about REST resource, REST V2 resource, and REST API descriptors, see *webMethods Service Development Help*.

2. Create a CAF web application project, and add a Business Console gadget to that project.
3. Configure the input and output mapping of REST resources to the Business Console user interface elements, using the REST UI Creation Wizard in the UI Development perspective.
4. Configure the gadget. For more information, see *Developing Gadgets for webMethods Business Console*.
5. Deploy your project to My webMethods Server and display the gadget in an AppSpace in Business Console.

To create a user interface for a REST resource

1. Create a web application project.

   For more information about creating a web application project, see *webMethods CAF and OpenUI Development Help*.
2. In the Solutions view, right-click the web application project and select **Business Console Gadgets > New Business Console Gadget**.
3. Configure your gadget as described in "New Business Console Gadget Wizard" on page 10.
4. In the Solutions view, double-click to open the view.xhtml file for your gadget in the view editor.
5. Double-click the REST API descriptor and select the **REST Resources** tab, and ensure that the parameters for REST operations such as `PUT`, `GET`, and others are correctly configured.

   All parameters must be set as `QUERY` for the `GET` operation because it can send parameters only through a query string. Similarly, `POST` and `PUT` operation parameters must be either `QUERY` or `FORMDATA`.

   Note: If you select the **Get Operation on Load** option in the Pagination and Loading Inputs Configuration page of the REST UI Creation Wizard, a `GET` operation with same parameters precedes each `PUT` operation. In this case, ensure that `PUT` parameters are set to `QUERY` because the corresponding `GET` operation can use only `QUERY` parameters.
6. Drag and drop the required REST API Descriptor to the Design Page of the view editor.

   The REST UI Creation Wizard appears.
7. On the Project Selection page, click **Next**.

   Note: You can edit the **Swagger URL** field when you start the wizard using the **New > Other > Software AG > UI Development > New REST UI** menu. Specify a publicly available HTTP URL, or a local file path in the Swagger URL field.

8. On the REST Resources Selection page, specify the following and click **Next**:
   - The REST resources and operations for which you want to create user interface controls. You can select only one operation at a time on the REST Resources Selection page.
   - Whether to use Cross-origin resource sharing (CORS) support on host servers, or the Business Console proxy REST service as the REST invocation method.
   - Whether to use a global or local URL. To use a global URL, you must specify an alias, and configure the global server settings for gadgets in My webMethods Server. In case of a local URL, the URL in the Swagger metadata is used for accessing the REST resources.
   - Whether to use Integration Server or other external servers. If you use other external servers, select **Basic** in the **Select AuthType** box, and enter the authentication password and user name. Additionally, you can select **None** for no authentication.
9. On the Field Configuration page, configure the form in which the Composite Application Framework can generate user interface controls for the selected REST resources as follows:

Field/Table	Description
Form Name	The name of the form to display in Business Console.
Input and Output Parameters	For each user interface control in the table, configure the following fields: ■ **Name**. Select to specify whether to display the control in Business Console. ■ **Display Label**. Click to modify the text displayed on the field label in Business Console. ■ **Type**. You can set the type for String Arrays as either String Array or String Table. ■ **Mandatory** (only for input parameters). Select to specify whether a form control is required. The field appears as a mandatory field on the user interface. ■ **Validation** (only for output parameters). The validation type to apply to the control. This value is populated automatically. The options are Text, E-mail, Alphanumeric, and Numeric. ■ **UI Control**. By default, the framework generates an appropriate user interface control type for

Field/Table	Description
	each REST parameter. For Ref type parameters, you can select the **Tab-Selected** option to display the parameter as a selected tab on the form. If you select the **Tab-Selected** option for a Ref type parameter, other Ref type parameters at the same level on the Field Configuration page are grouped and displayed as tabs on the form, but they are not selected by default. ■ **Default Value**. Initial value of Text and Numeric fields, String Array, and Numeric Array when the page is loaded. These fields are only for Table and Array type values: ■ **SortBy**. Parameter name to use for sorting (if REST service supports sorting). ■ **SortOrder**. Parameter name to use for sorting order (if REST service supports sorting order). ■ **DefaultSortColumn**. Select the default column to use for sorting. ■ **DeafultSortOrder**. Select either **Ascending** or **Descending** in the list.
Select Column Count for Input and Output Parameters	Select the number of columns in which Business Console displays the generated form. The options are: One, Two, Three, Four, and Six.

10. On the Pagination and Loading Inputs Configuration page, configure the pagination options.

    Select the parameter you want to use for each of the available pagination options.

Field	Description
Name	Select the parameter for which to apply pagination.
Type	You cannot modify the data type of the control.
Page Number	Select the parameter for number of pages to display in the Business Console gadget.

Field	Description
Page Size	Select the parameter for number of entries per page.
Start Index	Select the parameter for start index.
End Index	Select the parameter for end index.
Total Records	Total number of records to display.

11. If you have selected the `PUT` operation in the REST Resources Selection page, select how to load the input parameters. You can select one of the following:

Field	Description
None	No action is performed prior to the `PUT` operation.
Gadget Configuration Parameter	Specify one or more gadget configuration parameters and map them to input parameters.
Gadget Event	Specify one or more gadget events and map their response to input parameters.
Get Operation on Load	A `GET` operation with same parameters as the `PUT` operation precedes the `PUT` operation.

12. (Optional) Click **Browse** and save all settings in the REST UI Creation Wizard as a JSON file. This action enables you to load the wizard with the same settings later. To load the wizard using a JSON file, do the following:
    a. Go to **New > Other > Software AG > UI Development > New REST UI**, and click **Next**.
    b. Select **Import metadata**, and then click **Browse** to select the JSON file.
    c. Click **Next**.
13. Click **Finish**.

After Designer generates the user interface, you can configure it further, deploy it to My webMethods Server, and use it in Business Console. For more information about gadget development, see *Developing Gadgets for webMethods Business Console*. For more information about working with gadgets at runtime, see *Working with webMethods Business Console*.

Creating a webMethods AgileApps Cloud Form Gadget

You can import an AgileApps Cloud form in Designer and create a gadget based on the form.

To create an AgileApps Cloud form gadget

1. Create a portlet or web application project.

   For more information about creating a portlet or web application project, see *webMethods CAF and OpenUI Development Help*.

2. In the Solutions view, right-click the portlet or web application project and select **Business Console Gadgets > Generate AgileApps Form Gadget**.

3. Connect to AgileApps Cloud by specifying values for the following fields:

Field	Description
AgileApps Host	The name of the AgileApps Cloud server.
User ID	The user ID to log into the AgileApps Cloud server host.
Password	The password for the AgileApps Cloud user account.

4. Click **Next**.

5. Select an AgileApps Cloud form to import by specifying values for the following fields:

Field	Description
Application	Select an AgileApps Cloud application.
Objects	Select an object associated with the application.
Select a Form to import	Select one or more forms associated with the object for which you want to create gadgets.

6. Click **Finish**.

Designer creates one or more gadgets that you can configure further. For more information about programming gadgets, see *Developing Gadgets for webMethods Business Console*.

Previewing a Gadget in Designer

Prior to previewing a gadget, you must specify My webMethods Server as the runtime application server.

To preview a gadget

1. In the Solutions view, right-click the gadget.
2. Select **Run on Server** to publish the gadget.

   After publishing, the preview appears in your web browser. You can use the URL to view the gadget directly in the web browser at a later time. For more information about viewing a gadget directly in a web browser, see "Viewing a Gadget in a Web Browser" on page 22.

   If you have not specified My webMethods Server as the runtime application server, the Run on Server wizard appears. You can specify My webMethods Server as the runtime application server in the wizard.

Viewing a Gadget in a Web Browser

You can directly view a gadget in a web browser.

To view a gadget in a web browser

1. Specify the URL of the gadget in the following format:

   `http://host:port/business.console.gadgets#/projectName/gadgetName`

   where:

   - *host* is the host name of My webMethods Server.
   - *port* is the port number used by My webMethods Server.
   - *projectName* is the name of the web or portlet application project in which the gadget is defined.
   - *gadgetName* is the name of the gadget.

   You can also check the format in the **Gadget URL** field of the gadget definition editor. For more information about the gadget definition editor, see "Gadget Definition Editor" on page 23.
2. View the gadget in your web browser.

Gadget Definition Editor

You can open the gadget definition editor in one of the following ways:

- In the Solutions view, double-click a gadget.
- In the Navigator view or Package Explorer view, double-click the gadgetDefinition.xml file corresponding to a gadget in the *web_or_portlet_application* / WebContent/WEB-INF/gadgets/*gadgetName_gadgetID* folder.

The table below describes the various sections in the **Gadget Definition Editor** tab. For a description of gadget elements, see "New Business Console Gadget Wizard" on page 10.

Section	Description
General	Expand to edit the gadget properties such as gadget title, description, settings title, and group name.
Scripts and styles	Expand to provide the scripts (.js files) and styles (.scss) files to be used by the gadget. If you want to use external scripts such as the scripts delivered through a content delivery network (CDN), save the script file under / WebContent/<*gadget_folder_if_provided*>/*gadget_name* / Scripts folder, and then provide a relative link to the script file here. The same applies when you use an external style file. Do not import a script file directly in the view.xhtml file.
Parameters	Expand to provide the name and value for the custom properties of the gadget. It is optional to provide values for the properties. The values can be provided at runtime by using the settings.xhtml file. Parameter name must be unique within a gadget.
Hosts	Expand to provide the details for each server that you want to configure with the gadget. Host name must be unique within a gadget.

Section	Description
Publish events	Expand to provide the publish events for the gadget. Optionally, you can provide the payload for the publish event. This is applicable only to gadgets of type AngularJS.
Subscribe events	Expand to provide the subscribe events for the gadget. This is applicable only to gadgets of type AngularJS.
Dependencies	Expand to provide the dependent modules to be injected to the gadget framework. This is applicable only for AngularJS based gadgets. Independent AngularJS module(s) created in a gadget can be shared with another gadget. Make sure you add the dependencies only after the dependent modules are created in gadgets and deployed to My webMethods Server. Dependency name must be unique within a gadget.