

TIBCO ActiveMatrix® BPM Web Client Developer's Guide

*Software Release 4.2
August 2017*

Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, Two-Second Advantage, TIBCO ActiveMatrix BPM, TIBCO Administrator, TIBCO Business Studio, TIBCO Enterprise Message Service, TIBCO General Interface, TIBCO Hawk, TIBCO iProcess, TIBCO JasperReports, TIBCO Spotfire, TIBCO Spotfire Server, and TIBCO Spotfire Web Player are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

Enterprise Java Beans (EJB), Java Platform Enterprise Edition (Java EE), Java 2 Platform Enterprise Edition (J2EE), and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation in the U.S. and other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT 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.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 2005-2017 TIBCO Software Inc. All rights reserved.

TIBCO Software Inc. Confidential Information

Contents

TIBCO Documentation and Support Services	6
TIBCO ActiveMatrix BPM Web Client API	8
TIBCO ActiveMatrix BPM Web Client API Components and Services	8
Web Client API jar Files	9
Authenticating Access to a TIBCO ActiveMatrix BPM Web Client Service	9
Accessing Ad-hoc Activities in a Custom Client	11
Web Application Development	12
Creating a New Web Application Project	12
Enabling Openspace Gadget Contribution	13
Creating an Openspace Gadget	14
About Gadget Development	15
Enabling Managed Events	15
Compiling Your Web Application Project	18
Exporting and Deploying Your Openspace Gadget	18
Viewing the Gadget Specification in a Browser	19
Contributing Your Openspace Gadget	19
Configure the Gadget Hub Policy	21
Web Client API Service	22
Authentication	22
GetAuthenticatedGuid	22
GetAuthenticatedUserName	22
IsActionAuthorised	22
Business Services	23
GetBusinessServices	23
ListBusinessServiceCategories	24
StartBusinessService	24
UpdateBusinessService	24
Calendar Services	25
CopyCalendar	25
DeleteCalendar	25
DeleteCalendarEntries	26
GetCalendarEntries	26
GetCalendarReferences	27
GetWorkCalendar	27
ListCalendars	28
ListTimeZones	28

PurgeCalendarEntries	28
RemoveCalendarReference	29
RenameCalendar	29
ResolveReferences	30
SaveCalendar	30
SaveCalendarEntry	31
SaveCalendarReference	32
SetCalendarReferences	32
Dataview Client Service	33
CreateDataView	33
EditDataView	33
Login Services	34
LoginAction	34
Logout Services	35
LogOutAction	35
Organization Model Services	35
GetEntityName	35
GetResourceDetails	36
OpenOrgModel	36
RunResourceQuery	37
Process Template Services	37
CancelProcessInstance	37
GetEventDurationsFromEC	38
GetMigrationPoints	38
GetProcessInstanceData	39
GetProcessInstancesFromEC	39
GetProcessInstanceStatus	40
GetProcessTemplateList	40
IgnoreHaltedProcessInstance	41
ListMigrationRules	41
QueryEventCollector	41
QueryHaltedProcessInstances	42
QueryProcessInstances	42
QueryProcessTemplates	43
ResumeHaltedProcessInstance	43
ResumeProcessInstance	44
RetryHaltedProcessInstance	44
SetMigrationRules	45
StartProcess	45

SuspendProcessInstance	46
Session	46
InvalidateSession	46
SetSessionMaxInactiveInterval	47
Work Item Services	47
AllocateWorkItem	47
CancelWorkItem	48
CloseWorkItem	49
CloseWorkItems	50
CompleteWorkItem	50
GetOrderandFilterForResource	51
GetOrgEntityConfigAttribute	52
GetOfferSet	52
GetToDoList	52
OpenNextWorkItem	53
OpenWorkItem	54
PendWorkItemByDate	55
PendWorkItemByPeriod	55
PrioritiseWorkItem	56
ReAllocateWorkItem	57
SetOrderAndFilterForResource	58
SetOrgEntityConfigAttribute	58
ReofferWorkItem	58
SkipWorkItem	59
Work Views Services	59
CreateWorkListView	60
DeleteResourceFromView	61
DeleteWorkListView	62
EditWorkListView	62
GetEditableWorkViews	63
GetPublicWorkViews	64
GetWorkItemsForView	64
GetWorkViewDetails	65
GetWorkViewsForResource	65
UnlockWorkListView	66
Using Wild Card Characters	66
Constraints for Contributing Gadgets in Openspace	67

TIBCO Documentation and Support Services

Documentation for this and other TIBCO products is available on the TIBCO Documentation site. This site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, visit:

<https://docs.tibco.com>

Product-Specific Documentation

Documentation for TIBCO products is not bundled with the software. Instead, it is available on the TIBCO Documentation site. To directly access documentation for this product, double-click the following file:

`TIBCO_HOME/release_notes/TIB_amx-bpm_version_docinfo.html`

where *TIBCO_HOME* is the top-level directory in which TIBCO products are installed. On Windows, the default *TIBCO_HOME* is `C:\tibco`. On UNIX systems, the default *TIBCO_HOME* is `/opt/tibco`.

The following documents for this product can be found on the TIBCO Documentation site:

- TIBCO ActiveMatrix BPM SOA Concepts
- TIBCO ActiveMatrix BPM Concepts
- TIBCO ActiveMatrix BPM Developer's Guide
- TIBCO ActiveMatrix BPM Web Client Developer's Guide
- TIBCO ActiveMatrix BPM Tutorials
- TIBCO ActiveMatrix BPM Business Data Services Developer Guide
- TIBCO ActiveMatrix BPM Case Data User Guide
- TIBCO ActiveMatrix BPM Event Collector Schema Reference
- TIBCO ActiveMatrix BPM - Integration with Content Management Systems
- TIBCO ActiveMatrix BPM SOA Composite Development
- TIBCO ActiveMatrix BPM Java Component Development
- TIBCO ActiveMatrix BPM Mediation Component Development
- TIBCO ActiveMatrix BPM Mediation API Reference
- TIBCO ActiveMatrix BPM WebApp Component Development
- TIBCO ActiveMatrix BPM Administration
- TIBCO ActiveMatrix BPM Performance Tuning Guide
- TIBCO ActiveMatrix BPM SOA Administration
- TIBCO ActiveMatrix BPM SOA Administration Tutorials
- TIBCO ActiveMatrix BPM SOA Development Tutorials
- TIBCO ActiveMatrix BPM Client Application Management Guide
- TIBCO ActiveMatrix BPM Client Application Developer's Guide
- TIBCO Openspace User's Guide
- TIBCO Openspace Customization Guide
- TIBCO ActiveMatrix BPM Organization Browser User's Guide (Openspace)
- TIBCO ActiveMatrix BPM Organization Browser User's Guide (Workspace)

- TIBCO ActiveMatrix BPM Spotfire Visualizations
- TIBCO Workspace User's Guide
- TIBCO Workspace Configuration and Customization
- TIBCO Workspace Components Developer Guide
- TIBCO ActiveMatrix BPM Troubleshooting Guide
- TIBCO ActiveMatrix BPM Deployment
- TIBCO ActiveMatrix BPM Hawk Plug-in User's Guide
- TIBCO ActiveMatrix BPM Installation: Developer Server
- TIBCO ActiveMatrix BPM Installation and Configuration
- TIBCO ActiveMatrix BPM Log Viewer
- TIBCO ActiveMatrix BPM Single Sign-On
- Using TIBCO JasperReports for ActiveMatrix BPM

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, contact TIBCO Support:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

<http://www.tibco.com/services/support>

- If you already have a valid maintenance or support contract, visit this site:

<https://support.tibco.com>

Entry to this site requires a user name and password. If you do not have a user name, you can request one.

How to Join TIBCO Community

TIBCO Community is an online destination for TIBCO customers, partners, and resident experts. It is a place to share and access the collective experience of the TIBCO community. TIBCO Community offers forums, blogs, and access to a variety of resources. To register, go to the following web address:

<https://community.tibco.com>

TIBCO ActiveMatrix BPM Web Client API

You can develop custom clients in TIBCO Business Studio using Google Web Toolkit (GWT) that, in turn, use the Web Client API to access BPM runtime services hosted in a TIBCO ActiveMatrix container.

The Web Client API consists of model objects that can be used by your custom client. The model objects are simple POJOs that are serializable. The Web Client API uses the GWT Framework to achieve the serializing and deserializing of the model objects.

When your custom client calls an operation in the Web Client API, the GWT Framework takes the information from the Java call and:

1. Converts it to a Java object.
2. Serializes the object.
3. Sends the Java object to the appropriate service in the BPM runtime.
4. Deserializes the Java object for the BPM runtime.

When the BPM runtime returns the Java object, the GWT Framework serializes the Java object and then converts it back to a Java object.

TIBCO ActiveMatrix BPM Web Client API Components and Services

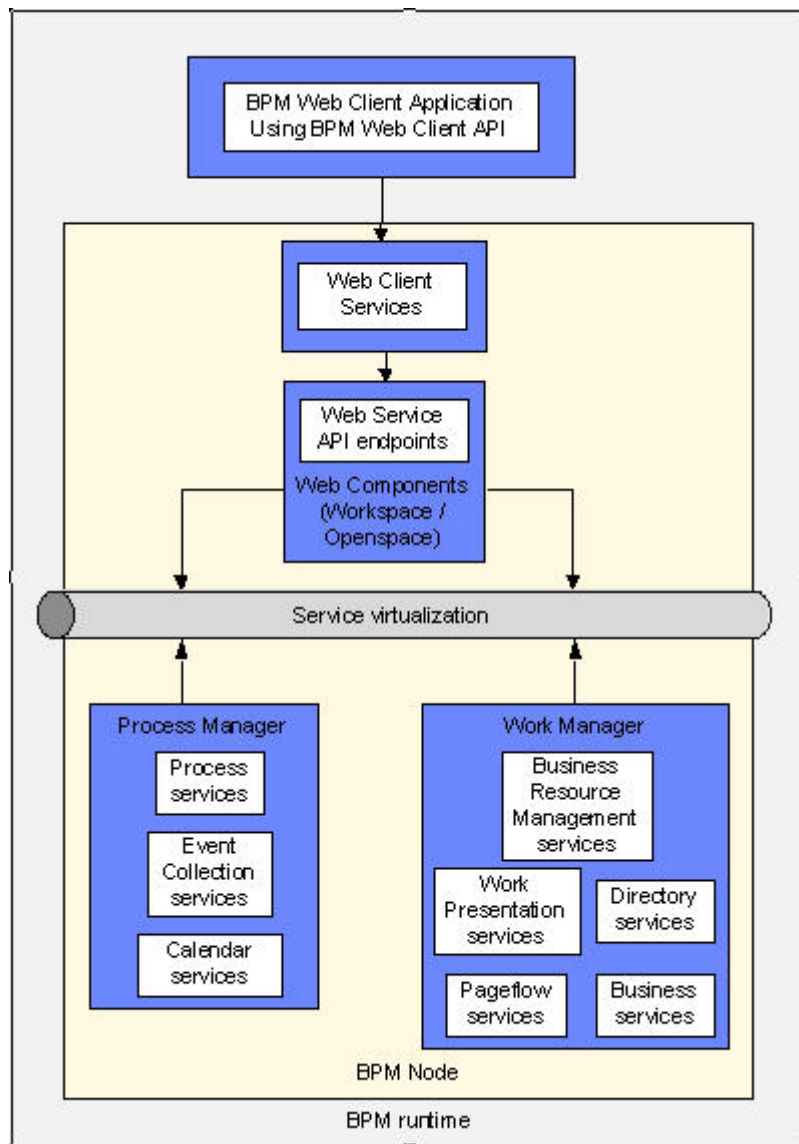
The Web Client API enables you to develop custom web clients that can access the BPM runtime services that are hosted in a TIBCO ActiveMatrix node. This is achieved by developing custom web clients in TIBCO Business Studio using Google Web Toolkit (GWT) that in turn uses the Web Client API to access the BPM runtime services.

TIBCO ActiveMatrix BPM is implemented as a number of independent components that expose services, exploiting the benefits of the Service Component Architecture (SCA) provided by the TIBCO ActiveMatrix BPM runtime.

These components are grouped into a set of logical nodes, of different types. (A logical node is a heterogeneous group of application fragments that must be deployed to the same physical node.)

The figure below provides an overview of this architecture, illustrating:

- the main interactions between logical nodes.
- the main external services provided by TIBCO ActiveMatrix BPM.



Web Client API jar Files

The Web Client API is provided as two jar files which are installed at the location where you installed TIBCO ActiveMatrix BPM. For example, C:\Program Files (x86)\tibco\amx-bpm\client-api\3.0\web-client-api.

The jar files are:

- bpm-web-client-api-1.4.jar
- bpm-web-client-dep-1.4.jar

The Web Client API is available to your development environment, when you create a new web application in TIBCO Business Studio. See [Creating a New Web Application Project](#).

Authenticating Access to a TIBCO ActiveMatrix BPM Web Client Service

At runtime, security policies are enforced on the endpoint of every TIBCO ActiveMatrix BPM web client service to ensure that access is restricted to authenticated users. Every API call to a TIBCO ActiveMatrix BPM service must be made using the identity of a user who is registered in the BPM organization model. An API call that does not meet this requirement will be rejected.

To execute a web client service a user must have a valid session. If there is no valid session, a `SessionTimedOutException` is thrown. The web client service checks whether or not this exception has been thrown and if it has been thrown, a `SessionInvalidEvent` is fired. This means that before executing any web client services, a user must subscribe to the `SessionInvalidEvent` event.

Procedure

1. A user from the BPM organization model must be logged in using the `LoginAction`. See [Login Services](#).
2. Once the user has logged in, they must subscribe to the `SessionInvalidEvent`.

Accessing Ad-hoc Activities in a Custom Client

TIBCO have created custom applications for ad-hoc activities that you can use when creating your own custom client.

Both the ad-hoc custom applications can be accessed directly and can be displayed anywhere on a page (provided a valid server session exists with TIBCO ActiveMatrix BPM). See [Authenticating Access to a TIBCO ActiveMatrix BPM Web Client Service](#).

There are two custom applications:

- `toolbar.html`. This application should be used in the context of an open work item form. It only displays enabled manual ad-hoc activities. It does not show disabled or automatic activities as they are not relevant in the context of an open form. It is designed to allow a user to quickly start or cancel an ad-hoc activity while continuing to work on the open work item form.
- `app.html`. The `app.html` application should be used if you want find out about the ad-hoc activities for a given work item, process instance or case reference. You can toggle off the automatic and disabled ad-hoc activities by selecting either **Show automatic** or **Show disabled**.

Note that:

- Each application can use the following parameters:
 - (Required) You must specify a locale parameter. The parameter is `locale=locale` where *locale* is a string representation of a locale that includes a language and, optionally, a country code in the format, `ll_CC` as specified in the ISO 639-2 standard.
 - (Optional) `workitemid=id`. Displays ad-hoc activities for the given work item ID. You can use this method if the process id is not known. This method makes another server call to retrieve the process details for the work item. For example, `http://localhost:8080/openspace/adhoc/app.html?workitemid=123&locale=en_US`
 - (Optional) `processid=id&processname=procname`. Displays ad-hoc activities for the given process id and process name. For example, `http://localhost:8080/openspace/adhoc/app.html?processid=pvm:0a123&processname=myprocess&locale=en_US`
 - (Optional) `globaldataref=ref`. Displays ad-hoc activities for the given case reference. For example, `http://localhost:8080/openspace/adhoc/app.html?globaldataref=BDS-3-com.example.claimbom3.Policy-1-0&locale=en_US`
- For each ad-hoc activity, the name of the activity displays. You can either start or cancel an ad-hoc activity, depending on its current state.
- You can also search for an activity using a text based search. It searches all the text that is currently displayed. The search is dynamic so as you start typing, the ad-hoc activities that match the text you are typing automatically display.
- You must be logged in as a user that has privileges with the `startAndCancelAdHocActivity` system action assigned to view the ad-hoc activities. There may also be privileges applied to each ad-hoc activity which are required to start an instance.
- The process instance must be started.

The URL for the custom applications is as follows: `http://host:port/openspace/adhoc/applicationname` where:

- *host* is the machine name or IP address of the server hosting TIBCO ActiveMatrix BPM.
- *port* is the port number that TIBCO ActiveMatrix BPM is using.
- *applicationname* is either `app.html` or `toolbar.html`, depending on your requirements.

Web Application Development

Using TIBCO Business Studio, you can develop web clients to be used with TIBCO ActiveMatrix BPM using the Web Client API. TIBCO Business Studio uses Google Web Toolkit (GWT) that in turn uses the Web Client API to access the BPM runtime services.



Select the **Openspace Gadget Development** installation profile when installing TIBCO Business Studio if you want this feature to be available in TIBCO Business Studio. The **Openspace Gadget Development** installation profile is available when you select **Customize Installation** during installation.



The GWT Framework provided by TIBCO Business Studio only supports standard GWT projects. GWT App Engine is not supported. See <http://opensocial.org> for more information.

Procedure

1. You must have selected the **Openspace Gadget Development** installation profile when installing TIBCO Business Studio if you want this feature to be available in TIBCO Business Studio. The **Openspace Gadget Development** installation profile is available when you select **Customize Installation** during installation. See *TIBCO Business Studio Installation Guide*.
2. In TIBCO Business Studio, create a new web client application project from the GWT menu. See [Creating a New Web Application Project](#). This creates the environment you need to develop your web client application. This includes:
 - The Eclipse Java IDE project provides the following functionality:
 - Source files containing the classes and methods to develop the basic client application.
 - Gadget XML that launches the application as a gadget in a gadget container.
 - Other resources such as CSS files, images, etc.
 - Libraries required to build and run the client application.
 - The TIBCO Active Matrix BPM Web Client API.
 - The Google plug-in for Eclipse. This must be GWT Version 2.5.1.
3. Develop your gadget, see [About Gadget Development](#).
4. Compile your gadget, see [Compiling Your Web Application Project](#).
5. Deploy your gadget to TIBCO ActiveMatrix BPM, see [Exporting and Deploying Your Openspace Gadget](#).
6. Contribute your gadget to Openspace, see [Contributing Your Openspace Gadget](#).

Creating a New Web Application Project

You must create a web application project in TIBCO Business Studio to provide the features you need to develop and contribute your GWT gadget to Openspace.



- The GWT Framework provided by TIBCO Business Studio only supports standard GWT projects. GWT App Engine is not supported. See <http://code.google.com/apis/opensocial/> for more information.
- GWT uses port 8888 for hosted mode testing and this conflicts with a port that is used internally by TIBCO Business Studio. Therefore, TIBCO recommends that you change the port used in GWT for hosted mode testing. Please see the documentation supplied by GWT for information on how to do this.

Procedure

1. From TIBCO Business Studio, click  .
2. Click **New Web Application Project ...**
3. Complete the **New Web Application Project** dialog as follows:

Field	Description
Project Name	Enter a descriptive name for the project.
Location	Either accept the default location for the project (your workspace) or de-select the Use default location check box and click Browse to select a different location.
Google SDKs	You must use GWT 2.5.1. Accept the default Google SDK if it is GWT 2.5.1 or select Use specific SDK and select GWT 2.5.1 from the drop-down list. If GWT 2.5.1 is not listed, click Configure SDKs to install and configure it.
Google App Engine	De-select.
Google Apps Market Place	De-select.
Sample Code	De-select.

4. Click **Finish** on this dialog to create a project with the settings you have made to that point.

Enabling Openspace Gadget Contribution

Once you have created your web application project, you must enable it. Enabling Openspace contribution automatically provides the libraries to develop an Openspace-compatible gadget.

Procedure

- You can do this in two ways.
 - select your project, click  and select **Enable Openspace Gadget Development**, or
 - right-click the project, select **Openspace Gadget Development > Enable Openspace Gadget Development**.

Creating an Openspace Gadget

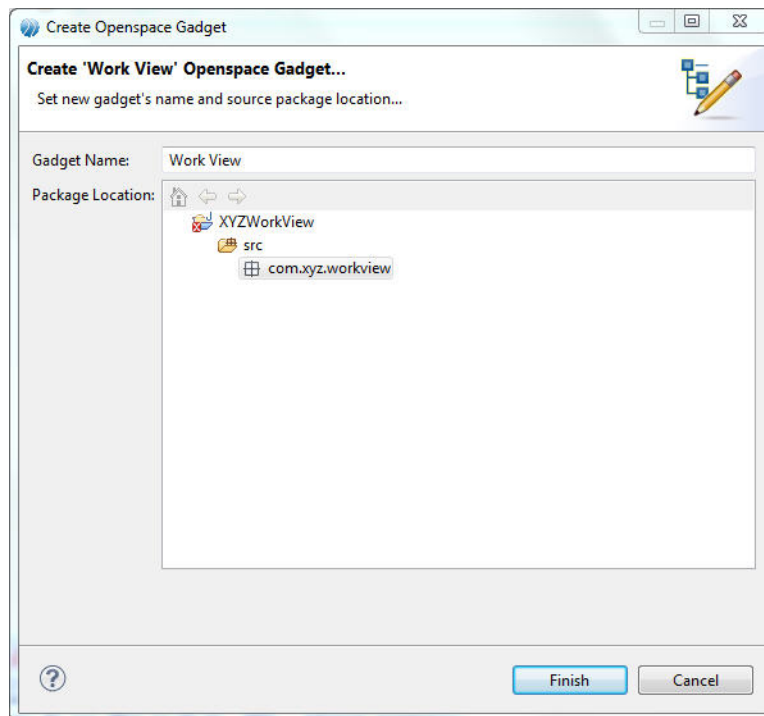
TIBCO have provided a sample gadget the you can use to model all your Openspace gadgets. You can modify and extend this gadget to produce your own gadget.



- This example and all the packaged JARs are only for internal use and should not be publicly distributed.
- The sample gadget example is supplied "as is" with no warranties. The code is intended as a simple illustration of the concepts and techniques needed to develop a custom gadget application. **It is not intended as a basis for production-ready code and should not be used as such.**
- Any references to any third party software in the code is not under TIBCO's control and we can offer no warranties.
- This sample gadget is only a demonstration of contributing a gadget to openspace. This gadget is developed invoking the BPM Web Client Services.

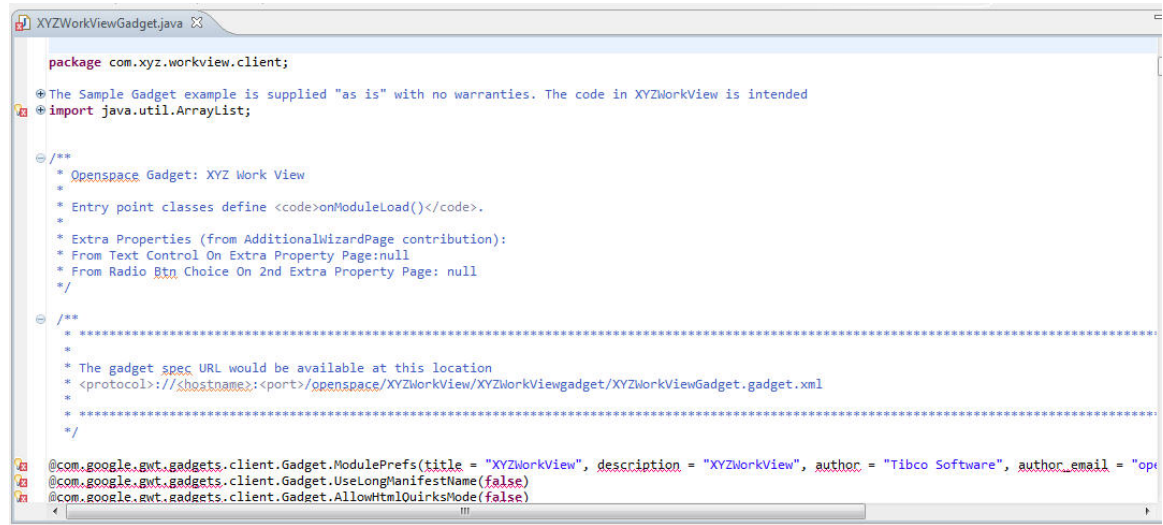
Procedure

1. Right-click your project and select **Openspace Gadget Development > Create Work View Openspace Gadget...**
2. From the **Create Openspace Gadget** dialog, do the following:
 - a) In **Gadget Name**, type the name of your gadget.
 - b) In **Package Location**, browse to the location of the package that you specified when you created your web application project.



3. Click **Finish**. The sample gadget is created.
4. To start developing the gadget, browse to `projectname > src > packagename.client > gadgetnameGadget.java` where:
 - `projectname` is the name of your project.
 - `packagename` is the name of the package you specified when you created your web application project.

- `gadgetnameGadget` is the name of the gadget you specified above, with the spaces removed.
5. Double-click `gadgetname.java` to open it. It should look similar to the one below:



About Gadget Development

When developing a gadget that uses the Web Client API, it must run in the context of the Openspace session, call the action (passing any parameters that are required) and, if required, support managed events.

- At runtime, the deployed gadget runs in the context of the current Openspace session and hence no separate login is required to invoke the BPM services. However, you must provide the service URL of the BPM node where you want to call the service. However, you only need to provide this once, as the service URL can be used again.
- Call the action, passing any parameters that are required. For a complete list of actions and parameters, see [Web Client API Service](#).
- If you want your gadget to support the publishing of managed events, you must enable managed events, see [Enabling Managed Events](#).

For a summary of all the public services provided by TIBCO Openspace, and links to detailed documentation generated from the individual APIs, see [Web Client API Service](#).

Enabling Managed Events

Openspace gadgets provide support for publishing managed events. If you want your gadget to communicate with other Openspace gadgets, you must enable support for managed events in your gadget.



For more information about managed events see "Administering Managed Events" in *TIBCO Openspace User's Guide*.

Procedure

1. Add the following elements to the project module XML file (`ProjectName.gwt.xml`):

```

<inherits name="com.google.gwt.json.JSON"/>
<inherits name="com.google.gwt.xml.XML" />
<inherits name="org.jsonmaker.gwt.Gwt_jsonmaker" />
<inherits name="com.tibco.bpm.web.client.model.Model" />

```

2. Ensure the generator is added to the `gwt.xml`

```

<generate-with
class="com.tibco.bpm.web.client.model.pagebus.rebind.ManagedEventGenerator">

```

```
<when-type-assignable
class="com.tibco.bpm.web.client.model.pagebus.hub.ManagedEvent"/>
</generate-with>
```

3. Create required `ManagedEvent` and `ManagedEventHandler` classes. See the following example classes:
 - `com.tibco.bpm.web.client.model.managedevents.TemplateManagedEvent`
`com.tibco.bpm.web.client.model.managedevents.TemplateManagedEventHandler`
4. Ensure the `PageBus.js` is injected using the `ScriptInjector`. Depending on how you are using GWT, the `<script>` tag may not be supported in the module `gwt.xml` file. You may receive an error similar to the one below when compiling:

The Cross-Site-Iframe linker does not support `<script>` tags in the `gwt.xml` files. For your application to run correctly, you must include these tags in your host page directly.

To do this, remove `<script src="PageBus.js" />` from the project module XML file and add the `PageBus.js` file using another method. The `ScriptInjector` sample code below shows an example of how this can be done using the `com.google.gwt.core.client.ScriptInjector` class:

```
import com.google.gwt.core.client.ScriptInjector;
import com.google.gwt.core.client.ScriptInjector.FromUrl;
protected static final String
PAGEBUS_JS_LOCATION="http://sampleHost:9090/samplepath/PageBus.js";
*
* =====
* METHOD : loadPageBus
* =====
*/
/**
 * The PageBus.js file must be loaded and injected because it cannot be
 * loaded via the module gwt.xml (ProjectName.gwt.xml) script tag:
 * <p/>
 * <code>
 * <script src="PageBus.js" />
 * </code>
 * <p/>
 * The Cross-Site-Iframe linker does not support <script> tags in the
 * gwt.xml files.
 */
private void loadPageBus()
{
    FromUrl fromUrl = ScriptInjector.fromUrl(PAGEBUS_JS_LOCATION);
    fromUrl.setCallback(new Callback<Void, Exception>()
    {
        public void onFailure(Exception reason)
        {
            Window.alert("Script load failed for: " + PAGEBUS_JS_LOCATION);
        }
        public void onSuccess(Void result)
        {
            buildLayout();
            // Verify that PageBus.js is injected and create the ManagedEventClient.
            if (isPageBusInjected())
            {
                sampleGadgetManagedEventClient = new SampleGadgetManagedEventClient("test");
            }
            else
            {
                Window.alert("ManagedEventClient cannot load because PageBus.js is not
                injected.");
            }
        }
    });
    // If setWindow is not called with the correct $wnd returned from the
    // getReferencedWindow() JSNI method below then the PageBus code is
    // not injected on the correct window and the subsequent
    // ManagedEventClient calls will fail with $wnd.OpenAjax is undefined.
    fromUrl.setWindow(getReferencedWindow());
    fromUrl.inject();
}
```

```

}
/*
 * =====
 * METHOD : getReferencedWindow
 * =====
 */
/**
 * Returns the correctly referenced nested frame window that the GWT compiled
 * script runs in. See the GWT document at:
 * <p/>
 * https://developers.google.com/web-toolkit/doc/latest/
 * DevGuideCodingBasicsJSNI
 * <p/>
 * When accessing the browser's window and document objects from JSNI, you must
 * reference them as $wnd and $doc, respectively. Your compiled script runs in a
 * nested frame, and $wnd and $doc are automatically initialized to correctly
 * refer
 * to the host page's window and document.
 *
 * @return The correctly referenced nested frame window.
 */
public native JavaScriptObject getReferencedWindow()
/*- {
return $wnd;
}- */;
/*
 * =====
 * METHOD : isPageBusInjected
 * =====
 */
/**
 * Test that the PageBus.js file has been loaded by verifying that
 * the wnd.OpenAjax property is initialized.
 * @return true if $wnd.OpenAjax is not undefined.
 */
public native boolean isPageBusInjected()
/*- {
return $wnd.OpenAjax != undefined;
}- */;

```

5. Create a HubPolicy.xml file that includes Publish and Subscribe elements matching the event topics used by the application.

The HubPolicy.xml file is deployed with the application so that the URL to this file can be specified when the gadget is contributed in Openspace. For example, include the HubPolicy.xml in the "public" folder in your project module root path.

The URI value must be JavaScript encoded as shown below:

```

var origUri = "http://<AMXBPMHost>:8080/gadgetserver/gadgets/ifr?
view=home&url=http://<gadgetxml_host>:<gadgetxml_port>/sample/testgadget/
TestGadget.gadget.xml";
var encodedUri = encodeURIComponent(origUri);

```

Use the value of the encodedUri. Please refer to the HubPolicy.xml file provided in the sample.

The base URI the Iframe loads must be an exact string match to the value used to set the HubPolicy domain URI (before it is encoded). The query_string may be included in the HubPolicy URI but does not need to match exactly.

The base URI is the first part of the URI up to the question mark character. The question mark character is a separator between the base URI and the query string. For example:

```
http://example_path/?query_string
```

where *example_path* might consist of host, port, program and so on.

An example of a base URI is shown below.

```
http://amxbpmhost:8080/gadgetserver/gadgets/ifr
```

The base URI must be an exact string match to the URI set in the HubPolicy (before it is encoded). For example, if *amxbpmhost* is at IP address: 92.223.243.87, both of the following resolve to the same IP address:

- `http://92.223.243.87:8080/`
- `http://amxbpmhost:8080/`

but the string values do not match. If the `HubPolicy.xml` file domain url is set using `http://92.223.243.87:8080/` but the actual IFrame is loaded from `http://amxbpmhost:8080/` these do not match and the gadget fails to load correctly.

Compiling Your Web Application Project

Once you have finished developing your web application project, you must compile it. You must compile your project before you export it to a Distributed Application Archive (DAA). The project is not automatically compiled when you export it as a DAA.

Procedure

1. From TIBCO Business Studio, click  . The Web Application menu displays.
2. Click **GWT Compile Project**.
3. Complete the **GWT Compile** dialog as follows:

Field	Description
Project	Click Browse to browse to the project you want to compile.
Compiler & Shell	• Select the log level you require from the drop-down list. • Select the output style you prefer from the drop-down list.
Entry Point Modules	Click Add or Remove to configure the entry point modules you require.
Advanced	Click the down arrow to display the advanced options: • Enter any additional compiler arguments in the Additional Compiler Arguments box. • Enter any VM arguments in the VM Arguments box.

4. Click **Compile**. You can view the progress messages in the **Console** tab.

Exporting and Deploying Your Openspace Gadget

To deploy your Openspace Gadget, you must first export your project to a Distributed Application Archive.



You must compile your project before you export it to a DAA. The project is not automatically compiled when you export it as a DAA.

Procedure

1. Select your project and drag and drop it on the server you want to deploy the gadget to. The **Select Openspace Gadget Project** dialog displays.

2. Make sure the project you want to deploy is selected and click **Finish**.

Viewing the Gadget Specification in a Browser

To view the gadget specification in a browser, type in the URL of the sample gadget. This is a way you can ensure that you can access the XML.

Procedure

- Type the URL of the sample gadget in your browser. For example, `protocol://amxbpmhost:8080/openspace/projectname/gadgetfoldername/MyGadget.gadget.xml` where:
 - *protocol* is the communications protocol being used by openspace, either HTTP or HTTPS. This was determined at installation.
 - *amxbpmhost* is the machine name or IP address of the machine where TIBCO ActiveMatrix BPM is installed.
 - *projectname* is the name of your web application project. The web application project name must be in lowercase.
 - *gadgetfoldername* is the name of your gadget folder.
 - *MyGadget.gadget.xml* is the name you have given to your Openspace Gadget.

Caching and Gadgets

The gadget server by default caches the gadget XML. If you change it and subsequently deploy it, your changes may not be reflected. To workaroud this, you can either:



- Open the following in a new tab `http://amxbpmhost:8080/gadgetserver/gadgets/ifr?view=home&url=http://gadgetxml_host:<gadgetxml_port>/MyGadget.gadget.xml&nocache=1&bpc=1` where *gadgetxml_host* is the location where your gadget XML is hosted. This notifies the gadget server not to cache the gadget XML. Once you have done this, reload the gadget into Openspace, or
- You can rename the gadget XML before redeploying.

Result

Now the application is running, you are ready to log in to Openspace, see *TIBCO Openspace User's Guide* for more information about using Openspace.

Contributing Your Openspace Gadget

Once the Openspace gadget is available, it can be contributed using the **Contribute Gadget** option in the Openspace Gadget Store.



You must be logged in as a user that has privileges with the ContributeGadget system action assigned to them to use the **Contribute Gadget** option.

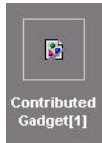
To do this, you must specify the URL of the gadget you want to add. For example `protocol://amxbpmhost:8080/openspace/projectname/gadgetfoldername/MyGadget.gadget.xml` where:

- *protocol* is the communications protocol being used by openspace, either HTTP or HTTPS. This was determined at installation.
- *amxbpmhost* is the machine name or IP address of the machine where TIBCO ActiveMatrix BPM is installed.
- *projectname* is the name of your web application project. The web application project name must be in lowercase.

- *gadgetfoldername* is the name of your gadget folder.
- *MyGadget.gadget.xml* is the name you have given your Openspace Gadget.

When adding a gadget to Openspace, note the following:

- When specifying a thumbnail or screenshot URL in `<ModulePrefs>`:
 - If you do not specify a thumbnail or screenshot URL that is publicly accessible URL, the gadget is displayed in Openspace as shown below:



- If you do not specify any thumbnail or screenshot URL, then Openspace uses its default thumbnail or screenshot and the gadget is displayed in Openspace as shown below:

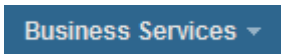


- If you do not specify a description in `<ModulePrefs>`, Openspace defaults to **Contributed Gadget[n]** where *n* is an incremental number for each gadget you add that has no description.

For more information:

- On the OpenSocial Initiative, see <http://opensocial.org>.
- On the OpenSocial Gadgets API Specification, see <http://opensocial.org/page/specs-1>.
- On the constraints that apply to the contribution of gadgets in Openspace, see [Constraints for Contributing Gadgets in Openspace](#).

Procedure

1. Click the down arrow of a tab to display the User Preferences area .
2. Click  next to **More Gadgets**.
3. Click  next to **Contribute Gadget**.
4. In the **Gadget Spec URL** box, type the URL of the gadget you want to add.
The first part of the URL is always `http://amxbpmhost:8080/openspace` where *amxbpmhost* is the IP address of the machine hosting TIBCO Openspace. The rest of the URL is the location you specified in your gadget XML.



Your system administrator can configure Openspace to display a specific Gadget Spec URL by specifying the path to the gadget server using the `useConfiguredServerUrl` property in the `OSProperties.properties` file. See *TIBCO ActiveMatrix BPM Administration* guide for more information.

5. Deselect the **URL is not a gadget spec - it is a website** check box.
The **URL is not a gadget spec - it is a website** checkbox is automatically selected when you start to type in the **Gadget Spec URL** box. This checkbox should be selected if you want to import an arbitrary web page and use it as a gadget in Openspace. See *TIBCO Openspace User's Guide* for more information.

6. In the **Hub Policy URL:** box, type the URL of the Hub Policy for the Openspace Sample Gadget.
The first part of the URL is always `http://amxbpmhost:8080/openspace` where *amxbpmhost* is the IP address of the machine hosting TIBCO Openspace. The rest of the URL is the location you specified in your `HubPolicy.xml`. Once you have typed in the URL of your Hub Policy, the Hub Policy Administrator dialog displays. You can close this dialog as it is for informational purposes only.
7. In the **Category of Gadget:** box, type the name of the category where you want the gadget to be stored in the Openspace Gadget Store. If the category does not already exist, it is created automatically. If you do not specify a category, the gadget is stored in the `UNCATEGORIZED` category.
8. Click **Import Gadget**.
The new gadget is added to the Openspace Gadget Store to the category you specified and is available to be added to a tab. See *TIBCO Openspace User's Guide* for more information.
9. Click **AddGadget** to add the gadget to the gadget store and close the Gadget Store.



The content of gadgets within Openspace may change dynamically. You can specify whether the Openspace user should be able to edit dynamic content, by setting the `isDynamicGadgetEditable` property in the `OSProperties.properties` file. By default, this is set to `false`. To enable editing, it should be set to `true`. See *TIBCO ActiveMatrix BPM - BPM Administration* guide for more information.

Configure the Gadget Hub Policy

You must configure the new gadget hub policy so that the managed events can be published.



You must be logged in as a user that has privileges with the **EditHubPolicy** system action assigned to them to configure the gadget hub policy.

Procedure

1. From the gadget tab you added above, click **More Gadgets**. The Openspace Gadget Store displays.
2. Click **Manage Gadgets > Communication**.
3. Select **Order by Domain**.
4. Select everything listed in the dialog.
5. Click **Save**.

Web Client API Service

The Web Client API provides public services that you can use to develop web clients for use with TIBCO ActiveMatrix BPM.



For information on how to obtain the TIBCO Openspace API, see [Web Client API jar Files](#).



The required system action is specified for each operation. See *TIBCO ActiveMatrix BPM - BPM Developer's Guide* for more information about system actions and their scope.

Authentication

Use **Authentication** operations to manage user authorization.

GetAuthenticatedGuid

Lists the Guid of the authenticated user in the session.

Action	GetAuthenticatedGuid
Return	AuthenticationResult

Required System Action

None.

GetAuthenticatedUserName

Lists the user name of the authenticated user in the session.

Action	GetAuthenticatedUserName
Return	AuthenticationResult

Required System Action

None.

IsActionAuthorised

Test whether the caller is authorized to perform one or more system actions, either globally or within the scope of a particular organization model entity.

- The request must specify the system actions to be tested for authorization and, optionally, an organization model entity that defines the scope of the test.
- The response returns a list of the requested system actions, defining for each its scope and whether the user is authorized to perform the system action.

Action	IsActionAuthorised
--------	------------------------------------

Parameter Notes	• entitytype (optional): Enumerated value defining the type of the organization model entity. • entityguid (optional): Globally unique ID of the organization model entity. (Although this value is unique across all entities, if the entity exists within more than one major version, it will share the same GUID.) • actioncomponent (optional): The name of the BPM component that owns the system action. It can be one of the following: – BRM - Business Resource Management – DAC - Deadline and Calendar – DE - Directory Engine – EC - Event Collector – PE - Process Manager – WSB - Workspace • actionname (optional): The name of the system action. • faultonunauthorised (optional, default=false): Boolean value indicating whether a service fault should be raised if any of the queried system actions are not authorized. If "false", an un-authorized system action will be reported in the response.
Return	IsActionAuthorisedResult

Required System Action

None.

Business Services

Use **BusinessService** operations to get information about and interact with deployed business services.

GetBusinessServices

Lists all the deployed business services.

- The request asks for a list of all the deployed business services.
- The result returns detailed descriptions of all the deployed business services.

Action	GetBusinessServices
Parameter Notes	• startRow: start position in the list of the first item on the response returned to the user. The list is zero-based. To start at the first item, specify 0. • rowCount: total number of business services listed in the response. • category: the business category you require. A category can contain wildcards. See Using Wild Card Characters . • channelId: the channel ID being used. For a GWT client, this is typically <code>opensourceGWTPull_DefaultChannel</code>.
Return	GetBusinessServiceResult

Required System Action

None.

ListBusinessServiceCategories

List categories of all deployed business services.

- The request asks for a list of the categories of all deployed business services.
- The result returns the detailed descriptions of the categories of the deployed business services.

Action	ListBusinessServiceCategories
Parameter Notes	channelId: the channel ID being used. For a GWT client, this is typically <code>openspaceGWTPull_DefaultChannel</code> .
Return	ListBusinessServiceCategoriesResult

Required System Action

None.

StartBusinessService

Starts an instance of a deployed business service.

- The request identifies the template of the business service of which you want to start an instance.
- The result returns the complete information about the started business service instance and the execution state is set to **IN_PROGRESS**.

Action	StartBusinessService
Parameter Notes	• template: the business service template you want to start. This is retrieved by calling GetBusinessServices. • channelId: the channel ID being used. For a GWT client, this is typically <code>openspaceGWTPull_DefaultChannel</code>.
Return	StartBusinessServiceResult

Required System Action

`startBusinessService`.

UpdateBusinessService

Updates the specified business service instance.

- The request identifies the business service instance to be updated.
- The result returns the complete information about the updated business service.

Action	UpdateBusinessService
--------	---------------------------------------

Parameter Notes	• item: the details of the business service that you want to update. For example, input/output fields. • template: the business service template to update. This is retrieved by calling GetBusinessServices . The business service must be started. • channelId: the channel ID being used. For a GWT client, this is typically <code>openspaceGWTPull_DefaultChannel</code>.
Return	UpdateBusinessServiceResult

Required System Action

`startBusinessService`.

Calendar Services

Use the Calendar Services requests to create, delete, edit, and get information about base and overlay calendars and the time zones and calendar entries that they use.

CopyCalendar

Copy an existing calendar to create a new calendar of the same type.

- The request must identify the existing calendar and include a valid name and namespace for the new calendar. Note that null is a valid namespace. The name specified must be unique within the given namespace.
- For base calendars, the properties to be copied from the original calendar will include the time zone and working hours. For both base and overlay calendars, all future calendar entries can also be copied if requested.

Action	CopyCalendar
Parameter Notes	• getNamespace, getName identify a valid namespace. Note that null is a valid namespace. • getDestNamespace, getDestName identify the destination namespace. Note that null is a valid namespace. • isIncludeEntries if true, copies to the new calendar all calendar entries with a future date. Defaults to false. • The result includes the details of the new calendar, with a version number.
Return	CopiedCalendarResult

Required System Action

`writeCalendars`

DeleteCalendar

Delete the named calendar, and all the working day and calendar entries associated with it. The deleted calendar cannot be recovered.

- The association of any calendar references to the deleted calendar will also be deleted; leaving those calendar references unfulfilled.

- In the request, pass the version value returned in the earlier request to retrieve the calendar ([GetWorkCalendar](#)). If the given version value does not match the current value held in the database, a conflicting update must have occurred since the initial request to retrieve the calendar, and the delete request will be rejected. The caller must then perform another [GetWorkCalendar](#) request.
- Amendments to existing calendars will not be reflected in those deadline calculations that have already been performed.

Action	DeleteCalendar
Parameter Notes	version, namespace, name identify the calendar to be deleted.
Return	VoidResult

Required System Action

deleteCalendar.

DeleteCalendarEntries

Delete the calendar entries identified by the GUIDs in the request.

- The version value returned in the earlier request to retrieve the calendar entry ([GetCalendarEntries](#)) should be passed in this [deleteCalendarEntries](#) request. If the given version value does not match the current value held in the database, a conflicting update must have occurred since the initial request to retrieve the Calendar Entry, and the delete request will be rejected. The caller must then perform another [GetCalendarEntries](#) request.
- Amendments to existing Calendar Entries will not be reflected in those deadline calculations that have already been performed.
- The deleted entries cannot be recovered. If any of the entries fails to be deleted, no entries will be deleted.

Action	DeleteCalendarEntries
Parameter Notes	• version, namespace, name identify the calendar from which entries are to be deleted. • entry-guid: The GUID of the calendar entry to be deleted. You can get this from a previous GetCalendarEntries or SaveCalendarEntry operation.
Return	VoidResult

Required System Action

deleteCalendars.

GetCalendarEntries

List the calendar entries for a given date range.

- Returns both working-day and recurring entries (see [SaveCalendarEntry](#) for these types of calendar entry).
- The date and time values of the calendar entries are specified and displayed as time-zone neutral. For base calendars, the time-zone identifier is included in the response.

- In the case of recurring calendar entries, those occurrences that fall within the given date range are listed in the response.

Action	GetCalendarEntries
Parameter Notes	• version, namespace, name identify the calendar from which entries are to be deleted. • entry-guid: The GUID of the calendar entry to be deleted. You can get this from a previous GetCalendarEntries or SaveCalendarEntry operation.
Return	GetCalendarEntriesResult

Required System Action

readCalendars.

GetCalendarReferences

Gets the calendar references for the identified calendar.

- The request must identify the calendar whose calendar references are to be returned.
- The response is an ordered collection of calendar references obtained by navigating the organization model hierarchy to which the identified model entity belongs. Each entry identifies the organization model entity that holds the calendar reference.

Action	GetCalendarReferences
Parameter Notes	End Date, Name, Namespace, Start Date identify the calendar whose references are to be listed.
Return	GetCalendarReferencesResults

Required System Action

browseModel or readCalendars.

GetWorkCalendar

List the details of a named calendar.

- The response also includes a version value that will be used for optimistic locking purposes. The same value should be passed in a request to save the calendar.
- For a base calendar, the time zone and working day entries are included in the response. No other calendar Entries will be included in the response.

Action	GetWorkCalendar
Parameter Notes	namespace, name identify the calendar whose details are to be listed. Note that the default SYSTEM calendar has no namespace.
Return	GetWorkCalendarResult

Required System Action

readCalendars

ListCalendars

List all known base and overlay calendars.

- The request specifies the namespace value, to return a list of only those calendars with the given namespace. If the given namespace ends in an asterisk, all calendars whose namespace starts with the given value will be returned.
- The response returns a summary of each calendar, including namespace, name, type, and any calendar references.

Action	ListCalendars
Parameter Notes	The request specifies the namespace value, to return a list of only those calendars with the given namespace. The default base calendar SYSTEM has no namespace so null is an acceptable value.
Return	ListCalendarResult

Required System Action

readCalendars

ListTimeZones

List all the time zones recognized by [Calendar Services](#).

- The response returns a list of all recognized time zones.
- These time zones are the possible values for the **time-zone** property of a base calendar.
- The calling operation may cache the response, and use the entries to allow the user to select a time zone when creating or editing a base calendar.

Action	ListTimeZones
Parameter Notes	empty: This parameter is a placeholder—its value is not referenced. Therefore, specify null for this parameter.
Return	ListTimeZonesResult

Required System Action

None.

PurgeCalendarEntries

Delete all the calendar entries with an end date earlier than the specified **to-date**, for the named calendar.

- If the **to-date** exceeds the current date, the current date will be used.
- The deleted entries cannot be recovered. If any of the entries fails to be deleted, no entries will be deleted.

Action	PurgeCalendarEntries
Parameter Notes	• version, namespace, name identify the calendar to be purged. • to-date all calendar entries with an end date before this date are deleted. • version if successful, updated in the response.
Return	VoidResult

Required System Action

deleteCalendars.

RemoveCalendarReference

Removes a collection of specified calendar reference GUIDs.

- The request identifies the calendar references to be removed.
- The result returns the calendar reference that has been removed.

Action	RemoveCalendarReference
Parameter Notes	• version, namespace, name identify the calendar to be purged. • references identify the calendar references to delete. • version if successful, updated in the response.
Return	RemoveCalendarReferenceResult

Required System Action

deleteCalendars.

RenameCalendar

Change a calendar's name, its namespace, or both.

- The change will not succeed if another calendar already exists with the new name and namespace combination.
- Any references to the calendar will not be affected by the change.

Action	RenameCalendar
Parameter Notes	• version, old-namespace, old-name identify the existing calendar to be renamed. • new-namespace, new-name the new name and/or namespace to be given to the calendar.
Return	VoidResult

Required System Action

writeCalendars.

ResolveReferences

Resolve a collection of specified calendar reference GUIDs, by returning the identifier of the calendar instance with which each reference is associated.

- A calendar reference is created in TIBCO Business Studio. It consists of either:
 - a calendar alias (free text that identifies a calendar), enabling a user to map the reference to the most appropriate base or overlay calendar, using the Calendar gadget in Openspace, or
 - a reference to a process data field that will provide the calendar alias at runtime.
- The calendar reference GUIDs used in this request might be obtained from the responses to [OpenOrgModel](#).
- More than one calendar reference may be associated with the same calendar.
- The response contains the same list of calendar references as requested, plus the identifier of the calendar instance that each reference is associated with.
- If any reference is not associated with a calendar instance, it is still included in the response but no calendar is identified for that reference.

Action	ResolveReference
Parameter Notes	reference contains the GUIDs of the calendar references to be resolved, one reference per repetition of the parameter.
Return	ResolveReferencesResult

Required System Action

readCalendars.

SaveCalendar

Update the specified calendar if it already exists, or create it if it does not.

- Date and time values in the request are time-zone neutral.
- If the saveCalendar request is one to update an existing Calendar, this request should include the version value returned in the earlier request to retrieve the calendar ([GetWorkCalendar](#)).
- If the given version value does not match the current value held in the database, a conflicting update must have occurred since the initial request to retrieve the calendar, and the save request will be rejected; The caller must then perform another [GetWorkCalendar](#) request, to obtain the current version.
- If the version value shows no conflict, the calendar is updated, and the response includes a new version value.
- Amendments to existing calendars will not be reflected in those deadline calculations that have already been performed.

Action	SaveCalendar
Parameter Notes	Calendar identifies the calendar to be saved.
Return	SaveCalendarResult

Required System Action

writeCalendars.

SaveCalendarEntry

Updates the specified calendar entry if it already exists, or create it if it does not.

- Date and time values in the request must be time-zone neutral.
- A recurring exclusion is defined using the recurrence rule expression, not as a series of individual dates and times. You cannot edit an individual occurrence of a recurring exclusion as a separate entity, so any amendments made to a recurring exclusion will be applied to all occurrences of that exclusion.
- If the request is to update an existing calendar entry, the version value returned in the earlier request to retrieve the calendar entry ([GetCalendarEntries](#)) should be passed in this `saveCalendarEntry` request. If the given version value does not match the current value held in the database, a conflicting update must have occurred since the initial request to retrieve the entry, and the save request will be rejected. The caller must then perform another [GetCalendarEntries](#) request.
- If the version value shows no conflict, the calendar entry will be updated, and the response will include a new version value.
- Amendments to existing calendar entries will not be reflected in those deadline calculations that have already been performed.

Action	SaveCalendarEntry
Parameter Notes	• namespace, name, version identify the calendar to which the entry or entries are to be saved. • start, end start and end dates of the period for which entries should be displayed in the response. Both dates must be specified, the date range cannot be open-ended. The response will show the exclusions that apply over the given start-to-end period. • create-working-day, update-working-day, create-recurring, update-recurring You have a choice of entering any <i>one</i> of these types of calendar entry in one <code>saveCalendarEntry</code> operation. • start, end start and end dates for this particular calendar entry. The end date is not mandatory. If the entry is a recurring one, any use of <i>count</i> or <i>until</i> in the recurrence rule overrides any end date specified here. • duration the length of each occurrence of a recurring entry. Defined as in RFC-5545. • rrule the recurrence rule. This is the expression used to define a recurring calendar entry. The format of the rule is defined as in RFC-5545. For more information about recurring exclusions, see "Adding Recurring Exclusions" in the <i>TIBCO ActiveMatrix BPM Developer's Guide</i>. • guid the unique ID of the calendar entry. – If you are creating a working-day entry or a recurring entry, the response returns the GUID that has been assigned to the new calendar entry. – If you are updating an existing entry, include the GUID of the entry in the request message to identify the correct entry. You can get the GUID from a previous call to GetCalendarEntries .

Return	SaveCalendarEntryResult
--------	---

Required System Action

writeCalendars.

SaveCalendarReference

Save one or more calendar references to an association with a named Calendar.

- A calendar reference is created in TIBCO Business Studio. It consists of:
 - a namespace, a name, and a type (Base or Overlay). These are used by the TIBCO Business Studio user to identify the calendar required, but do not necessarily match the namespace and name of any actual calendar,
 - a GUID generated automatically by TIBCO Business Studio, that provides a unique identifier for this one reference.
- The calendar reference GUIDs used in this request might be obtained from the responses to [OpenOrgModel](#).
- If the request includes any calendar reference that is already associated with another calendar, that association is removed before adding the reference to the calendar identified in the request.
- It is not possible to remove a calendar reference from a calendar without saving it against another calendar.
- The response identifies (by namespace and name) those calendars affected by the request, that is those whose calendar references have been modified. It also lists the updated version number of each calendar.

Action	SaveCalendarReference
Parameter Notes	• version, namespace, name identify the calendar with which the references are to be associated. • reference contains the calendar references to be associated with the specified calendar, one reference per repetition of the parameter.
Return	SaveCalendarReferenceResult

Required System Action

writeCalendars.

SetCalendarReferences

Assigns calendar references to organization model entities. The assignments will overwrite any existing assignments to the same organization model entities.

- The request identifies the assignment of the given calendar alias to the identified organization model entity.
- The response returns a fault if it was an unsuccessful assignment.

Action	SetCalendarReferences
--------	---------------------------------------

Parameter Notes	reference contains the calendar references to be associated with the specified calendar, one reference per repetition of the parameter.
Return	VoidResult

Required System Action

writeCalendars.

Dataview Client Service

Use case data services to perform searches on case data based on specific criteria.

CreateDataView

Allows a search query to be stored as a dataview that can be retrieved later and executed again.

- The request specifies the search query to be used in the dataview. It identifies the case class and the search condition.
- The response returns the dataview ID.

Action	CreateDataView
Parameter Notes	• caseClass The fully qualified name of the case class. For example, <code>com.example.gddemo.Car</code>. • caseModelVersion The version of the case model. For example, 2 or 2.0.0. • category Specifies the category. See "Defining Dataview Categories" in <i>TIBCO ActiveMatrix BPM Developer's Guide</i>. • name The name of your dataview. This must be unique. • description A description of your dataview. • condition Specifies the conditions to use on the searchable attributes. See "Defining the Dataview Condition" in <i>TIBCO ActiveMatrix BPM Developer's Guide</i>. • attrPath Specifies the attribute to sort. See "Defining Attribute Path" in <i>TIBCO ActiveMatrix BPM Developer's Guide</i>. • sort Specifies the sort order, either ASC for ascending or DESC for descending.
Return	CreateDataViewResult

Required System Action

manageDataView

EditDataView

Allows you to edit a saved dataview.

- The request contains a dataview ID or name of the dataview to edit and the changes to make to the dataview.
- The response indicates success by returning the editDataViewResponse.

Action	editDataView
--------	------------------------------

Parameter Notes	• viewID The unique identifier of the dataview. • caseClass The fully qualified name of the case class. For example, <code>com.example.gddemo.Car</code>. • caseModelVersion The version of the case model. For example, 2 or 2.0.0. • category Specifies the category. See "Defining Dataview Categories" in <i>TIBCO ActiveMatrix BPM Developer's Guide</i>. • name The name of your dataview. This must be unique. • description A description of your dataview. • condition Specifies the conditions to use on the searchable attributes. See "Defining the Dataview Condition" in <i>TIBCO ActiveMatrix BPM Developer's Guide</i>. • attrPath Specifies the attribute to sort. See "Defining Attribute Path" in <i>TIBCO ActiveMatrix BPM Developer's Guide</i>. • sort Specifies the sort order, either ASC for ascending or DESC for descending.
Return	editDataViewResult

Required System Action

`manageDataView`

Login Services

Provides the services that enables users to log in.

LoginAction

Allows the resource to login to TIBCO ActiveMatrix BPM applications. The Login action is part of `executeLogin`.

Action	LoginAction
Parameter Notes	• getUser contains the user name of the user to log in to the BPM application. • getPassword contains the password of the user to log in to the BPM application. • isUserInfo can be set to <code>true</code> or <code>false</code>. If set to <code>false</code>, then the <code>LoginResult</code> only returns <code>true</code> or <code>false</code>, depending on whether the log in has been successful or not. If set to <code>true</code>, the <code>ArrayList</code> of the <code>OrgModelEntityID</code> is returned as well as <code>true/false</code>.
Return	LoginResult

```

Example  LoginAction loginAction = new LoginAction(view.getUserName(),
view.getPassword(), view.isUserInfo());
BPMWebClientService.getInstance().executeLogin(loginAction, new
AsyncCallback<LoginResult>()
{
@Override
public void onFailure(Throwable caught)
{
view.onLoginFailure(caught);
}
@Override
public void onSuccess(LoginResult result)
{
BPMWebClientService.getInstance().execute(new
CreateOrUpdateUser(view.getUserName()),
new AsyncCallback<CreateOrUpdateUserResult>()
{
@Override
public void onFailure(Throwable caught)
Window.alert(caught.getMessage());
}
@Override
public void onSuccess(CreateOrUpdateUserResult result)
{
//
}
});
view.onLoginSuccess(result);

```

Required System Action

None.

Logout Services

Provides the services that enables users to log out.

LogOutAction

Allows the resource to log out of TIBCO ActiveMatrix BPM applications.

Action	LogOutAction
Parameter	None.
Notes	
Return	LogOutResult

Required System Action

None.

Organization Model Services

Provides the services that request information about organization models.

GetEntityName

Requests the details of the entities that form the organization model identified by the given major version.

- The request specifies the name of the organizational entity (identified by a GUID).

- The response returns the name of the specified organization entity.

Action	GetEntityName
Parameter Notes	OrgModelEntityID : can be obtained by OpenOrgModel .
Return	StringResult

Required System Action

resolveResource.

GetResourceDetails

Returns details about one or more specified resources.

- The request includes a GUID for each resource whose details you want returned.
- The response returns properties and attributes for the requested resources, such as positions and groups in which the resource is a member, privileges and capabilities held by the resource, the LDAP entry from which the resource is derived, etc.

Action	GetResourceDetails
Parameter Notes	guid of the resource. Can be obtained using OpenOrgModel .
Return	OrgModelEntityIDResult

Required System Action

resolveResource.

OpenOrgModel

Provides an outline of the organization model. It then lists the following organization model entities:

- Organization
- Organization Unit
- Position
- Group
- Location

It also returns

- the unique ID of each of these entities.
- a list of those resources associated with the organization model entities, identified by their unique identifiers (GUIDs).

Action	OpenOrgModel
--------	------------------------------

Parameter Notes	• getVersion: the version number of the organization model you want to open. • isGetCalendarRefs: If set to true, the calendar references for the organizational entities are also retrieved.
Return	OrgModelEntityResult
Example	<pre>private void getOrgModel() { BPMWebClientService.getInstance().execute(new OpenOrgModel(-1,false), new AsyncCallback<OrgModelEntityResult>() { @Override public void onFailure(Throwable caught) { Window.alert("Error: " + caught.getMessage()); } @Override public void onSuccess(OrgModelEntityResult result) { if (result != null && result.getEntities() != null) { for (OrgModelEntity entity : result.getEntities()) { TreeItem item = new TreeItem(entity.getName()); addChildren(item, entity.getChildren()); orgModelTree.addItem(item); } } } }); }</pre>

Required System Action

browseModel.

RunResourceQuery

Provides information about resources in the organization model based on certain criteria.

Action	RunResourceQuery
Parameter Notes	query: the query to execute. This must conform to the RQL syntax. See "Resource Query Language" in <i>TIBCO Business Studio BPM Implementation</i> guide for more information.
Return	OrgModelEntityResult

Required System Action

browseModel and resourceAdmin.

Process Template Services

Use Process Template services operations to manage process templates and process instances.

CancelProcessInstance

Cancel a specific process instance.

- The request must specify the unique ID of the process instance that is to be cancelled.

- The response indicates whether or not the process instance has been successfully cancelled.

Action	CancelProcessInstance
Parameter Notes	processId : A unique alphanumeric value identifying this particular process instance, for example, pvm:0a100. Can be obtained from GetProcessInstanceData .
Return	VoidResult

Required System Action

cancelProcessInstance

GetEventDurationsFromEC

Returns the event durations for a specified process instance.

Action	GetEventDurationsFromEC
Parameter Notes	processId : A unique alphanumeric value identifying this particular process instance, for example, pvm:0a100. Can be obtained from GetProcessInstanceData .
Return	GetEventDurationsFromECResult

Required System Action

queryProcessInstance.

GetMigrationPoints

List the permissible process instance migration points for one or more process templates. (Migration points are the points in the process from which a process instance can be migrated to a different version.)

- The request must specify the process templates for which process instance migration points are to be listed.
- The response returns the valid process instance migration points for the specified process templates.

For information about process migration, see "Migrating Process Instances" in *TIBCO Openspace User's Guide*.

Action	GetMigrationPoints
Parameter Notes	• moduleName: the name of the .xpd1 file where the process is stored. This is defined in TIBCO Business Studio. • processName: the name of the process template. This is defined in TIBCO Business Studio at design-time. • version: identifies the version of the process template.
Return	GetMigrationPointsResult

Required System Action

handleProcessMigration

GetProcessInstanceData

Returns data about process instance instances from the TIBCO ActiveMatrix BPM runtime. Note that the TIBCO ActiveMatrix BPM runtime does not have data about process instances that are CANCELED or COMPLETED. The following information is displayed:

- endTime
- duration
- priority
- originator
- processId
- processState
- startTime

Action	GetProcessInstanceData
Parameter Notes	• moduleName: the name of the .xpdl file where the process is stored. This is defined in TIBCO Business Studio. • processName: the name of the process template. This is defined in TIBCO Business Studio at design-time. • version: identifies the version of the process template.
Return	ProcessInstancesDataResult

Required System Action

queryProcessInstance.

GetProcessInstancesFromEC

Returns data about process instance instances from the TIBCO ActiveMatrix BPM Event Collector. Note that the TIBCO ActiveMatrix BPM Event Collector includes data from process instances that are CANCELED and COMPLETED. The following data is displayed:

- endTime
- duration
- priority
- originator
- processId
- processState
- startTime

Action	GetProcessInstancesFromEC
--------	---

Parameter Notes	filter: a filter string to query Event Collector. For example, to retrieve the data about version 1.0.0 of a process template called process template , the following could be used: <code>processTemplateName = 'templateName' AND version = '1.0.0' AND moduleName = 'moduleName'</code> . See <i>TIBCO ActiveMatrix BPM Developer's Guide</i> for more information about querying Event Collector.
Return	ProcessInstancesDataResult

Required System Action

queryProcessInstance.

GetProcessInstanceStatus

Provides the status of a specific process instance.

See *TIBCO Openspace User's Guide* for more information about the process instance states.

Action	GetProcessInstanceStatus
Parameter Notes	processId: A unique alphanumeric value identifying this particular process instance, for example, <code>pvm:0a100</code> . Can be obtained from GetProcessInstanceData .
Return	ProcessInstancesDataResult

Required System Action

queryProcessInstance.

GetProcessTemplateList

Provides a list of all available process templates. The following data is displayed:

- endTime
- duration
- priority
- originator
- processId
- processState
- startTime

See *TIBCO Openspace User's Guide* for more information about the process template columns.

Action	GetProcessTemplateList
Parameter Notes	• moduleName: the name of the .xpdL file where the process is stored. This is defined in TIBCO Business Studio. • processName: the name of the process template. This is defined in TIBCO Business Studio at design-time.
Return	ProcessTemplateListResult

Required System Action

queryProcessInstance.

IgnoreHaltedProcessInstance

You can force a halted process instance to cancel the task that has halted and continue with the process. This means that the process instance continues processing from the point in the process after the failed activity. This could mean, for example, that a field may not have been set but the process would continue as if the field had been set.

You cannot do this if the process instance is in a failed state. See *TIBCO Openspace User's Guide* for more information.

Action	IgnoreHaltedProcessInstance
Parameter Notes	processId : A unique alphanumeric value identifying this particular process instance, for example, pvm:0a100. Can be obtained from GetProcessInstanceData .
Return	StringResult

Required System Action

haltedProcessAdministration

ListMigrationRules

Provides a list of the process instance migration rules that are set for a specified process template.

- The request must specify the process templates for which process instance migration rules are to be listed.
- The response returns the process instance migration rules that are set for the specified process templates.

Action	ListMigrationRules
Parameter Notes	• moduleName: the name of the .xpdl file where the process is stored. This is defined in TIBCO Business Studio. • processName: the name of the process template. This is defined in TIBCO Business Studio at design-time.
Return	ListMigrationRulesResult

Required System Action

handleProcessMigration.

QueryEventCollector

List process instances that match certain criteria.

Action	QueryEventCollector
--------	-------------------------------------

Parameter	query: executes the specified query against Event Collector. For example, to retrieve all the Process Engine events for instance ID pvm:0a100, the following string could be used: <code>componendId = 'BX' AND messageCategory = 'PROCESS_INSTANCE' AND managedObjectId = 'pvm:0a100'</code> . See <i>TIBCO ActiveMatrix BPM Developer's Guide</i> for more information about executing queries against Event Collector.
Return	QueryEventCollectorResult

Required System Action

queryProcessInstance.

QueryHaltedProcessInstances

List process instances that have halted (as a result of an activity failure) that match certain criteria:

- The request must specify a SQL query that defines the search criteria.
- The response returns a list of HALTED process instances that match the specified criteria, either as a complete list or as the first page of a paged list (in which case a paging ID is also returned).



This operation only queries HALTED process instances. To list process instances that are in a state other than HALTED, use [QueryProcessInstances](#) instead.

Action	QueryHaltedProcessInstances
Parameter Notes	• query: executes the specified query against Event Collector. • attributeMap: If a user-defined attribute is used in the SELECT statement of a paginated query, that attribute and its type should be included in the attributeMap, otherwise a database query must be made to get the type of the user-defined attribute, making the request less efficient. • pageSize: Specifies the size of the page. Specify 0 for a non-paginated result. If you specify a paged result (<code>pageSize = -1</code> or a number greater than 0), you must clear resources when you are finished with the paged list. (Note that the result limit for non-paginated queries (<code>pageSize=0</code>) is 500.)
Return	QueryHaltedProcessInstancesResult

Required System Actions

queryProcessInstance.

QueryProcessInstances

Query information about process instances that match specified criteria.

- The request must specify a query that defines the search criteria.
- The response returns information from process instances according to the specified criteria, either as a complete list or as the first page of a paged list (in which case a paging ID is also returned).

To list process instances that are in a HALTED state, use [QueryHaltedProcessInstances](#) instead.

Action	QueryProcessInstances
--------	---------------------------------------

Parameter Notes	• query: For information about valid syntax, see "Sorting and Filtering Lists of Process Templates and Instances" in the <i>TIBCO ActiveMatrix BPM Developer's Guide</i>. • attributeMap: If a user-defined attribute is used in the SELECT statement of a paginated query, that attribute and its type should be included in the attributeMap, otherwise a database query must be made to get the type of the user-defined attribute, making the request less efficient (see the example below). • pageSize: Specifies the size of the page. Specify 0 for a non-paginated result. If you specify a paged result (pageSize = -1 or a number greater than 0), you must clear resources when you are finished with the paged list. (Note that the result limit for non-paginated queries (pageSize=0) is 500.)
Return	QueryProcessInstancesResult

Required System Actions

queryProcessInstance

QueryProcessTemplates

Query information about process templates that match specified criteria.

- The request must specify a query that defines the search criteria.
- The response returns information from process templates according to the specified criteria, either as a complete list or as the first page of a paged list (in which case a paging ID is also returned).

Action	QueryProcessTemplates
Parameter Notes	• filter: a filter string to query Event Collector. For example, the following filter string could be used: <code>DEFINITION.NAME='abc' AND DEFINITION.DESRIPTION>'desc' AND DEFINITION.VERSION<>'1.2.3'</code>. An empty filter returns all templates. • sort: a sort string to query Event Collector. For example, the following sort string could be used: <code>DEFINITION.NAME ASC , DEFINITION.DESRIPTION ASC , DEFINITION.VERSION DESC</code>. See <i>TIBCO ActiveMatrix BPM - BPM Developer's Guide</i> for more information about executing queries against Event Collector. • startItem: index of first process template to be retrieved. • numItems: the number of process templates to retrieve.
Return	ProcessTemplateListResult

Required System Actions

queryProcessTemplate.

ResumeHaltedProcessInstance

Resume execution of a process instance that has halted as a result of an activity failure. Execution continues as if the activity had failed:

- The request must specify the ID of the process instance that is to be resumed.

- The response indicates whether the process instance has been successfully resumed.



The process instance immediately enters a FAILED state. This operation effectively allows the erroring process instance to resume and then fail as if it had never halted. It can be used when investigation of the halted process instance indicates that nothing can be done to retrieve it.

Action	ResumeHaltedProcessInstance
Parameter Notes	processId : A unique alphanumeric value identifying this particular process instance, for example, pvm:0a100. The specified process instance must be in a HALTED state for the request to be valid. An <code>illegalStateException</code> is returned if the process instance is in any other state.
Return	ResumeHaltedProcessInstanceResult

Required System Action

haltedProcessAdministration.

ResumeProcessInstance

Resume a previously suspended process instance. See also: [SuspendProcessInstance](#).

- The request must specify the unique ID of the process instance that is to be resumed.
- The response indicates whether or not the process instance has been successfully resumed.



This operation can only be used on a process instance that is in a SUSPENDED state. To resume a process instance that is in a HALTED state (as a result of an activity failure) you can use [ResumeHaltedProcessInstance](#) to resume the process instance as if the activity had failed. (The process instance immediately enters the FAILED state.)

Action	ResumeProcessInstance
Parameter Notes	processId : A unique alphanumeric value identifying this particular process instance, for example, pvm:0a100.
Return	VoidResult

Required System Action

haltedProcessAdministration

RetryHaltedProcessInstance

Progress a halted process instance by retrying the task that has failed. For example, you may want to do this if a database task has failed because the database connection was lost but is now restored. You cannot do this if the process instance is in a failed state.

Action	RetryHaltedProcessInstance
Parameter Notes	processId : A unique alphanumeric value identifying this particular process instance, for example, pvm:0a100. Can be obtained from GetProcessInstanceData .
Return	StringResult

Required System Action

haltedProcessAdministration

SetMigrationRules

Set one or more process instance migration rules. (Each migration rule consists of a unique set of module name, module version from which to migrate, process template name, task name from which to migrate, and module version to which to migrate.)

- The request must specify the process instance migration rules that are to be set.
- The response returns whether the specified process instance migration rules have been successfully set.

Action	SetMigrationRules
Parameter Notes	• processName: the name of the process template. This is defined in TIBCO Business Studio at design-time. • moduleName: the name of the xpdL file where the process is stored. This is defined in TIBCO Business Studio. • rules: Details of the migration rules to be applied. The map should be keyed on the source version for the migration. Note that: – Each source version in the map should contain a list of Migration Rule entries for that version. – Rules with <code>isApplied = true</code> and <code>isDeleted = true</code> are removed from the Engine. – Rules with <code>isApplied = false</code> and <code>isDeleted = false</code> are applied to the Engine. – The client will keep track of which rules have been applied and set the <code>isApplied</code> flag appropriately.
Return	VoidResult

Required System Action

handleProcessMigration.

StartProcess

Create a process instance using a starter operation.

- The request must specify the starter operation and its associated parameter values that should be used to create the process instance.
- The response returns the unique ID of the newly created process instance.

Action	StartProcess
--------	------------------------------

Parameter	• moduleName: the name of the .xpd file where the process is stored. This is defined in TIBCO Business Studio. • processToStart: the name of the process template to start an instance of. • version: identifies the version of the process template. • starterOperation: the starter operation to perform. This can be retrieved by calling QueryProcessTemplates .
Return	StartProcessResult

Required System Action

startProcess.

SuspendProcessInstance

Suspend a process instance from being scheduled for execution.

- The request must specify the unique ID of the process instance that is to be suspended.
- The response indicates whether or not the specified deadline has been successfully set.

Action	SuspendProcessInstance
Parameter Notes	processId : A unique alphanumeric value identifying this particular process instance, for example, pvm:0a100.
Return	VoidResult

Required System Action

suspendProcessInstance

bulkSuspendProcessInstance (required if suspending multiple process instances)

Session

Use **Session** operations to manage your Openspace sessions.

InvalidateSession

Invalidates your current Openspace session. For example, if you can move to another application from your Openspace gadget, you can then invalidate the current Openspace session. This may be necessary for security reasons.

Action	InvalidateSession
Return	VoidResult

Required System Action

None.

SetSessionMaxInactiveInterval

Set the maximum inactive timeout for the current session.

Action	SetSessionMaxInactiveInterval
Parameter Notes	maxInactiveInterval: defines the timeout period for a user session. If the defined period has lapsed without activity, the Openspace session times out. The value is in seconds, and the default timeout is 30 minutes.
Return	SessionResult

Required System Action

None.

Work Item Services

Provides the actions that are responsible for distributing and managing work.

AllocateWorkItem

Allocate a work item to a single resource:

- The request must specify the work item that should be allocated and the resource to whom it is to be allocated.
- The response returns details of the allocated work item.

This operation can only be used if the work item is in a state from which it can be allocated.

Action	AllocateWorkItem
Parameter Notes	• WorkItem: uses sets of ManagedId objects to identify a set of work items to allocate. A ManagedId object consists of a work item Id and version. • userId: the ID (GUID) of the user to allocate the work item to.
Return	VoidResult

```

Example    private void allocateWorkItem(WorkListItem item)
           {
           ManagedId id = toManagedObject(item);
           HashSet<ManagedId> ids=new HashSet<ManagedId>();
           ids.add(id);
           BPMWebClientService.getInstance().execute(new AllocateWorkItem(ids,
           ClientConstants.USER_ID),
           new AsyncCallback<VoidResult>()
           {
           @Override
           public void onFailure(Throwable caught)
           {
           Window.alert("Error: " + caught.getMessage());
           }
           @Override
           public void onSuccess(VoidResult result)
           {
           Window.alert("Work item allocated");
           getWorkList();
           }
           });
           }

```

Required System Action

workItemAllocation.

CancelWorkItem

Data entered or changed for the specified workitem is canceled.

Action	CancelWorkItem
Parameter	• userId: the ID (GUID) of the user to allocate the work item to. • item: the WorkItem retrieved from a call to OpenWorkItem or OpenNextWorkItem . • channelId: the channel ID being used. For a GWT client, this is typically openspaceGWTPull_DefaultChannel. • channelType: the channel type being used. For a GWT client, this is typically GWTChannel.
Return	VoidResult

```

Example    private void cancelWorkItem(WorkItem item)
           {
           ManagedId id = toManagedObject(item);
           BPMWebClientService.getInstance().execute(
           new CancelWorkItem(ClientConstants.USER_ID, id,
           ClientConstants.CHANNEL_ID, ClientConstants.CHANNEL_TYPE), new
           AsyncCallback<VoidResult>()
           {
           @Override
           public void onFailure(Throwable caught)
           {
           Window.alert("Error: " + caught.getMessage());
           }
           @Override
           public void onSuccess(VoidResult result)
           {
           Window.alert("Work item cancelled");
           getWorkList();
           }
           });
           }

```

Required System Action

cancelWorkItem.

CloseWorkItem

Closes a work item (and updates the associated input and output data).

- The request must specify the work item that should be closed.
- The response returns the ID of the closed work item.
- This operation puts the work item into a PendHidden state if a hidden period is specified, or into a Pended state if not.
- This operation can only be used if the work item is in a state from which it can be closed. See *TIBCO BPM Developers Guide* for information.
- This operation can only be used if the work item is assigned to the user with whose credentials the operation is being invoked.

Action	CloseWorkItem
Parameter Notes	• userId: the ID (GUID) of the user to close the work item. • item: the WorkItem retrieved from a call to OpenWorkItem or OpenNextWorkItem . • channelId: the channel ID being used. For a GWT client, this is typically <code>openspaceGWTPull_DefaultChannel</code>. • channelType: the channel type being used. For a GWT client, this is typically <code>GWTChannel</code>.
Return	WorkItemResult

```

Example    private void closeWorkItem(WorkListItem item)
           {
           ManagedId id = toManagedObject(item);
           HashSet<ManagedId> ids=new HashSet<ManagedId>();
           ids.add(id);
           BPMWebClientService.getInstance().execute(new CloseWorkItems(ids), new
           AsyncCallback<VoidResult>()
           {
           @Override
           public void onFailure(Throwable caught)
           {
           Window.alert("Error: " + caught.getMessage());
           }
           @Override
           public void onSuccess(VoidResult result)
           {
           Window.alert("Work item closed");
           getWorkList();
           }
           });
           }

```

Required System Action

Requires `closeOtherResourcesItems` in order to close a work item currently allocated to another user—for example, when a manager is closing an item in a supervised work list (see *TIBCO ActiveMatrix BPM - BPM Developer's Guide* for information on supervised work lists).

Otherwise, none.

CloseWorkItems

Closes previously opened Work Items. It closes work items, even if they were previously opened by other users, without saving the data.



Calling this service could result in loss of data, if changes to a work item have been made by another user and not saved.

The specified open work items are closed and returned to their offered state.

Action	CloseWorkItems
Parameter Notes	workItemIds: uses sets of <code>ManagedId</code> objects to identify a set of work items to close. A <code>ManagedId</code> object consists of a work item Id and version.
Return	VoidResult

Required System Action

`CloseOtherResourcesItems`.

CompleteWorkItem

Complete a work item (and update the associated input and output data).

- The request must specify the work item that should be completed.
- The response indicates whether or not the operation succeeded.

Action	CompleteWorkItem
--------	----------------------------------

Parameter	• item: the WorkItem retrieved from a call to OpenWorkItem or OpenNextWorkItem. • channelId: the channel ID being used. For a GWT client, this is typically <code>openspaceGWTPull_DefaultChannel</code>. • channelType: the channel type being used. For a GWT client, this is typically <code>GWChannel</code>. • userId: the ID (GUID) of the user to close the work item.
Return	WorkItemResult
Example	<pre>private void completeWorkItem(WorkItem item) { BPMWebClientService.getInstance().execute(new CompleteWorkItem(item, ClientConstants.CHANNEL_ID, ClientConstants.CHANNEL_TYPE, ClientConstants.USER_ID), new AsyncCallback<WorkItemResult>() { @Override public void onFailure(Throwable caught) { Window.alert("Error: " + caught.getMessage()); } @Override public void onSuccess(WorkItemResult result) { Window.alert("Work item completed :" + result.getToDoItem().getId()); getWorkList(); } }); }</pre>

Required System Action

None.

GetOrderandFilterForResource

Retrieves the sort/filter criteria for work item lists for a resource. If sort/filter criteria is set for a resource it is applied whenever the work list for the specified resource is retrieved. The setting is persisted in the BPM database.

- The request must specify the sort/filter criteria to retrieve.
- The response returns whether the request succeeded or failed.

Action	getOrderAndFilterForResource
Parameter Notes	orderFilterCriteria : the sort and filter criteria to set. See <i>TIBCO Openspace User's Guide</i> for details of the syntax and content of these criteria.
Return	OrderFilterResult

Required System Action

`getResourceOrderFilterCriteria`.

GetOrgEntityConfigAttribute

Gets the value of the specified configuration attribute for the calling resource. For example, for OpenNextWorkItem, the value can be set to either TRUE or FALSE.

- The request must specify the name of the configuration attribute whose value you want, for example, OpenNextWorkItem.
- The response returns the attribute value for the configuration attribute you specified, for example, whether the value is TRUE or FALSE.

Action	getOrgEntityConfigAttribute
Parameter Notes	attributeName: the name of the configuration attribute whose value you want.
Return	StringResult

Required System Action

None.

GetOfferSet

Get the offer set for a work item (The offer set is the set of organization model entities to whom a work item is to be offered.)

- The request must specify the work item for which the offer set is required.
- The response returns the IDs of the organization model entities that comprise the offer set for this work item.

Action	GetOfferSet
Parameter Notes	item: work item to retrieve the offer set for. A list of work items can be retrieved by calling GetToDoList .
Return	GetOfferSetResult

Required System Action

None.

GetToDoList

Provides a list of workitems for a specified user.

Action	GetToDoList
--------	-----------------------------

Parameter Notes	• startPos: index of the first work item to retrieve. The list is zero-based. Specify 0 to get the first work item in the list. • numberOfItems: number of work items to retrieve. • order: a string that specifies the order in which the work items should be retrieved. For example, appInstance ASC , appInstanceDescription ASC , id ASC , priority ASC , startDate ASC , endDate ASC , attribute1 ASC , attribute2 ASC , attribute3 ASC , attribute4 ASC. See <i>TIBCO ActiveMatrix BPM - BPM Developer's Guide</i> for more information about WorkListService operations that can be included in an expression when defining the sort and/or filter criteria to be applied to a work item list. • resourceId: the user ID (GUID) of the user retrieving the work item list.
Return	GetToDoListResult
Example	<pre> private void reofferWorkItem(WorkListItem item) { ManagedId id = toManagedObject(item); HashSet<ManagedId> ids=new HashSet<ManagedId>(); ids.add(id); BPMWebClientService.getInstance().execute(new ReofferWorkItem(ids), new AsyncCallback<VoidResult>() { @Override public void onFailure(Throwable caught) { Window.alert("Error: " + caught.getMessage()); } @Override public void onSuccess(VoidResult result) { Window.alert("Work item re-offered"); getWorkList(); } }); } </pre>

Required System Action

viewWorkList.

OpenNextWorkItem

Allocate the next available work item for a specified resource to that resource and immediately open the work item (to get the associated input and output data). "Available" means a work item that is not locked nor suspended. The operation fails if there is no next item in the resource's work list.

- The request must specify the required resource.
- The response returns full details of the allocated work item.

This operation can only be used if the work item is in a state from which it can be allocated and opened.

Action	OpenNextWorkItem
--------	----------------------------------

Parameter Notes	• channelId: the channel ID being used. For a GWT client, this is typically <code>openspaceGWTPull_DefaultChannel</code>. • channelType: the channel type being used. For a GWT client, this is typically <code>GWChannel</code>. • userId: the user ID (GUID) of the user opening the next work item.
Return	WorkItemResult

Required System Action

`workItemAllocation`.

OpenWorkItem

Allocate a work item to a single resource and immediately open the work item (to get the associated input and output data).

- The request must specify the work item that should be allocated and the name of the resource to whom it is to be allocated.
- The response returns full details of the allocated work item.

This operation can only be used if the work item is in a state from which it can be allocated and opened.

Action	OpenWorkItem
Parameter Notes	• channelId: the channel ID being used. For a GWT client, this is typically <code>openspaceGWTPull_DefaultChannel</code>. • channelType: the channel type being used. For a GWT client, this is typically <code>GWChannel</code>. • userId: the user ID (GUID) of the user opening the work item.
Return	WorkItemResult

```

Example    private void openWorkItem(WorkListItem item)
           {
           ManagedId id = toManagedObject(item);
           BPMWebClientService.getInstance().execute(
           new OpenWorkItem(id, ClientConstants.CHANNEL_ID,
           ClientConstants.CHANNEL_TYPE, ClientConstants.USER_ID), new
           AsyncCallback<WorkItemResult>()
           {
           @Override
           public void onFailure(Throwable caught)
           {
           if (caught instanceof DispatchException)
           {
           }
           Window.alert("Error: " + caught.getMessage());
           openItem = null;
           }
           @Override
           public void onSuccess(WorkItemResult result)
           {
           Window.alert("Work item opened :" + result.getToDoItem().getId());
           openItem = result.getToDoItem();
           getWorkList();
           int index =
           result.getToDoItem().getFormURL().indexOf(ClientConstants.CHANNEL_ID);
           final String bomJSPPath =
           result.getToDoItem().getFormURL().substring(0, index - 1);
           FormRunner.loadForm(result.getToDoItem().getFormURL(),
           result.getToDoItem().getDataFeed(),
           bomJSPPath, "en_US", formPanel.getElement().getId(),
           new DefaultFormLoaderCallbackHandler());
           }
           });
           }

```

Required System Action

workItemAllocation.

PendWorkItemByDate

The specified work item is hidden from the worklist until a specified date. A Work Item can only be Pended if it is Allocated to the current user. The specified workitem is hidden from the worklist for the specified period. Note that a pendWorkItem operation that specifies a hiddenPeriod of 0 cancels a hiddenPeriod set by a previous pendWorkItem operation on the same work item.

Action	PendWorkItemByDate
Parameter Notes	item: work item to pend. This can be retrieved by calling GetToDoList. pendDate: The date to pend the work item until.
Return	VoidResult

Required System Action

pendWorkItem.

PendWorkItemByPeriod

The specified work item is hidden from the worklist until a specified period. A Work Item can only be Pended if it is Allocated to the current user. The specified workitem is hidden from the worklist for the

specified period. Note that a `pendWorkItem` operation that specifies a `hiddenPeriod` of `0` cancels a `hiddenPeriod` set by a previous `pendWorkItem` operation on the same work item.

Action	PendWorkItemByPeriod
Parameter Notes	• item: work item to pend. This can be retrieved by calling GetToDoList. • years: the number of years to pend the work item to or 0. • months: the number of months to pend the work item to or 0. • weeks: the number of weeks to pend the work item to or 0. • days: the number of days to pend the work item to or 0. • hours: the number of hours to pend the work item to or 0. • minutes: the number of minutes to pend the work item to or 0.
Return	VoidResult
Example	<pre>private void pendWorkItem(WorkListItem item) { ManagedId id = toManagedObject(item); HashSet<ManagedId> ids=new HashSet<ManagedId>(); ids.add(id); BPMWebClientService.getInstance().execute(new PendWorkItemByPeriod(ids, 0, 0, 0, 0, 0, 1), new AsyncCallback<VoidResult>() { @Override public void onFailure(Throwable caught) { Window.alert("Error: " + caught.getMessage()); } @Override public void onSuccess(VoidResult result) { Window.alert("Work item pended for 1 minute"); getWorkList(); } }); }</pre>

Required System Action

`pendWorkItem`.

PrioritiseWorkItem

Alters the priority of the specified work items.

Action	PrioritiseWorkItem
--------	------------------------------------

Parameter Notes	• Items: The list of work items to change the priority on. This can be retrieved by calling GetToDoList. • Priority: The priority to set (absolute or relative). The resultant priority should be between 0 and 100. • isAbsolute: set absolute or relative priority. If <code>true</code> the priority is set to the value specified, otherwise the priority is changed relative to the current priority. If <code>false</code> negative and positive values can be used to alter the current priority. For example, if you have a work item whose priority is 50 and – <code>Priority = -10</code> – <code>isAbsolute = false</code> then the work item priority would be set to 40 or if <code>Priority</code> is <code>+10</code> then the work item priority would be set to 60.
Return	VoidResult

Required System Action

`changeAnyWorkItemPriority`

`changeAllocatedWorkItemPriority`

ReAllocateWorkItem

Reallocate a work item to a different resource and update the work item with new data.

- The request must specify the work item and the data that should be reallocated, and the resource to whom it is to be reallocated.
- The response returns details of the reallocated work item.
- Note that the system actions held by the caller restrict the users to whom a work item can be reallocated— see below. Also, to reallocate a work item to the original offer set, the work item must have a state of Offered, Allocated, or Pending; to reallocate a work item to the "the world", the work item must have a state of Allocated or Pending.

Action	ReAllocateWorkItem
Parameter Notes	• userId: the user ID (GUID) of the user opening the work item. • item: the work item to reallocate. This can be retrieved by calling GetToDoList.
Return	VoidResult

Required System Actions

- [reallocateToOfferSet](#) - This system action allows you to reallocate work items to only the users in the original offer set.
- [reallocateWorkItemToWorld](#) - This system action allows you to reallocate work items to any user in the organization model.

SetOrderAndFilterForResource

Set sort/filter criteria for work item lists for a resource. If sort/filter criteria is set for a resource it is applied whenever the work list for the specified resource is retrieved. The setting is persisted in the BPM database.

- The request must specify the sort/filter criteria to set and identify the resource for whom they should be set.
- The response returns whether the request succeeded or failed.

Action	setOrderAndFilterForResource
Parameter Notes	• resourceID: the GUID of the resource for whose work list you want to set sort and filter criteria. • orderFilterCriteria: the sort and filter criteria to set. See TIBCO Openspace User's Guide for details of the syntax and content of these criteria.
Return	VoidResult

Required System Action

setResourceOrderFilterCriteria.

SetOrgEntityConfigAttribute

Sets the value of a configuration attributes for the calling resource.

- The request must specify the configuration attribute name and the value you want to set.
- The response is as follows:
 - if an error occurs an exception is thrown,
 - otherwise, an empty result is returned. There is no indication of success in the response.

Action	setOrgEntityConfigAttribute
Parameter Notes	• attributeName: the name of the configuration attribute you want to set. • attributeValue: the value of the configuration attribute you want to set.
Return	VoidResult

Required System Action

The autoOpenNextWorkItem system action is required to set the WorkItemAutoOpen attribute. Otherwise, none.

ReofferWorkItem

Workitems that are currently offered to specific user are re-offered to the users in the original offer set.

Action	ReofferWorkItem
--------	---------------------------------

Parameter Notes	item : the work item to reoffer. This can be retrieved by calling GetToDoList .
Return	VoidResult
Example	<pre> private void reofferWorkItem(WorkListItem item) { ManagedId id = toManagedObject(item); HashSet<ManagedId> ids=new HashSet<ManagedId>(); ids.add(id); BPMWebClientService.getInstance().execute(new ReofferWorkItem(ids), new AsyncCallback<VoidResult>() { @Override public void onFailure(Throwable caught) { Window.alert("Error: " + caught.getMessage()); } @Override public void onSuccess(VoidResult result) { Window.alert("Work item re-offered"); getWorkList(); } }); } </pre>

Required System Action

reallocateToOfferSet

SkipWorkItem

Skip a work item. This means that no action is carried out on the work item.

- The request must specify the work item that should be skipped.
- The response identifies whether the request succeeded or failed.
- This operation can only be carried out on work items that have default values for all required output and in/out data fields, or have values set already for such fields.

Action	SkipWorkItem
Parameter Notes	item : the work item to reoffer. This can be retrieved by calling GetToDoList .
Return	VoidResult

Required System Action

skipWorkItem

Work Views Services

Use **WorkListService** requests to get work item lists for organization model entities, and to get or to set sort/filter criteria for work item lists.

AddResourceToView

Enable the calling organizational model entity to view a public work view. The work view must be specified as public either when it is created (using [CreateWorkListView](#)) or by editing it (using [EditWorkListView](#)). An Author can also use this operation to add themselves as a user to a view.

- The request must specify the unique entityID of the work view that the organizational model entity requires access to.
- The response contains the unique entity ID of the work view.

Action	AddResourceToView
Parameter Notes	• workListViewID: the unique entity ID of the work view.
Return	VoidResult

Required System Action
viewWorkList

CreateWorkListView

Create a new work view.

- The request must specify a name and the details of the new work list view. For example, a description, any sort/filter criteria to be applied to this work view, the organizational entity whose work items you want to view or the resources that can access the work view.
- The response contains the unique entity ID (workListViewID) of the new work view.

Action	CreateWorkListView
--------	------------------------------------

Parameter Notes	• Name: the name of the work view. • Description (Optional): description of the work view. • Owner (Optional): The resourceID of the owner of this work view. You can get the resourceID from a lookupUser operation. If omitted, the ID of the resource calling the createWorkListView API is used. • Public (Optional): whether the work view can be viewed by all TIBCO ActiveMatrix BPM users. If true then it is public. • entityID (Optional): the GUID of the organizational entity whose work list you want to retrieve. If the entity is left empty, the view is of the calling resource's own work list. For a public view, this means each user who logs in has the filter and sort criteria from the view applied to their own work list. • orderFilterCriteria (Optional): Specification of sort/filter criteria to be applied to the work item list. See <i>TIBCO ActiveMatrix BPM - BPM Developer's Guide</i> for more information about order and filter syntax. • authors (Optional): Array of XmlModelEntityId objects for entities (GROUPS, POSITIONS, ORGUNITs and RESOURCES) who are allowed to edit the view. • users (Optional): Array of XmlModelEntityId objects for resources who are allowed to use the view. • customData (Optional): A string attribute used to provide any custom data to be stored with the work list view. can be used to store any data that you want to associate with your work list view. For example, if you are using a custom client, you may want to specify which work item columns to display when using this work list view. If storing XML data, the value should be wrapped in a CDATA section.  If you are using the API with TIBCO Openspace you must not use the customData parameter. This is because TIBCO Openspace uses customData to store any of its data that is associated with the work list view. For example, the work item columns to display.
Return	WorkListIDResult

Required System Action

viewWorkList.

DeleteResourceFromView

Remove a public work view from the work view list for the calling resource. An Author can also use this operation to remove themselves from a user's list of a non-public view.

- The request must specify the unique entity ID of the work view that the calling resource no longer requires access to.
- The response contains the unique entity ID of the work view.

Action	DeleteResourceFromView
Parameter Notes	workListViewID: the unique entity ID of the work view.
Return	VoidResult

Required System Action

viewWorkList.

DeleteWorkListView

Enables the calling resource to delete an existing work list view. The calling resource must be an Owner or Author of the work view to be able to delete it. The work view must be locked using [GetWorkViewDetails](#) before it can be deleted.

- The request must specify the unique entity ID of the work list view to be deleted.
- The response contains the unique entity ID of the deleted work list view.

Action	DeleteWorkListView
Parameter Notes	workListViewID : the unique entity ID of the work view.
Return	WorkListIDResult

Required System Action

viewWorkList

EditWorkListView

Edit an existing work view. A work view must be locked using the [GetWorkViewDetails](#) operation before running the [EditWorkListView](#) operation.

- The request must specify the name and the unique entity ID of the work view to be edited. It can also optionally specify the details of the work view that you want to edit. For example, its description, any sort/filter criteria or the organizational model entities that can access the work view. If any of the optional elements are specified, they overwrite the existing attributes specified for this view.



You only need to pass the data you want to change. Parameters that are not passed are unchanged.

- The response contains the unique entity ID of the edited work view.

Action	EditWorkListView
--------	----------------------------------

Parameter Notes	• Name: the name of the work view. • Description (Optional): description of the work view. • Owner (Optional): The resourceID of the owner of this work view. You can get the resourceID from a lookupUser operation. If omitted, the ID of the resource calling the EditWorkListView API is used. • Public (Optional): whether the work view can be viewed by all TIBCO ActiveMatrix BPM users. If true then it is public. • entityID (Optional): the GUID of the organizational entity whose work list you want to retrieve.  If the entity has been previously set it cannot be cleared but it can be changed. This means if you create a view as tibco-admin, you cannot change the view back to a view of your own work list as you would have to set the entity to an empty string. • orderFilterCriteria (Optional): Specification of sort/filter criteria to be applied to the work item list. See <i>TIBCO ActiveMatrix BPM Developer's Guide</i> for more information about order and filter syntax. • authors (Optional): Array of XmlModelEntityId objects for entities (GROUPS, POSITIONS, ORGUNITS and RESOURCES) who are allowed to edit the view. • users (Optional): Array of XmlModelEntityId objects for resources who are allowed to use the view. • customData (Optional): A string attribute used to provide any custom data to be stored with the work list view. can be used to store any data that you want to associate with your work list view. For example, if you are using a custom client, you may want to specify which work item columns to display when using this work list view. If storing XML data, the value should be wrapped in a CDATA section.  If you are using the API with TIBCO Openspace you must not use the customData parameter. This is because TIBCO Openspace uses customData to store any of its data that is associated with the work list view. For example, the work item columns to display.
Return	WorkListIDResult

Required System Action

viewWorkList

GetEditableWorkViews

Retrieve a list of work views that the current resource is allowed to edit. In other words, work views that the current resource is an Owner or Author of.

- The request must specify the start position and number of work items to retrieve.
- The response contains the work list for the calling resource ID.

Action	getEditableWorkViews
--------	--------------------------------------

Parameter Notes	• startPosition: the position in the work list from which to start the page of results. The list is zero-based. To start at the first item, specify 0. • numberOfItems: the number of items from the work view to include.
Return	WorkViewsResult

Required System Action

viewWorkList

GetPublicWorkViews

Retrieves a list of all public work views. Work views are specified as public either when they are created (using [CreateWorkListView](#)) or edited (using [GetWorkViewDetails](#)).

- The request must specify the start position and number of items to retrieve.
- The response contains the unique entity ID of the work view. It also displays any optional work list view details, if they are specified. For example, its description, creation date and owner.

Action	getPublicWorkViews
Parameter Notes	• startPosition: the position in the work list from which to start the page of results. The list is zero-based. To start at the first item, specify 0. • numberOfItems: the number of items from the work view to include.
Return	WorkViewsResult

Required System Action

viewWorkList

GetWorkItemsForView

Retrieves the work item list for a specific work view. This must be executed as a resource who has access to the work view.

- The request must specify the unique entity ID of the work list view and the number of items to retrieve.
- The response contains the work item list for the specified work view.

Action	getWorkItemsForView
Parameter Notes	• startPosition: the position in the work list from which to start the page of results. The list is zero-based. To start at the first item, specify 0. • numberOfItems: the number of items from the work view to include.
Return	WorkViewsResult

Required System Action

viewWorkList

GetWorkViewDetails

Retrieve the details of a specific work view. For example, its description, creation date, whether or not its public and its owner. It can also be used to optionally lock a work view for editing. If you want to edit a work view using the [EditWorkListView](#) operation, you must lock it first using this operation.

- The request must specify the unique entity ID (workViewListID) of the work view to be retrieved. It can also optionally specify whether or not this work view should be locked for editing.
- The response contains the name of the work view, its unique entity ID and the date it was created and/or modified. It also displays any optional work list view details, if they are specified .

Action	getWorkViewDetails
Parameter Notes	• workListViewID: the unique entity ID of the work view. • lockView (Optional): whether or not the work list view should be locked for editing. Set lockView as follows: – 0: the work view is not locked – 1: a lock record is placed against the view for the calling resource ID. The lock is removed when the view is updated or an explicit unlock is called.
Return	WorkViewsResult

Required System Action

viewWorkList.

GetWorkViewsForResource

Retrieves a list of work views the calling resource has access to. In other words, it returns the views a user is owner or user of. Typically, this call is used when the user logs in to the client to show their list of views.

- The request must specify the number of items to retrieve.
- The response contains the unique entity ID of the work views and their details, for example their description, creation date and owner.

Action	getWorkViewsForResource
Parameter Notes	• startPosition: the position in the work list from which to start the page of results. The list is zero-based. To start at the first item, specify 0. • numberOfItems: the number of items from the work view to include.
Return	WorkViewsResult

Required System Action

viewWorkList

UnlockWorkListView

Unlock a locked work list view. A work view must be locked before it can be edited, using [GetWorkViewDetails](#). Once a work view has been edited (using the [EditWorkListView](#) operation), it can be unlocked using this operation.

- The request must specify the unique entityID of the work list view to be unlocked.
- The response contains the unique entity ID of the unlocked work view.

Action	UnlockWorkListView
Parameter	workListViewID : the unique entity ID of the work view.
Notes	
Return	WorkListIDResult

Required System Action

viewWorkList.

Using Wild Card Characters

The wild card characters '*' and '?' can be used when filtering on text attributes (work item name, description, category, and so on.) — wild card characters cannot be used with any other attribute type.

The '*' and '?' characters are evaluated as follows:

- '*' matches zero or more of any character. For example:
 - "bo*" matches anything beginning with "bo". For example, "bo", "bobbl", "boulton_anne", etc.
 - "s*s" matches anything beginning with "s" and ending with "s". For example, "ss", "sues", "sid_lewis", etc.
 - "*s" matches anything ending with "s". For example, "franks", "ls", "martha_lyons", etc.
- '?' matches any single character. For example:
 - "bill?" matches anything beginning with "bill", plus one more character. For example, "bills", "bill5", "billh", etc.
 - "c?ndy" matches "cindy", "candy", "c3ndy", etc.

Also note that:

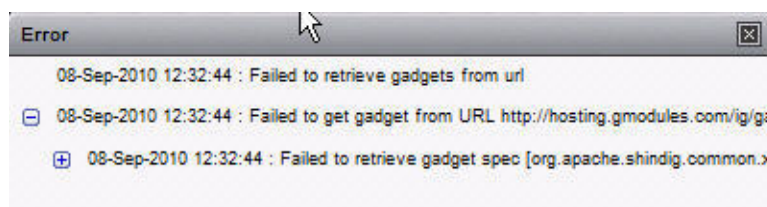
- all character matching is case sensitive.
- you can use multiple wild card characters.

Constraints for Contributing Gadgets in Openspace

When adding a gadget to Openspace, you must specify the URL of the gadget you want to add. The gadget URL must point to an OpenSocial gadget that complies with the OpenSocial Specification V0.8/V0.9. For more information, see [Web Application Development](#).

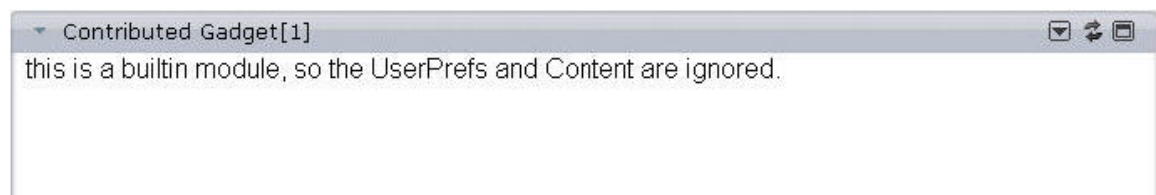
However, not all the client side customizations available to OpenSocial gadgets are supported by Openspace. This can result in three possible situations.

- The gadget is uploaded and rendered but as the gadget contains some client side customizations that are not supported, that functionality is not available in the uploaded gadget. In this scenario, you must make sure that the gadget specification that the gadget URL is pointing to does not use the following features:
 - Canvass View, Default and Preview views.
 - Gadget-side Pub Sub library.
 - OpenSocial API.
 - Tab support.
 - Locales feature.
 - Flash.
 - Skins.
 - Dynamic Height.
 - Mini Message.
 - User Preferences.
 - Capability JavaScript (CAJA - a subset of JavaScript).
- In this scenario, the gadget fails to upload and the following error message displays.



This error is caused because the gadget specification that the gadget URL is pointing to is using RSS feed gadget specifications. To rectify this, make sure that the gadget specification does not use RSS feed gadget specifications.

- In this scenario, if the gadget requires OAuth authentication support, the gadget is uploaded but fails to render and the following error message displays.



To rectify this, you should make sure that the gadget does not require support for OAuth authentication. For each scenario, you must make sure that the gadget specification that the gadget URL is pointing to is publicly accessible and is an HTTP URL.