TIBCO® Openspace Customization Guide

Software Release 4.2
August 2017

Document Update: December 2017

Two-Second Advantage®

TIBCO®

Important Information

Contents

TIBCO Documentation and Support Services

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website, mainly in HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, 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

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, visit http://www.tibco.com/services/support.
- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support portal at https://support.tibco.com.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to https://support.tibco.com. If you do not have a user name, you can request one by clicking Register on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the TIBCO Ideas Portal. For a free registration, go to https://community.tibco.com.

Openspace Configuration

You can customize Openspace, depending on your requirements. There are several properties files that you can use to perform customizations.

The two main `config.properties` files are for TIBCO Openspace and Accessible Openspace. They are located in the TIBCO ActiveMatrix BPM configuration directory.

The location of the BPM configuration directory is *`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment-adminservername`*`\datan\host\plugins\com.tibco.openspace.login_`*`n`*`\` where:

- *CONFIG_HOME* is the directory that stores the TIBCO BPM configuration information.
- *adminenvironment* is the name of your Administrator environment.
- *bpmenvironment* is the name of your BPM environment.
- *adminservername* is the name of your TIBCO Administrator server.
- data*n* where *n* is the version number of your Administrator environment.
- *n* is the current version number of the Openspace application files.

For example, the location of the Openspace `config.properties` files are as follows:

- TIBCO Openspace

  `C:\ProgramData\amx-bpm\tibco\data\tibcohost\Admin-AMX BPM-AMX BPM Server \data_3.2.x\host\plugins\com.tibco.openspace.login_1.6.0.013\resources \config.properties`.

- Accessible Openspace

  `C:\ProgramData\amx-bpm\tibco\data\tibcohost\Admin-AMX BPM-AMX BPM Server \data_3.2.x\host\plugins\com.tibco.os.a11y.app_1.1.1.00n\com\tibco\os\a11y\app \public\config.properties`

Configuring Authentication

The `authenticate` property in the `config.properties` files controls whether or not you must log in to Openspace after reloading your browser.

The `authenticate` property is in both the TIBCO Openspace and Accessible Openspace `config.properties` files. See Openspace Configuration Overview for information on locating and editing both the `config.properties` files.

There are a couple of reasons why your browser reloads.

- If your Openspace session times out.
- If you are opening a work item using the Openspace URL.

There are three modes you can set for the `authenticate` property.

- `0` - This mode uses the existing authentication in the session (if any). You may want to use this mode if you have provided an alternative login panel to authenticate and do not want to display the Openspace login panel.

  TIBCO recommends that the `authenticate` property is set to `0` on a system using single sign-on (SiteMinder or Kerberos). This means that users are not presented with the Login panel when exiting a SiteMinder or Kerberos session.
- `1` - This is the default option. This mode invalidates the current Openspace session. This means, if you reload your browser, the Openspace Login panel displays and you must log in to Openspace. However, if you have entered a username and password as part of the Openspace URL, then the Openspace Login Panel is not displayed, and therefore, you do not have to log back into Openspace.
- `2` - This mode always invalidates the current Openspace session when you reload your browser. This means that whether you have logged in via the Openspace Login panel or entered a username and password in the Openspace URL, the Openspace Login panel displays and you must log back into Openspace.

Prerequisites

TIBCO recommends you take a back up of the `config.properties` file before amending it.

Procedure

1. Open the `config.properties` file in a text editor.
2. Depending on your requirements, amend the `authenticate` property to `0`, `1` or `2`. The default is `1`.
3. Save and close the `config.properties` file.
4. Log out and log back into TIBCO Openspace for the changes to take effect.

Openspace Rebranding

You can rebrand TIBCO Openspace to match your corporate style. Integrating your corporate style will enhance your users experience with Openspace. Rebranding Openspace is achieved by replacing or amending the Openspace cascading style sheet (.css) files.

There are three areas that you can rebrand.

- Login
- Masthead
- Individual gadgets

TIBCO recommend that you take a back up of the .css files you want to replace/amend before you start rebranding.

The following table describes the .css files you can amend.

.css File	Feature	Location
`authentication.css`	Login	*`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment- adminservername`*`\datan\host\plugins\com.tibco.openspace.login_`*`n`*`\resources`
`tibco.css`	Masthead Gadget	`\\`*`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment- adminservername`*`\datan\host\plugins\com.tibco.openspace.login_`*`n`*`\resources\themesArchive\themes\tibco\`

If you have changed the display theme for a gadget, you must edit the .css file that corresponds to that theme, instead of `tibco.css`. For example, if you are using the *black* theme, edit `...\themesArchive\themes\black\black.css`.

See Openspace Configuration Overview for more information.

Login Rebranding

To rebrand the login, you must amend the classes and IDs that represent the specific areas of the login.

To rebrand the login, you must amend the `authentication.css` file. See Amending Openspace CSS Files. The following table describes the login classes and IDs that you can amend.

Login Classes and IDs that can be Used for Rebranding

Class/ID	Attribute
splashbackground	• Gradient • Color
splash	URL image

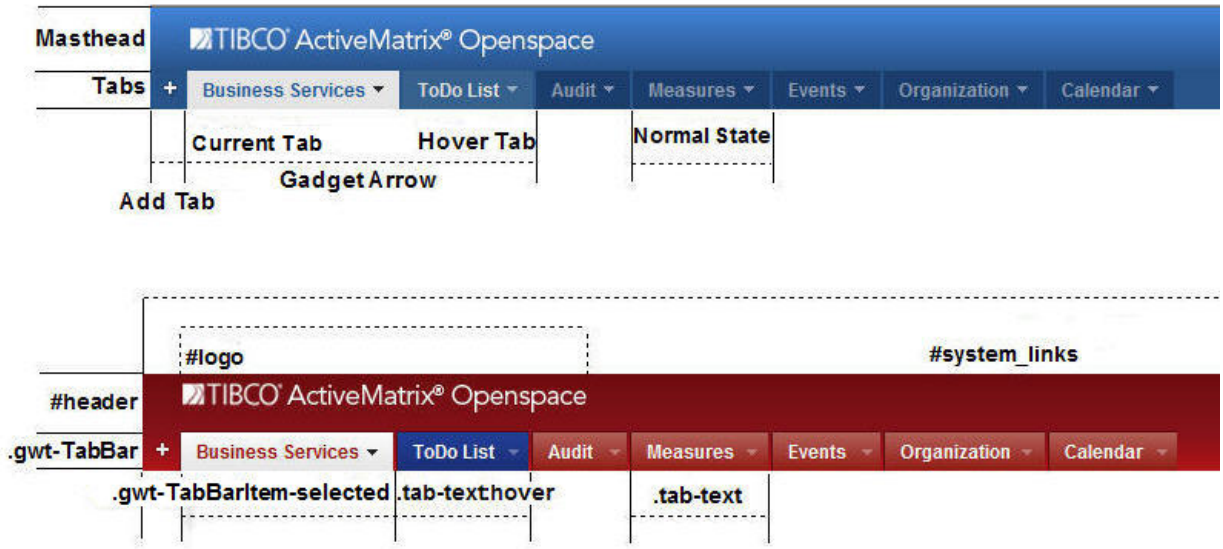
Class/ID	Attribute
splash_logo_amx	• URL image • Position • Padding
username, password, remember	Color/Size/Font Family
login button	Color/Size/Font Family

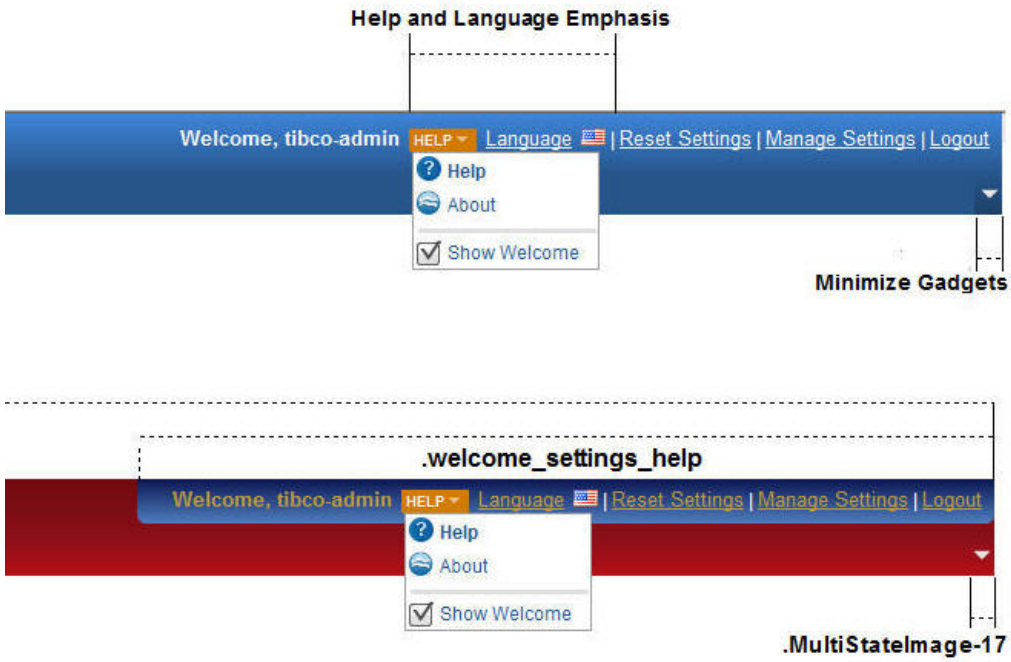
Note that:

- The main styling controls allow you to change the background and the inner background of the login form. As positioning is used, when adding new elements you may have to override program code with the `!important` statement in the css rule.
- For information on replacing the logo image, see Replacing Images.

Masthead Rebranding

To rebrand the masthead, you must amend the classes and IDs that represent the specific areas of the masthead.





To rebrand the masthead, you must amend the `tibco.css` file. (If you have changed the display theme for a gadget, you must edit the .css file that corresponds to that theme, instead of `tibco.css`.) See Amending Openspace CSS Files. The following table describes the masthead classes and IDs that you can amend.

Masthead Properties that can be Used for Rebranding

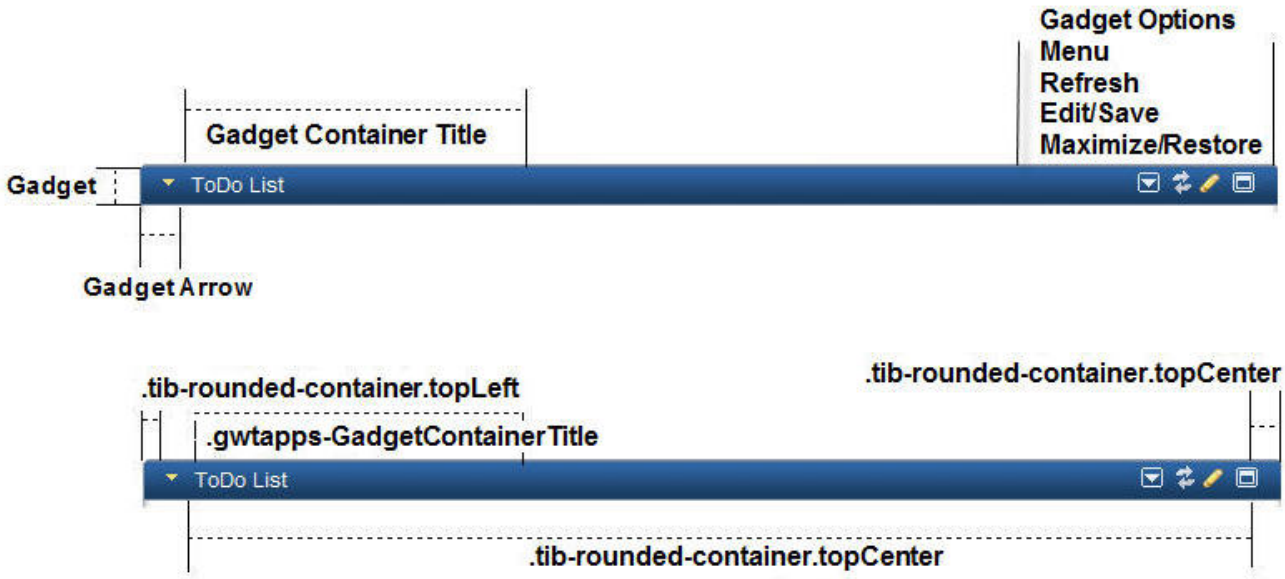
Class/ID	Attribute
header	• Gradient • Color
logo	URL image
system_links	• Color • Padding
welcome_settings_help	Color

Note that:

- To change the masthead background color, edit the background property. Gradient or solid colors can be used.
- Your logo dimensions should not exceed 30 pixels in height and be no longer than 300 pixels in width.
- To add a horizontal band to the masthead, set the background color of the `system_links` declaration. Decrease the padding depending on how thick you want the band to be.
- For information on replacing the logo image, see Replacing Images.

Gadgets Rebranding

To rebrand the gadgets, you must amend the classes and IDs that represent the specific areas of the gadgets that can be rebranded.



To rebrand the gadgets, you must amend the `tibco.css` file. (If you have changed the display theme for a gadget, you must edit the .css file that corresponds to that theme, instead of `tibco.css`.) See Amending Openspace CSS Files. The following table describes the gadget classes and IDs that you can amend.

Gadget Classes and IDs that can be Rebranded

Class/ID	Feature	Attribute
gwt-TabBar	Tab bar	• Background – Gradient – Color • Border – Top – Right
MultiStateImage-16	Add tab image	• Background – URL (image)
gwt-TabBarFirst	Decorative gutter	• Background – Color

Class/ID	Feature	Attribute
gwt-TabBarSelected tib-tab a.tab-text	Selected tab	• Background – Gradient – URL (image) – Color • Font – Color
gwt-TabBarItem tib-tab a.tab-ext:link	Normal tab	• Background – Gradient – URL (image) – Color • Font – Color • Border – Right
gwt-TabBarItem tib-tab a.tab-ext:hover	Hover tab	• Background – Gradient – URL (image) – Color • Font – Color • Border – Right
tib-tab	Tab definition	Border - Margin
MultiStateImage17	Minimize all gadgets	• Background – URL (image) – Color • Border – Top – Bottom

Background Rebranding

To change the background behind the tabs, use gwt-TabBar. You can change the color, add a gradient or enable borders if your branding is prominently white.

Set the background color to transparent or match the background color of the tab row. The decorative gutter also requires you set the background color accordingly.

Borders Rebranding

A border on a tab highlights to a user that an action can be performed. Set the margin to zero for the `gwt-TabBarItem` if using borders. This increases the visual separation.

Tab States Rebranding

Tabs have three states:

- selected
- normal
- hover

You can set a number of styling attributes. These include the font color, background gradient or solid color and borders to give the tab definition. Do not use the same color as the help background. TIBCO recommends using a contrasting font color.

Masthead Images Rebranding

You can rebrand the masthead images.

Currently there are two types of image displayed in the tab bar:

- the add new tab represented as a plus symbol
- the minimize all gadgets. This is positioned at the far right of the tabs and is represented as a down arrow.

You can also use the down arrow to reflect the three different tab states.

See Replacing Images for more information about replacing these images.

Replacing Images

You can replace the images used in Openspace. Images are used in several places in Openspace. For example, in the gadgets.

TIBCO recommends that you take a copy of every image you want to replace before you start rebranding.

Procedure

1. Update the style sheet with the new image name as references to images are called from the style sheet.
2. Replace the images in the locations described in the table below. Images are stored in either one or both of the following locations:

    - `\\`*`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment-adminservername`*`\datan\host\plugins\com.tibco.openspace.login_`*`n`*`\resources`

- `\\CONFIG_HOME\tibcohost\adminenvironment-bpmenvironment-adminservername\datan\host\plugins\com.tibco.openspace.login_n\resources\images`

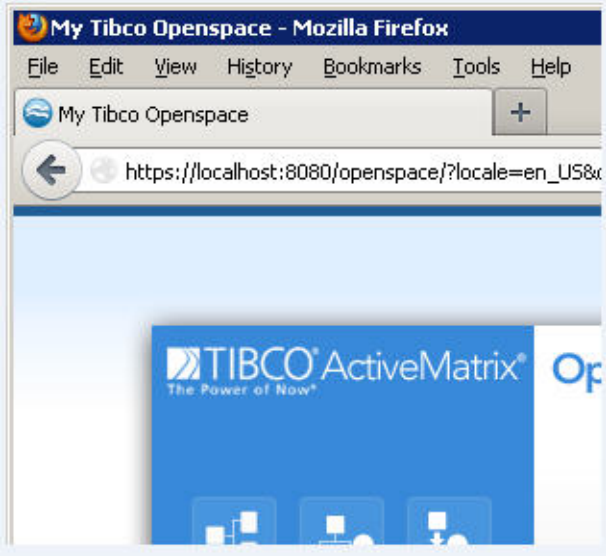
  See Openspace Configuration Overview for more information.

The following table describes the images you can replace and their paths. Where there are two locations for an image file, you must replace the image in both locations.

Image	Feature	Path
logo		• `resources\` • `resources\images\`
trans_add.png	Add tab	`resources\images\icons\`
tab_menu_view.png	Table menu	`resources\images\tabs\`
pulldown_arrow_current.png	Selected tab	• `resources\` • `resources\images\bg\`
pulldown_arrow_opacity_m.png	Normal and Hover tab	• `resources\` • `resources\images\bg\`
TIBCOOpenspace_logo_white.png		`resources\images\`

Customizing the Openspace Window Title

You can customize the Openspace window title.



- TIBCO recommend that you take a back up of the `config.properties` and `BannerPanelMessage.properties` file before amending them.
- If Openspace has been localized the `BannerPanelMessage.properties` may have a language and country code appended to it. For example, `BannerPanelMessage_ll_CC.properties`. See Localizing Openspace Overview on page 44.

Procedure

1. Open the `config.properties` file in a text editor. See Openspace Configuration Overview for the location of this file.
2. Change the `branding.type` property from `AMX` to `CUSTOM`. For example, `branding.type=CUSTOM`
3. Save and close the `config.properties` file.
4. Open the *`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment-adminservername`*`\data\host\plugins\com.tibco.openspace.login_`*`n`*`\resources\locale\BannerPanelMessage.properties` file in a UTF-8 text editor or an editor capable of editing UTF-8 files. For example, TIBCO Business Studio.
5. Change the `TIBCO Openspace` property to the new window title. For example, `customWindowTitle=My Custom Window Title`
6. Save and close the *`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment-adminservername`*`\data\host\plugins\com.tibco.openspace.login_`*`n`*`\resources\locale\BannerPanelMessage.properties` file.
7. Log out and reload the browser web page for the changes to take effect.

Configuring the Welcome Page

Openspace provides a welcome page that can be displayed when you first login. From the welcome page, you can perform activities, such as starting business services, viewing your work queues, performing administration tasks, or personalizing Openspace. By default, the welcome page is not displayed. To display the welcome page by default, amend the `initial.welcome` property in the `config.properties` file.

You can view the welcome page at any time by:

- clicking the Openspace logo.
- selecting **Help** and selecting the **Show Welcome** check box. If you select the **Show Welcome** check box, the welcome page is displayed every time you login.

This will override the `initial.welcome` property. However, if you click **Reset Settings**, Openspace reverts back to using the `initial.welcome` property setting.

For more information about the location of the `config.properties` file, see Openspace Configuration Overview for more information.

Prerequisites

TIBCO recommends you take a back up of the `config.properties` file before amending it.

Procedure

1. Open the `config.properties` file in a text editor.
2. Depending on how the `initial.welcome` property is set, either type `true` if you want the welcome page to display or `false` if you do not.
3. Save and close the `config.properties` file.

4. Log out and log back into TIBCO Openspace for the changes to take effect.

Configuring Time Allowed for Gadgets to Load

The `life.cycle.timeout` property in the `config.properties` file controls the amount of time that Openspace should wait for all the gadgets to load before it times out.

When Openspace is loading the gadgets, it waits no longer than the time specified by `life.cycle.timeout` for all the gadgets to load. This means that if one of the gadgets is broken, Openspace doesn't keep waiting for it to load but instead returns control to the user interface.

Prerequisites

TIBCO recommends you take a backup of the `config.properties` file before amending it. For the location of the `config.properties` file, see Openspace Configuration Overview.

Procedure

1. Open the `config.properties` file in a text editor.
2. Depending on your requirements, amend the `life.cycle.timeout` property to the maximum number of milliseconds that you want Openspace to wait for all the gadgets to load. The default is 20000 ms or 20 s. Do not include any commas in the number.
3. Save and close the `config.properties` file.
4. Log out and log back into TIBCO Openspace for the changes to take effect.

Configuring the TIBCO Openspace Date Time Format

Date and time stamps are often used in Openspace. For example, the gadget status bar sometimes shows a timestamp and you can specify a start date and end date in the Process Instances Measures gadget.
There are two files where the date and time format can be configured for Openspace.

- `TranslatorMessages_ll_CC.properties` file. The date and time for the majority of the gadgets in Openspace is configured using this file.
- `CommonMessages_ll_CC.properties` file. Currently, this file is used by the Data Views gadget only.

where:

- *ll* is a lowercase, two-letter ISO 639 language code. For a list of language codes, visit the following web site: http://www.loc.gov/standards/iso639-2/langhome.html.
- *CC* is an uppercase, two-letter ISO 3166 country code. For a list of country codes, visit the following web site: http://www.iso.org/.

These files are located in the TIBCO ActiveMatrix BPM configuration directory.

- For example, the `CommonMessages_ll_CC.properties` file is available in `C:\Program Data\amx-bpm\tibco\data\tibcohost\Admin-AMX BPM-AMX BPM Server\data_3.2.x\host\plugins \com.tibco.os.dataview_1.0.0.017\resources\dataview\locale`.
- For example, the `TranslatorMessages_ll_CC.properties` file is available `C:\ProgramData\amx-bpm\tibco\data\tibcohost\Admin-AMX BPM-AMX BPM Server\data_3.2.x\host\plugins \com.tibco.openspace.login_1.7.0.036\resources\locale`.

See "Openspace Configuration Overview" in the *TIBCO Openspace Customization Guide* for information about the location of these files.

TIBCO recommends that you take a back-up of these files before you amend them.

Both the files contain the core messages that are used by most of the gadgets across Openspace. You must configure both a `TranslatorMessages_ll_CC.properties` file and a `CommonMessages_ll_CC.properties` file for each language you want Openspace to support.

The format of both files is the same. The patterns making up the various `dateTimePattern` properties are made up of meta characters as defined in Google Web Toolkit. The table below shows the meta characters you can use.

Symbol	Meaning	Presentation	Example
G	era designator	Text	AD
y	year	Number	1996
M	month in year	Text or Number	July (or) 07
d	day in month	Number	10
h	hour in am/pm (1-12)	Number	12
H	hour in day (0-23)	Number	0
m	minute in hour	Number	30

Symbol	Meaning	Presentation	Example
s	second in minute	Number	55
S	fractional second	Number	978
E	day of week	Text	Tuesday
a	am/pm marker	Text	PM
k	hour in day (1-24)	Number	24
K	hour in am/pm (0-11)	Number	0
z	time zone	Text	Pacific Standard Time
Z	time zone (RFC 822)	Number	-0800
v	time zone (generic)	Text	Pacific Time
’	escape for text	Delimiter	’Date=’
"	single quote	Literal	’o’ ’clock’

The following table shows some example patterns.

Example	Formatted Text
`"yyyy.MM.dd G 'at' HH:mm:ss vvvv"`	1996.07.10 AD at 15:08:56 Pacific Time
`"EEE, MMM d, ''yy"`	Wed, July 10, '96
`"h:mm a"`	12:08 PM
`"hh 'o''clock' a, zzzz"`	12 o'clock PM, Pacific Daylight Time
`"K:mm a, vvv"`	0:00 PM, PT
`"yyyyy.MMMMM.dd GGG hh:mm aaa"`	01996.July.10 AD 12:08 PM

See http://www.gwtproject.org/javadoc/latest/com/google/gwt/i18n/client/DateTimeFormat.html for more information.

If a `dateTimePattern1` property is not defined in the locale specific `TranslatorMessages_`*`ll_CC`*`.properties` file then the property value falls back to the property defined in the `TranslatorMessages.properties` file.

The `common=`*`filename`* property in the data views `config.properties` file configures the Data Views gadget to use the date time patterns that are available in the `CommonMessages_`*`ll_CC`*`.properties` file. For example, if your filename is `exampleMessages_`*`ll_CC`*`.properties` then the `common` property must be `common=example`.

The Logger gadget enables you to quickly identify any words that are not translated in a locale pack. Specify the level of logging in the Logger gadget at INFO level, and an error displays if there are any words that are not translated. Words that are not translated default to American English. See *Openspace User's Guide* for more information.

Adding New Themes

Themes determine the colors of the title bar and gadget windows in Openspace. From the Themes Selector in Openspace, users can select different themes for some or all of their gadgets, depending on their requirements.

Selecting different themes for different gadgets is a useful way of distinguishing between them. See "Selecting Themes" in the *TIBCO Openspace User's Guide* for more information

By default, four themes are available in the Themes Selector when you first log into Openspace. However, you are allowed six themes in Openspace in total. In other words, you could have six themes in your Themes Selector, if required. You can use the theme properties in the `config.properties` file to determine which themes are available in the Themes Selector. You can also add new themes and make them available in the same way.

The four available themes are:

- `theme.a=tibco`
- `theme.b=default`
- `theme.c=black`
- `theme.d=red`

There can only be a maximum of six themes available in the Themes Selector. Themes must be numbered sequentially and there can be no gaps in the numbering.

You can make all four themes available, add new themes or restrict the number of themes available by commenting out the ones you do not require.

As themes must be numbered sequentially with no gaps in the numbering, this means you cannot comment out `theme.a` and enable `theme.e`. You must enable `theme.a`, then `theme.b`, then `theme.c` and so on.

If no themes are enabled then, `theme.b=default` is used.

To add a new theme, create your new `.css` file, and save it, along with any images, to the `themesArchive` folder. For example, `C:\ProgramData\amx-bpm\tibco\data\tibcohost\Admin-AMX BPM-AMX BPM Server\data_3.2.x\host\plugins \com.tibco.openspace.login_1.6.0.013\resources\themesArchive\`*`newTheme`*.

- You could use the `default.css` as a template and amend it accordingly.
- There are 6 themes available in the `themesArchive` folder. However, silver and purple are deprecated themes. You must not base your custom themes on these as they are not maintained.
- There must be no spaces in your theme name.
- Create a new *`theme_icon`*`.gif` as a thumbnail to distinguish the new theme from the other themes.

Once you have created your new theme, amend the next available theme property in your list in the `config.properties` file. For example, if `theme.a` and `theme.b` are enabled, the next theme property is `theme.c=`*`newTheme`*. As only six themes can be available in the Themes Selector at any time, make sure you have the correct number of themes commented out. In other words, if you have six themes currently available for selection and you want to add a new one, you must uncomment one of the existing themes before your new theme can be available in the Themes Selector.

See Openspace Configuration Overview for information about the `config.properties` file.

Prerequisites

- Make sure you have created your new `.css` file, and saved it, along with any images, to the `themesArchive` folder.

- TIBCO recommends you take a back up of the `config.properties` file before amending it.

Procedure

1. Open the `config.properties` file in a text editor.
2. (Optional) If you have created a new theme, amend the next available theme property in your list. For example, `theme.e=`*`newTheme`*
   If all six themes are enabled, amend one of the existing theme properties.
3. Save and close the `config.properties` file.
4. Log out and log back into TIBCO Openspace for the changes to take effect.

Layout Customization

You can customize the layouts that are available in Openspace. By default, there are four built-in layouts in Openspace and only these are available for use. There are also some sample layouts defined in the `config.properties` file. You can customize the built-in layouts, make the sample layouts available as they are or customize them, or create new layouts.

If you do change your layout in Openspace so that it is using a layout that displays gadgets in different widths (if you amend your layout by dividing the tab into 3 boxes instead of 2 boxes, for example), and then drag and drop a gadget to a new box, the gadget's width does not resize to fit the box correctly. This applies whether the gadget must increase or decrease in size. To work around this, resize your browser window, either by maximizing or minimizing or using the arrows, and the gadget resizes correctly.

In Openspace, you can think of layouts as a grid of cells, divided into columns and rows. When defining a layout, you must specify the number of boxes, whether the boxes are horizontal or vertical, and the number of cells that the boxes should span. Boxes are represented by a number and must be allocated left-to-right and top-to-bottom. For example:

- A layout defined as (1,2) means that there are two boxes in the layout and the boxes are vertical. Each box spans 2 cells. In other words, each box has 50% of the available width.

- A layout defined as (1,1,1,2) means that there are two boxes and the boxes are vertical. Box 1 spans 3 cells (75% of the width) and box 2 spans 1 cell (25% of the width).

- A layout defined as (1,1) (2,3) means that there are three boxes and the boxes are horizontal. Box 1 spans 2 cells (100% of the width). Boxes 2 and 3 span 1 cell each. In other words, each box has 50% of the width available.

Although you can have as many boxes as you want, TIBCO recommends that you have no more than four, as the size of the boxes are too small when displayed in Openspace.

By default, there are four built-in layouts available in Openspace. There are also some sample layouts that you can either make available as they are or customize according to your requirements (see Layout Properties) or you can create new layouts. There are some rules that you must follow when customizing layouts:

- For the built-in layouts, the number of boxes must remain the same. You cannot add or remove boxes. You can add and remove boxes in the sample layouts or in new layouts.
- In the definition spanned cells making up each box must be either horizontal or vertical but not both (because gadgets do not dynamically change height. This is because they only have 3 possible heights:

  - minimized
  - default fixed
  - maximized
- The box numbers must be allocated left-to-right, top-to-bottom.
- Box numbers start at 1 and the top-left corner must be allocated to 1.

Layout Properties

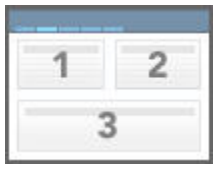
Layouts are defined in the `config.properties` file. A layout needs to be defined and allocated a layout ID. Then, depending on the layout, it must be enabled by adding the layout ID to the `layout.ID's` property.

There are some layout ID's already defined in the `config.properties` file. There are built-in layout ID's that are available in Openspace by default. There are sample layouts that have a layout ID defined. You can also create your own layout ID's. Layout ID's are in the format `layout.`*`n`*` = (`*`x`*`)` where

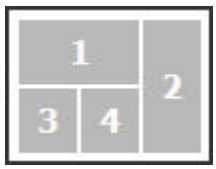
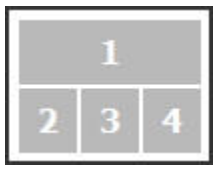
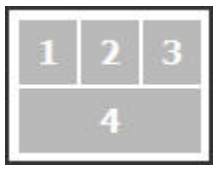
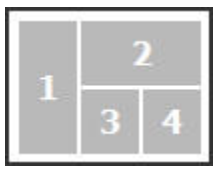
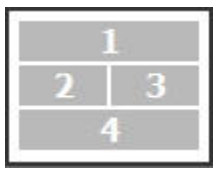
- *n* is the ID of the layout. This can be whatever you want.
- *x* is the number of boxes you require. You can have more than one box, and boxes can be horizontal as well as vertical. See the layout IDs below.

To enable sample layout IDs and new layout IDs, add the layout ID's property. By default, the layout.ID's property is configured as follows: `layout.IDs = 1a,2a,3b,3c`

The following built-in layouts are defined:

Layout ID	Layout
`layout.1a = (1)`	
`layout.2a = (1,2)`	
`layout.3b = (1,1),(2,3)`	
`layout.3c = (1,2),(3,3)`	

The following custom sample layouts are defined:

Layout ID	Layout
`layout.3a = (1,2,3)`	
`layout.4a = (1,2,3,4)`	
`layout.4b = (1,1,2),(3,4,2)`	
`layout.4c = (1,1,1),(2,3,4)`	
`layout.4d = (1,2,3),(4,4,4)`	
`layout.4e = (1,2,2),(1,3,4)`	
`layout.4f = (1,1),(2,3),(4,4)`	

Enabling Sample/New Layouts

To make the sample or new layouts available in Openspace, you must add the layout ID to the `layout.IDs` property in the `config.properties` file.

If you are adding a new layout, you must define a new layout ID to the `config.propeties` file. See Openspace Configuration Overview for more information.

If you amended a built-in layout you do not have to do anything. This is because the built-in layouts are enabled by default. Therefore, the changes take effect as soon as you log out and log in again to Openspace.

Prerequisites

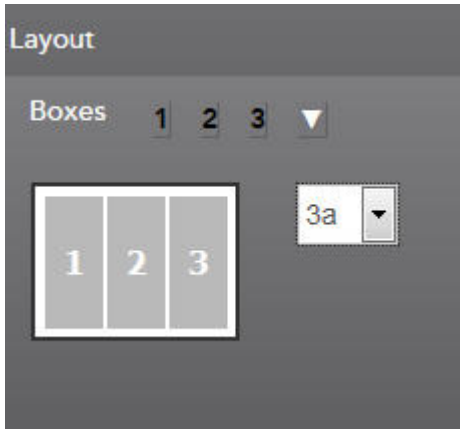
TIBCO recommends you take a back up of the `config.properties` file before amending it.

Procedure

1. Open the `config.properties` file in a text editor.
2. If you want to create a new layout, define a new layout ID. For example, `layout.4g = (1,1,1,2).`
3. Amend the `layout.IDs` property as follows:
   - If you want to make a sample layout available, you must add the sample layout ID to the list of layout IDs. For example, `layout.IDs = 1a,2a,3b,3c,3a,4a`
   - If you have created a new layout, you must add the new layout ID you have created to the list of layout IDs. For example, `layout.IDs = 1a,2a,3b,3c,4g`
4. Save and close the `config.properties` file.
5. Log out and log back into TIBCO Openspace for the changes to take effect.

What to do next

If built-in layouts have been customized, the new built-in layouts are automatically available the next time you log into Openspace. If you have enabled a sample layout or a new layout, the next time you log in to Openspace, a **Layout Selector** box is displayed that enables you to select the layout you require from a drop-down list.



Openspace Features Lockdown

You can customize who has access to specific TIBCO Openspace features.

When you first login to TIBCO Openspace, all users have access to all Openspace features. However, you can customize access to specific Openspace features by configuring the lockdown properties in the `config.properties` file. See config_properties_file for the location of this file.

TIBCO recommends that you take a backup of the `config.properties` file before amending it.

Locking down features means that the feature is NOT available to one or more users, depending on how you have configured your lockdown properties.

Lockdown Properties

The Lockdown properties are located in the `config.properties` file.

The following types of properties are included in the file.

- `lockdown.`*n* where *n* is a numerical value. You can associate the lockdown property with a system action and assign it to one or more Openspace features. This means that only users who have been granted those privileges can view the Openspace features you have specified, for example, `lockdown.1=WSB,applicationconfiguration`. See Configuring Lockdown System Actions.
- `lockdown.`*x* where *x* is a specific Openspace feature. For example `lockdown.showResetSettings=TRUE`. These lockdown properties can be set to TRUE or FALSE. This enables you to configure access to one or more Openspace features for all users on your system. For example, you may never want any users to have access to the User Preferences Panel. In this case, you can set the `lockdown.allowPrefPanel` to `TRUE`. See Openspace Feature Lockdown Properties.

  When configuring lockdown properties, no users are exempt. For example, lockdown properties apply to the tibco-admin user in the same way as any other user.

Configuring Lockdown System Actions

You can configure lockdown system actions in the Organization Browser to allow different users access to different Openspace features.
There are some system actions that you can use to associate one or more Openspace features to.

- openspaceFeatureSetA
- openspaceFeatureSetB
- openspaceFeatureSetC

However, you can associate any system actions with any Openspace features and add as many as you require. For example, you may want different types of users to have access to different Openspace features. In an insurance company, for example, you may have the following types of users; customer service representatives, managers, administrators. The following table shows how you might want to lockdown Openspace features depending on these types of users.

Openspace Features	Customer Service Representative	Managers	Administrators
`showResetSettings`	Disable	Enable	Enable
`showLogoutButton`	Disable	Enable	Enable

Openspace Features	Customer Service Representative	Managers	Administrators
showAddTab	Disable	Enable	Enable
allowPrefPanel	Disable	Enable	Enable
showFullGadgetMenu	Disable	Disable	Enable
allowMinRestoreAll	Enable	Enable	Enable
enableWelcomePage	Disable	Disable	Disable
max10Tabs	Enable	Disable	Disable

Procedure

1. From Openspace, go to the Organization Browser. Assign the following system actions to the different types of users:

Users	System Actions
Customer Service Representatives	OS, openspaceFeatureSetA
Managers	OS, opensapceFeatureSetA OS, opensapceFeatureSetB
Administrators	WSB, applicationConfiguration OS, opensapceFeatureSetA OS, opensapceFeatureSetB OS, opensapceFeatureSetC

2. Associate the system actions with Openspace features as shown in the table below:

System Actions	Openspace Features
OS, openspaceFeatureSetA	allowMinRestoreAll
OS, openspaceFeatureSetB	showResetSettings showLogoutButton showAddTab allowPrefPanel
OS, openspaceFeatureSetC	max10Tabs
WSB, applicationConfiguration	showFullGadgetMenu

3. Open the `config.properties` file (see Openspace Configuration Overview for the location of this file) in a text editor and amend the lockdown properties as follows:

```
lockdown.1=OS,openspaceFeatureSetA
lockdown.2=OS,openspaceFeatureSetB
lockdown.3=OS,openspaceFeatureSetC
lockdown.4=WSB,applicationConfiguration
lockdown.showResetSettings=lockdown.2
lockdown.showLogoutButton=lockdown.2
lockdown.showAddTab=lockdown.2
lockdown.allowPrefPanel=lockdown.2
lockdown.showFullGadgetMenu=lockdown.4
lockdown.allowMinRestoreAll=lockdown.1
lockdown.enableWelcome=false
lockdown.max10Tabs=!lockdown.3
```

In this example, `lockdown.4` is granted the `WSB, applicationConfiguration` system action. The `WSB, applicationConfiguration` system action is typically granted to Administrators who are allowed privileged functions, such as Manage Settings. In this example only the Administrators would get the full gadget menu. Out-of-the-box, only the tibco-admin user is granted this system action.

Openspace Feature Lockdown Properties

The following table describes the properties that represent the Openspace features you can customize access to.

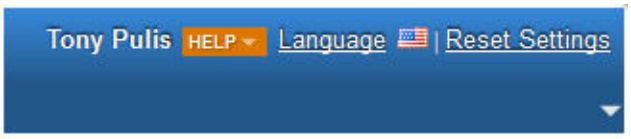
Property	Description
`lockdown.showResetSettings`	Configures whether the **Reset** button on the main menu is displayed.
`lockdown.showLogoutButton`	Configures whether the **Logout** button on the main menu is displayed.
`lockdown.showAddTab`	Configures whether you can add a new tab.
`lockdown.allowPrefPanel`	Configures whether the **User Preferences** panel is displayed.
`lockdown.showFullGadgetMenu`	Configures whether you can refresh, minimize, maximize, remove or toggle the status of gadgets.
`lockdown.allowMinRestoreAll`	Configures whether you can select the **Minimize All** or **Restore All** button.
`lockdown.enableWelcome`	Configures whether the Welcome page is displayed.
`lockdown.max10Tabs`	Configures whether you can add more than 10 tabs in Openspace.

You can set the values as follows:

- set the value to a lockdown property that has been associated with a system action. For example, `lockdown.1`, `lockdown.2` and so on. See Configuring Lockdown System Actions for more information.
- set the value to `TRUE` or `FALSE` if the setting for the Openspace feature must apply to all users on the system.
  - `TRUE` means the feature is available.
  - `FALSE` means the feature is not available.
- insert an `!` before the value if you want to invert the logic. For example, you may wish to have a system action that enables complete access to Openspace that can be granted to administrative users. You may wish to associate this system action with the `lockdown.1` property. In this situation, for the `lockdown.max10Tabs` property, you would specify a value of `!lockdown.1`. This means that users who have the `lockdown.1` privilege *can* create more than 10 tabs.

For example:

- Setting `lockdown.showLogoutButton`to `FALSE` means **Logout** is not available as shown below.



- Setting `lockdown.showAddTab` to `FALSE` means new tabs cannot be added as shown below.

Configuring the Lockdown Properties

You can customize access to specific Openspace features by configuring the lockdown properties in the `config.properties` file.

See Openspace Configuration Overview for the location of this file.

TIBCO recommends that you take a backup of the `config.properties` before amending it.

To configure the lockdown properties, do the following:

Procedure

1. Open the `config.properties` file in a text editor.
2. Uncomment and amend the lockdown properties. See Openspace Feature Lockdown Properties.
3. Save and close the `config.properties` file.
4. Log out and log back into TIBCO Openspace for the changes to take effect.

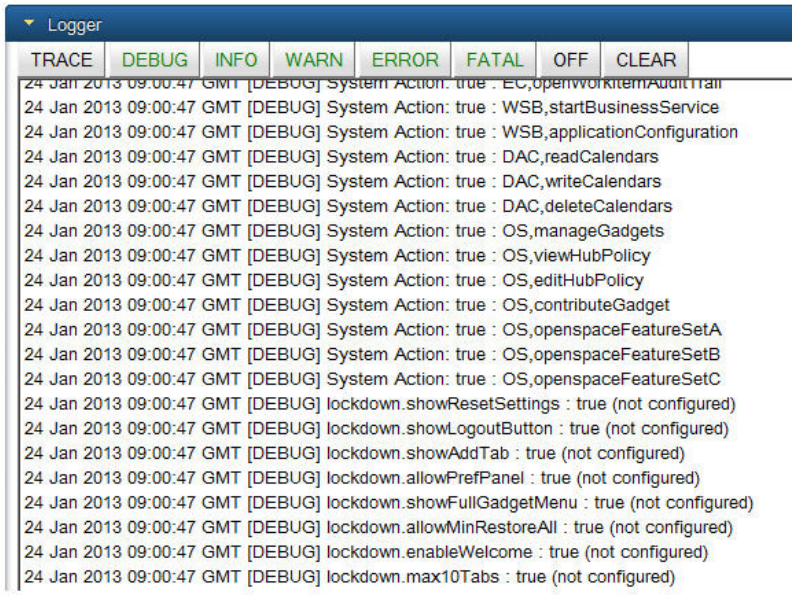
Checking Lockdown Properties

You can use the Logger gadget to check what system actions are applied for the user currently logged into Openspace and to check the lockdown properties for `lockdown.`*n*, if configured.

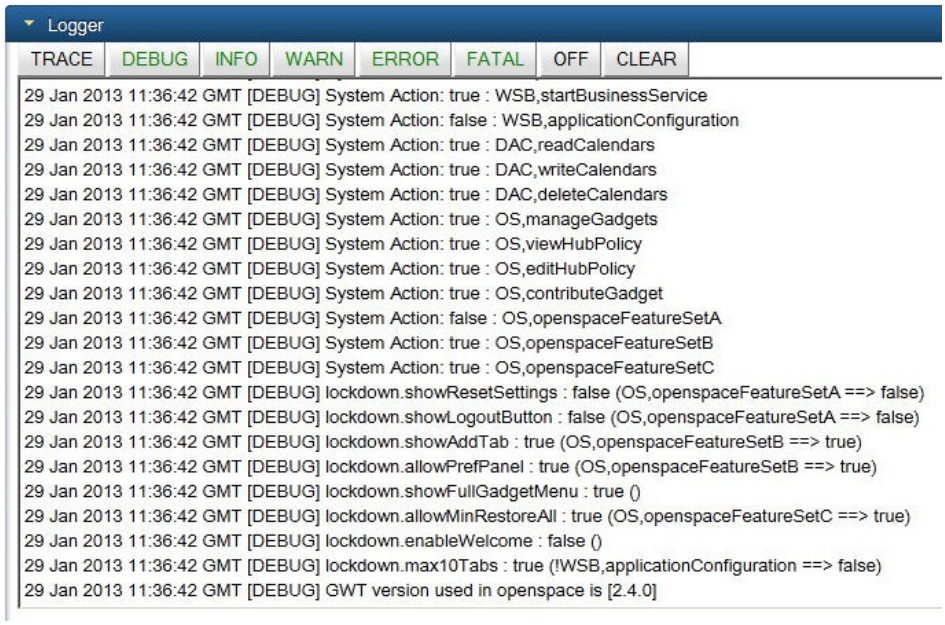
Procedure

1. From the Logger gadget, click the pen icon ( ) and select **DEBUG** as the **Default Log Level**.
2. Click the pen icon ( ) again to close the dialog.
3. Look for the part of the trace that shows the evaluations of the lockdowns.

   The screenshot below shows the log of a user for whom no lockdown properties are configured. When you first install Openspace, no lockdown properties are configured.



The following screenshot shows the log of a user for whom lockdown properties have been configured.



Displaying Work Item Attributes

TIBCO ActiveMatrix BPM allows you to define work item attributes and work item attribute facades at design-time in TIBCO Business Studio. Work item attributes can be populated by a script in a process. They can be used to contain data associated with a work item. Work item facades allow you to define different display names for your work item attributes. You can configure how Openspace displays the work item attribute names. By default, the default work item attribute names display, for example, Attribute 1, Attribute 2 and so on.

Work item attributes are available in the following:

- In Work Views:
  - Use the Column Selector to view work item attributes in your work views.
  - Use the Filter and Sort dialogs to include work item attributes in filter and sort criteria.
  - Use **Search** to search on work item attributes.
- In Process Views, work item attribute names display in the **Event Attribute** pane on the **Audit** tab.
- In Event Views, work item attribute names display in the list of event attributes.
- If you are using JMS publication, any activities performed in Openspace that uses work item attributes are published to JMS.

You can configure how Openspace displays the work item attribute names using the `workViews.useAttribDisplayNames` attribute in the `config.properties` file.

Out of the box, the default setting is `Use System Default`. This means the default work item attribute names display, for example, Attribute 1, Attribute 2 and so on. See *TIBCO Openspace User's Guide* for more information.

You can use the `workViews.useAttribDisplayNames` attribute to configure how the work item attribute names are displayed in Openspace.

If the `workViews.useAttribDisplayNames` attribute is set to `true`, then work item attribute names that are defined in the deployed work item facade are displayed. Work item facades are defined at design-time in TIBCO Business Studio. See *TIBCO Business Studio Modeling Guide* for more information.

If the `workViews.useAttribDisplayNames` attribute is set to `false`, then the default work item attribute names are displayed, for example, Attribute 1, Attribute 2 and so on.

Users can override this setting in the Work Views gadget by configuring the **use attribute display names** option. See *TIBCO Openspace User's Guide* for more information.

Prerequisites

TIBCO recommends you take a back up of the `config.properties` file before amending it.

Procedure

1. Open the `config.properties` file in a text editor.
   See Openspace Configuration Overview for information on the location of the `config.properties` file.
2. Amend the `workViews.useAttribDisplayNames` attribute.
3. Save and close the `config.properties` file.
4. Log out and log back into TIBCO Openspace for the changes to take effect.

Configuring the Location of the Help Files

Selecting **Help** in TIBCO Openspace displays the Openspace help files that are located on the TIBCO Product Documentation site. If your users have internet access, no further configuration is required. However, if your users do not have internet access, you can configure Openspace to use a different location for the Openspace help.

To do this, you must download the TIBCO ActiveMatrix BPM Help located on the TIBCO Documentation site to your system and host the help on another site.

If you download and store the Openspace documentation, you may not be aware of changes that are made to those documents on the TIBCO Documentation web site. Occasionally, out-of-cycle (OOC) updates are published for some documents, which means that they are updated between software releases. To be notified when OOC updates occur, you can subscribe to RSS feeds by using the **Subscribe**  button on the "TIBCO ActiveMatrix BPM" page of the TIBCO Documentation web site.

Note, however, you cannot subscribe to RSS feeds for only the Openspace documentation; this notifies you of changes to any TIBCO ActiveMatrix BPM documentation.

Procedure

1. Download the TIBCO ActiveMatrix BPM documentation from the TIBCO Documentation web site.
   You cannot download just the Openspace documentation; it is bundled with all of the TIBCO ActiveMatrix BPM documentation.
   a) Open a browser and go to the following URL:
      `https://docs.tibco.com/products/tibco-activematrix-bpm`
      This URL always takes you to the documentation for the most recently released version of TIBCO ActiveMatrix BPM. If you want the documentation for an earlier version, click the appropriate tab on the page.
   b) Click **Download All**.
      This downloads both the HTML and PDF documentation.
2. Unzip the file that you downloaded and host the HTML files on an internal web server.
   The HTML help is located in the `bpmhelp` folder; the HTML help is accessed using `index.html`.
3. Open the TIBCO Openspace `config.properties` file in a text editor.
   For information about the location of the `config.properties` files for both TIBCO Openspace and Accessible Openspace, see Openspace Configuration Overview.
4. Specify the new URL for the Openspace help files in the `help.context` property.
   This must be the full URL to the help files, for example:
   `http://<ip_address>:8080/doc/html/bpmhelp/`
5. Save and close the `config.properties` file.
6. Repeat steps 3-5 for Accessible Openspace.
7. Log out and log back into TIBCO Openspace for the changes to take effect.
   If your changes do not take effect, you may also need to clear the browser cache.

Configuring Custom Functionality

Openspace allows you to call your own customized scripts when various events are triggered in Openspace. For example, you may want users to log into a third-party system before automatically logging into Openspace or you may want them to be redirected to a customized URL when logging out of Openspace.

If you are redirecting to a URL when logging out of Openspace and you press F5 (Refresh) then some browsers (specifically, Chrome) may prevent the new window from opening. This is because some browsers do not allow a new window to be opened when they are in the process of reloading the original URL. Therefore, it is not recommended that you refresh your browser when logging out of Openspace.

To achieve this, Openspace has provided a script called `callouts.js` that you can edit to call your customized scripts. The script is located in *`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment-adminservername`*`\datan\host\plugins\com.tibco.openspace.login_`*`n`*`\resources\callouts\callouts.js`. See Openspace Configuration Overview for more information about the `config.properties` file.

There are four properties that allow you to configure when the `callouts.js` script is run. The script is automatically run when one or more of the properties are enabled. The properties are stored in the `config.properties` file.

The four properties are:

- `hook.onLoadOpenspace`. The script runs when Openspace loads.
- `hook.onLoginOpenspace`. The script runs when a user logs in to Openspace.
- `hook.onLogoutOpenspace`. The script runs when a user logs out of Openspace.
- `hook.onUnLoadOpenspace`. The script runs when Openspace is unloading.

There is also a `hook.onCollateStrings` property. Please see Configuring Sort Order for more information.

An example of the `callouts.js` script is shown below:

```
// Copyright© 2005-2013, TIBCO Software Inc

    function onLoadOpenspace()
    {
        if(document.getElementById) {
            window.alert = function(txt) {
                // override window.alert() and instead do INFO level log in
Openspace
                openspaceLog(txt);    // can also use openspaceLogN(txt,level);
where level 4==WARN, 2==ERROR, 1==FATAL, 8==INFO, 16=DEBUG, 32==TRACE
etc.
            }
        }

        window.alert("a) javascript onLoadOpenspace");
    }

    function onLoginOpenspace(user)
    {
        window.alert("b) javascript onLoginOpenspace " + user);
    }

    function onLogoutOpenspace()
    {
        window.alert("c) javascript onLogoutOpenspace" );
    }

    function onUnLoadOpenspace(timedOut, sessionInvalidated, urlReload)
    {
        window.alert("d) javascript onUnLoadOpenspace timedOut=" + timedOut + "
```

```
sessionInvalidated=" + sessionInvalidated + " urlReload=" + urlReload );

        return urlReload;
    }

    function onCollateStrings(locale, a, b, context)
    {
        var result = "0";

        var ret = a.localeCompare(b);

        if (ret < 0)
        {
            result = "-1";
        }
        else
        {
            if( ret > 0)
            {
                result = "1";
            }
        }
        window.alert("e) javascript onCollateStrings (" + context + ") locale=" +
locale + " a=" + a + " b=" + b+" result="+result );

        return result;
    }
```

TIBCO recommends that you take a backup of the `config.properties` file before amending it.

Procedure

1. Open the `config.properties` file in a text editor.
2. Uncomment and amend the hook properties. Initially, all the hook properties are commented out and set to `false`. Depending on which hook property you want, amend `false` to `true` to enable it.

   For example, if you wanted your users to be redirected to `www.tibco.com` when they are logging out of Openspace, the `hook.onLogoutOpenspace` property should be amended as follows: `hook.onLogoutOpenspace=true.`

   By default, the callouts are logged as INFO messages in the Logger Gadget.
3. Save and close the `config.properties` file.
4. Log out and log back into TIBCO Openspace for the changes to take effect.

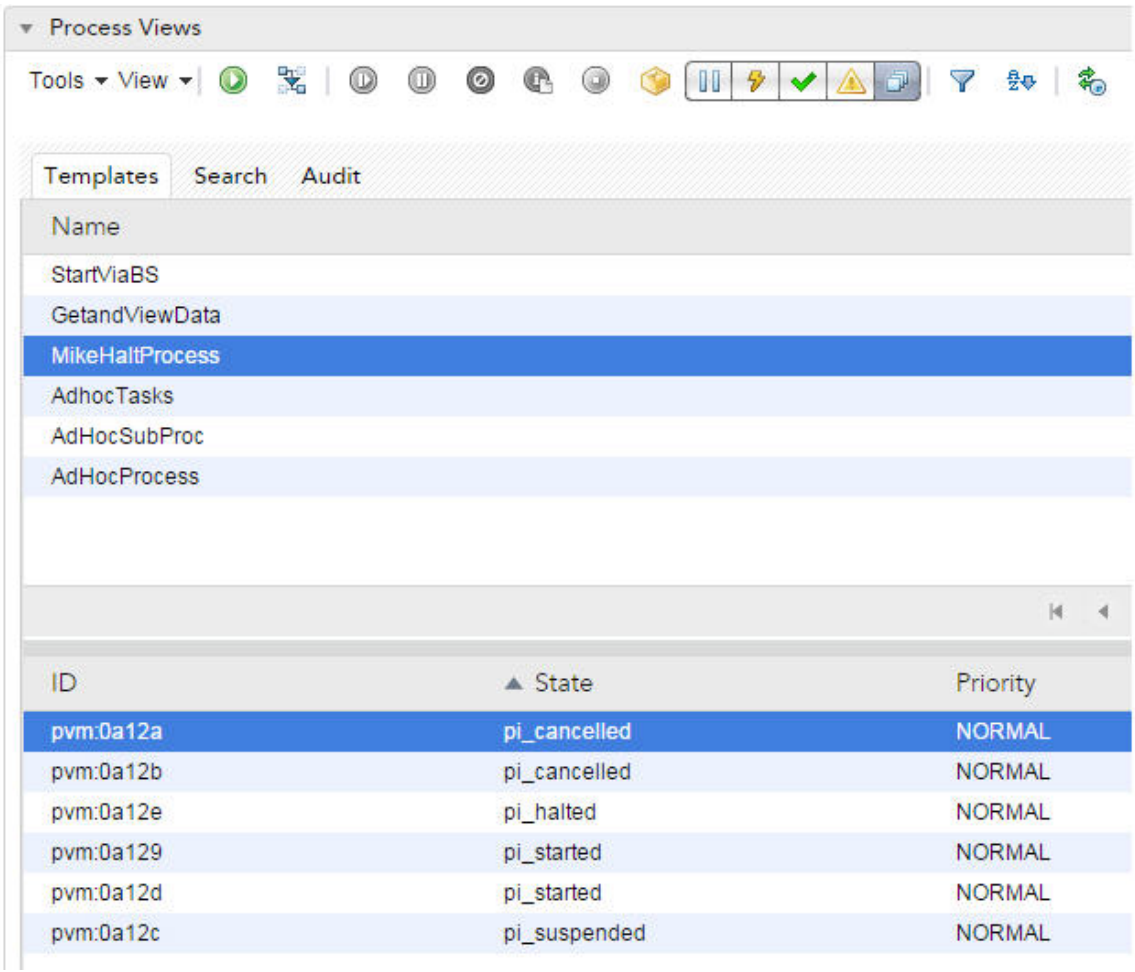
Configuring Sort Order

Sort order relates to the order of all characters in the unicode character set (for example, numeric, alphabetic, symbol, and so on). You may wish to customize the sort order that Openspace uses, for example, because of locale-specific conventions.

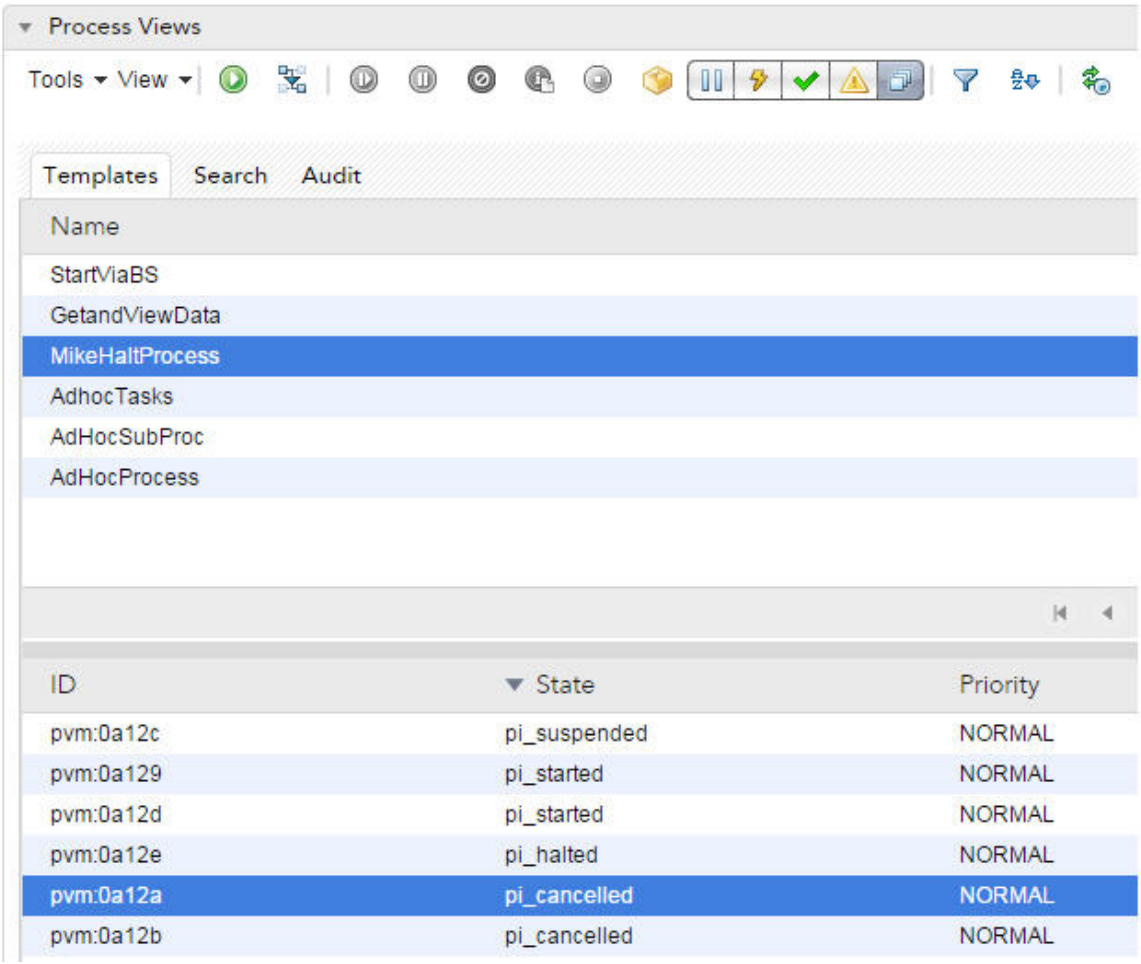
If you want to change the sort order, you can write your own customized sort definition in the `callout.js` script provided by Openspace. See Configuring Custom Functionality for more information about the `callout.js` script file. Then, by setting the `hook.onCollateStrings` property in the `config.properties` file to `true`, every string comparison goes through the `hook.onCollateStrings` property and calls `callout.js` until the column is sorted. See Openspace Configuration Overview for the location of the `config.properties` file.

By default, the `hook.onCollateStrings` property is set to `false`. If you set the `hook.onCollateStrings` property to `true`, but do not amend the `callout.js` script, then the comparison function used is `a.localeCompare(b)`. This compares two strings together using the locale that your browser is using.

The following example illustrates changing the default sort order of the **State** column in the Process Views gadget. It swaps `pi_halted` and `pi_started` so they start at opposite ends in the sequence. Shown below is the default ascending sort order for the **State** column in the Process Views gadget.



Shown below is the default descending order of the **State** column in the Process Views gadget.



The following Javascript swaps `pi_halted` and `pi_started` so they start at opposite ends in the sequence

```
var result = "0";
if ( (context == "ProcessTemplageGadgetView-instances") && ("pi_" ==
a.substr(0,3)) )
{
if( a ! = b)
 {    if( a == "pi_halted" || b == "pi_started" )
  {
   result = "-1"; //always collate lower
  }
  else
  {
   result= "1"; //always collate higher
  }
 }
}
else
}
 var ret = a.localeCompare(b);
 if (ret < 0)
 {
  result= "-1";
 }
 else
 }
  if( ret > 0)
  {
   result= "1";
  }
 }
```

```
}
window.alert("e) javascript onCollateStrings(" + context + ") locale=" + locale + "
a=" + a + " b=" + b+" result="+result return result;
```

You can capture diagnostics about the sort comparisons made by the `callout.js` script in the Logger gadget. By setting `hook.onLoadOpenspace=true` in the `config.properties` file, any `window.alert` messages in the `callouts.js` script are redirected to the Logger gadget.

Note that:

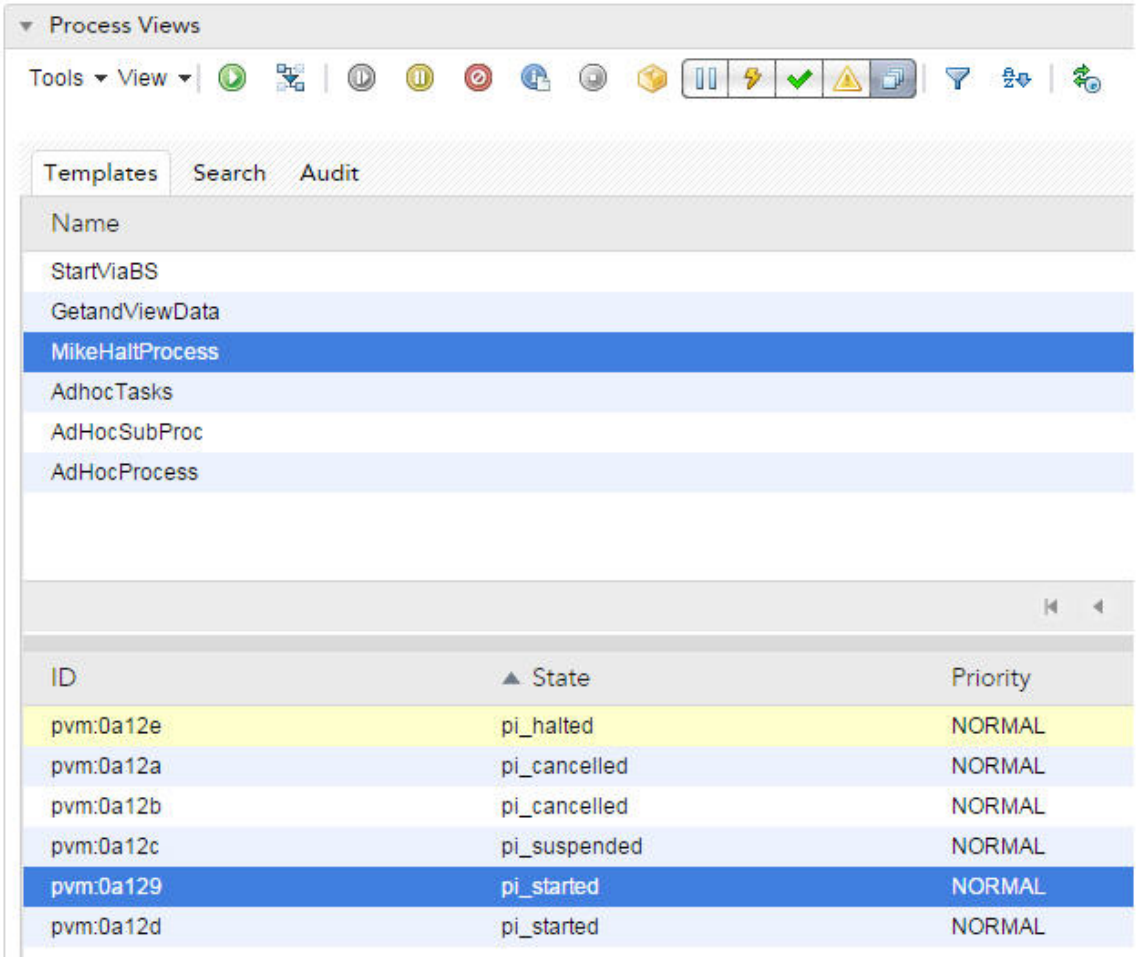
- Passing the parameter `locale` allows you to change different sort orders for different locales.
- Passing a parameter context indicates which component is calling the `hook.onCollateStrings` property.
- Parameters `a` and `b` are the two strings that are compared.
- The function must return

  `-1` to indicate lower.

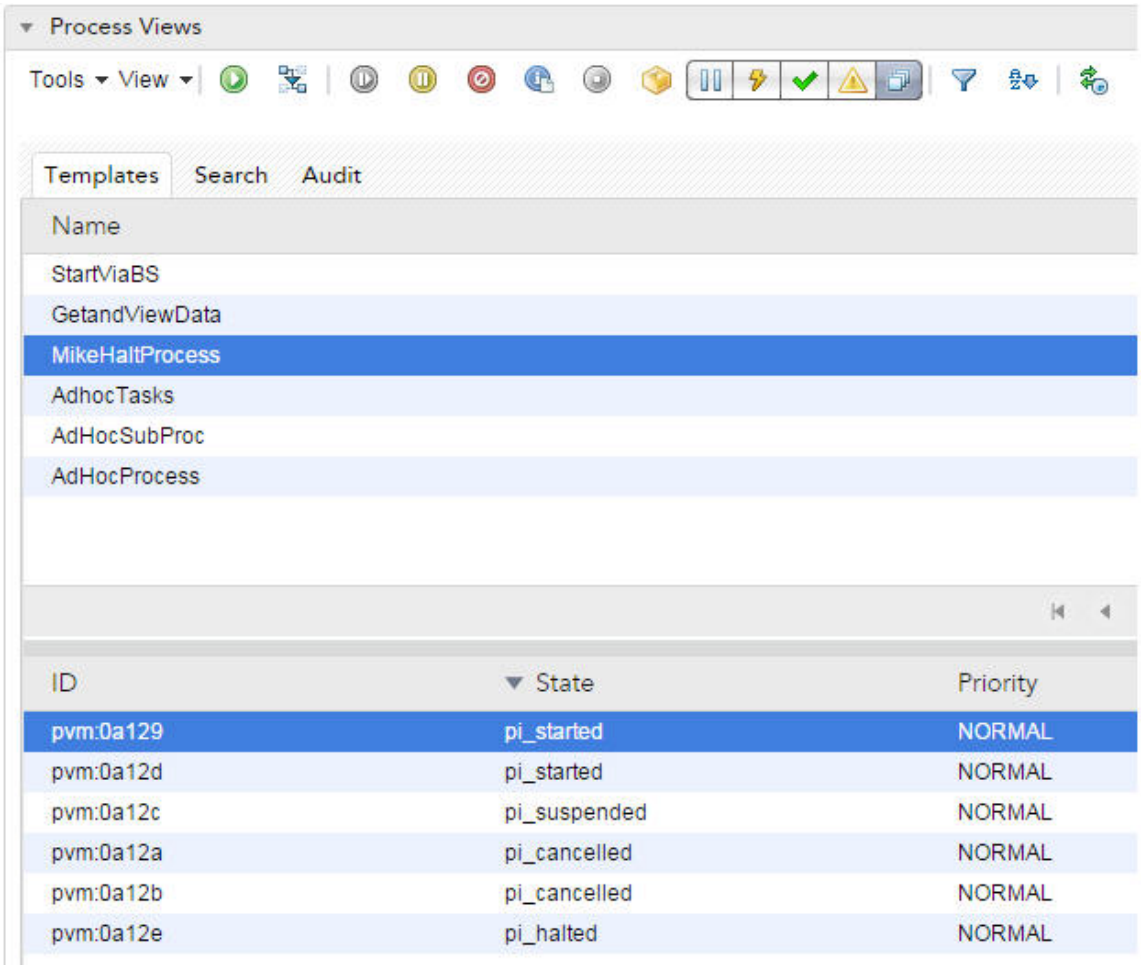
  `0` to indicate identical.

  `1` to indicate higher.

Shown below is the customized sort of the **State** column in the Process Views gadget in ascending order.



Shown below is the customized sort of the **State** column in the Process Views gadget in descending order.



Procedure

1. Open the `config.properties` file in a text editor.

   TIBCO recommend that you take a backup of the `config.properties` file before amending it.

2. Uncomment the `hook.onCollateStrings` property and set it to `true`. Initially, the property is commented out and set to `false`. For example, `hook.onCollateStrings=true`.
3. Save and close the `config.properties` file.
4. Log out and log back into TIBCO Openspace for the changes to take effect.

Business Service App (BizApp)

The Business Service App (BizApp) is available as part of Openspace. It enables you to start business services or inject events into business services that have already started, by providing the appropriate URL. It can render TIBCO Forms and also supports the complete functionality of static and dynamic sub-processes.

BizApp is accessed via a URL (for clarity, the URL is shown with each parameter on a separate line):

protocol://*host*:*port*/openspace/bizapp/bizapp.html
?module=*module name*
&process=*process name*
&version=*version number*
&channel=*channel name*
&payLoad=*payload*
&managedIframe=true|false
&eventWithData=true|false
&tibco_jsonp=true|false
&event=*event name*
&processid=*process ID*
&username=*BPM username*
&auth0=*plain text BPM password*
&auth1=*Base64 BPM password*

The URL parameters are explained in the following table:

Parameter Name	Description
protocol (required)	The communications protocol that is used by Openspace: either http or https. This was determined at installation.
host (required)	The DNS name or IP address of the server that hosts the BPM runtime.
port (required)	The port that is used by Openspace. The default is 8080.
module (required)	The module that defines the business service. It can be found in Openspace under the **BUSINESS SERVICE SUMMARY** that is displayed for a selected Business Service. The value must be URI-encoded.
process (required)	The name of the business service process.
version (required)	The version number of the business service, in the format of *major.minor.micro .[qualifier]*. If you do not specify a qualifier, then the latest deployed version of the application is used.
channel	The channel that is used to open the business service. The default is openspaceGWTPull_DefaultChannel.

Parameter Name	Description
payload	The JSON payload that is passed to the business service.
managedIframe	If this is set to true, the business service is opened in a Managed Frame.
eventWithData	If this is set to true, the BusinessServiceDataEvent is triggered in addition to the BusinessServiceEvent.
tibco_jsonp	If this is set to true, cross-domain access to the business service form definitions is enabled.
event	The name of the event to inject into a business service. You should use this parameter only if you are injecting an event into a business service.
processid	The process ID of the business service that is being injected into. You should use this parameter only if you are injecting an event into a business service.
username	The BPM runtime login name.
auth0	The BPM runtime login password, as plain text.
auth1	The BPM runtime login password, as a Base64-encoded string. You need a Base64 encoder to encode the password.

If a parameter contains spaces, it must be URL-encoded.

Authentication to access BizApp is controlled by the `authenticate` property in the `config.properties` file. For more information, see Configuring Authentication.

Examples

Example URL for starting a business service:

```
http://machinename:8080/openspace/bizapp/bizapp.html?module=%2FCMISSample%2FProcess
%2FCMISSample.xpdl&process=CMISSampleCatchMessageEvent&version=3.0.0
```

Example URL that includes payload as a business service:

```
http://machinename:8080/openspace/bizapp/bizapp.html?module=%2FORDER%2FProcess
%20Packages
%2FOrderProc.xpdl&process=StartOrderForCustomer&version=1.0&channel=openspaceGWTPull
_DefaultChannel&tibco_channel=desktop&payLoad={"items":[
{"$param":"customerId","$value":"123456789","type":"Integer","mode":"IN"}
]}
```

Localizing Openspace Overview

By default, Openspace is available in American English or British English. You can make Openspace available in other languages.

To make Openspace available in different languages, you can do one of the following:

- Deploy one of the TIBCO language packs, available on the TIBCO Software Product Download website.

  From ActiveMatrix BPM 3.0.0 onward, TIBCO is starting to publish language packs as Distributed Application Archives (DAAs). You should use DAA language packs instead of the former Multilingual User Interface (MUI) language packs. DAA language packs can be deployed by ActiveMatrix Administrator, in the same way as any other DAA.

  For some languages, Multilingual User Interface (MUI) language packs will continue to be available until they are replaced later by DAA language packs. MUI language packs can be deployed using the documentation supplied with the language pack.
- Manually translate Openspace to use a new language, if the language you require is not available in a language pack.

Manually Localizing Openspace Overview

You can dynamically change both Openspace and the individual gadgets within Openspace to use different languages, depending on your requirements.

- You can make the whole of Openspace available in a different language. You can either:
  - select a different language when you log in. See *TIBCO Openspace User's Guide*.
  - select a different language after you have logged in. See *TIBCO Openspace User's Guide*.
- You can make individual gadgets within Openspace available in different languages, that are different from both Openspace and other gadgets. For example, Openspace may be configured to use French but you want to open a work item in the Work Views gadget that relates to an English speaking customer. From the Work Views gadget, you can select English from the **Language Selector** drop-down list and translate the Work Views gadget into English while the rest of Openspace remains in French. See *TIBCO Openspace User's Guide*.

Locales Overview

Openspace uses locales to distinguish between different language/country combinations.

Each localized language is represented by a two-letter code, `ll` where `ll` is a lowercase, two-letter ISO 639 language code. For a list of language codes, visit the following web site: http://www.loc.gov/standards/iso639-2/langhome.html

Each country is represented by a two-letter code,`CC` where `CC` is an uppercase, two-letter ISO 3166 country code. For a list of country codes, visit the following web site: http://www.iso.org/iso/country_codes/iso_3166_code_lists.htm.

A locale key is a string representation of a locale that includes a language and, optionally, a country code in the following format:

```
ll_CC
```

You do not have to include a country code but may wish to do so if you want a particular version of a language. For example, if you just want Spanish but do not mind which alphabets or dialects are used, you could just use `es` as the locale key. If you do want a particular alphabet or dialect, Mexican Spanish for example, you can include the country code in the locale key. In this case, the locale key would be `es_ MX`.

Configuring Locales

By default, Openspace automatically provides two built-in language packs (American English and British English). Openspace also provides a `locale.properties` files that lists all 57 known locales that are available. When Openspace loads, it makes a call to the server to check whether any of the language packs for the 57 locales have been installed. If a language pack is installed, then Openspace loads the relevant data for the installed language pack. However, testing for a large number of potential locales can have an impact on the time taken to load Openspace. If you are finding that Openspace is taking a long time to load, TIBCO recommends that you amend the `locale.limit` property in the `config.properties` file to limit the number of locales that Openspace tests for.

The `locale.properties` file assigns a locale property to each of the 57 locales. A snap shot of the file is shown below:

```
locale.0=en_US
 locale.1=en_GB
 locale.2=fr_FR
 locale.3=fr_CA
 locale.4=es_ES
```

By default, Openspace checks for all 57 locales specified in the `locale.properties` file. The `locale.limit` property allows you to limit how many locales Openspace checks for. If you have configured the `locale.limit` property, Openspace processes this file sequentially from the top, until it reaches the limit specified in the `locale.limit` property.

The format of the `locale.limit` property is `locale.limit=`*`value`* where *value* is either:

- numeric. In this case, you must specify the integer that indicates the locale limit that you want Openspace to reach. For example, if you only want en_US and es_ES then the `locale.limit` property should be configured as `locale.limit=4`. Openspace still checks for en_GB, fr_FR and fr_CA but will stop at es_ES. If you want, you can reorder the `locale.properties` file. For example, you could specify that `locale.2=es_ES` and then set the `locale.limit` property to `locale.limit=2`.
- text. In this case, you can specify any text you like and then add that text as part of an extended URL. For example, you can specify the `locale.limit` property as `locale.limit=restrict`. You can then specify the limit as part of an extended URL in Openspace. For example, if you only wanted en_US, you could specify the URL as

  ```
  https://localhost:8080/openspace/openspace/?restrict=0
  ```

The `locale.limit` property is commented out by default.

See Openspace Configuration Overview for the location of the `locale.properties` and `config.properties` files.

Prerequisites

TIBCO recommends that you take a backup of the `config.properties` file before amending it.

Procedure

1. Open the `config.properties` file in a text editor.
2. Uncomment the `locale.limit` property.
3. Specify the *value*, depending on your requirements.
4. Save and close the `config.properties` file.
5. Log out and log back into Openspace for the changes to take effect.

Configuring the TIBCO Openspace Date Time Format

Date and time stamps are often used in Openspace. For example, the gadget status bar sometimes shows a timestamp and you can specify a start date and end date in the Process Instances Measures gadget.
There are two files where the date and time format can be configured for Openspace.

- `TranslatorMessages_ll_CC.properties` file. The date and time for the majority of the gadgets in Openspace is configured using this file.
- `CommonMessages_ll_CC.properties` file. Currently, this file is used by the Data Views gadget only.

where:

- *ll* is a lowercase, two-letter ISO 639 language code. For a list of language codes, visit the following web site: http://www.loc.gov/standards/iso639-2/langhome.html.
- *CC* is an uppercase, two-letter ISO 3166 country code. For a list of country codes, visit the following web site: http://www.iso.org/.

These files are located in the TIBCO ActiveMatrix BPM configuration directory.

- For example, the `CommonMessages_ll_CC.properties` file is available in `C:\Program Data\amx-bpm\tibco\data\tibcohost\Admin-AMX BPM-AMX BPM Server\data_3.2.x\host\plugins\com.tibco.os.dataview_1.0.0.017\resources\dataview\locale`.
- For example, the `TranslatorMessages_ll_CC.properties` file is available `C:\ProgramData\amx-bpm\tibco\data\tibcohost\Admin-AMX BPM-AMX BPM Server\data_3.2.x\host\plugins\com.tibco.openspace.login_1.7.0.036\resources\locale`.

See "Openspace Configuration Overview" in the *TIBCO Openspace Customization Guide* for information about the location of these files.

TIBCO recommends that you take a back-up of these files before you amend them.

Both the files contain the core messages that are used by most of the gadgets across Openspace. You must configure both a `TranslatorMessages_ll_CC.properties` file and a `CommonMessages_ll_CC.properties` file for each language you want Openspace to support.

The format of both files is the same. The patterns making up the various `dateTimePattern` properties are made up of meta characters as defined in Google Web Toolkit. The table below shows the meta characters you can use.

Symbol	Meaning	Presentation	Example
G	era designator	Text	AD
y	year	Number	1996
M	month in year	Text or Number	July (or) 07
d	day in month	Number	10
h	hour in am/pm (1-12)	Number	12
H	hour in day (0-23)	Number	0
m	minute in hour	Number	30

Symbol	Meaning	Presentation	Example
s	second in minute	Number	55
S	fractional second	Number	978
E	day of week	Text	Tuesday
a	am/pm marker	Text	PM
k	hour in day (1-24)	Number	24
K	hour in am/pm (0-11)	Number	0
z	time zone	Text	Pacific Standard Time
Z	time zone (RFC 822)	Number	-0800
v	time zone (generic)	Text	Pacific Time
’	escape for text	Delimiter	’Date=’
"	single quote	Literal	’o’ ’clock’

The following table shows some example patterns.

Example	Formatted Text
`"yyyy.MM.dd G 'at' HH:mm:ss vvvv"`	1996.07.10 AD at 15:08:56 Pacific Time
`"EEE, MMM d, ''yy"`	Wed, July 10, '96
`"h:mm a"`	12:08 PM
`"hh 'o''clock' a, zzzz"`	12 o'clock PM, Pacific Daylight Time
`"K:mm a, vvv"`	0:00 PM, PT
`"yyyyy.MMMMM.dd GGG hh:mm aaa"`	01996.July.10 AD 12:08 PM

See http://www.gwtproject.org/javadoc/latest/com/google/gwt/i18n/client/DateTimeFormat.html for more information.

If a `dateTimePattern1` property is not defined in the locale specific `TranslatorMessages_`*`ll_CC`*`.properties` file then the property value falls back to the property defined in the `TranslatorMessages.properties` file.

The `common=`*`filename`* property in the data views `config.properties` file configures the Data Views gadget to use the date time patterns that are available in the `CommonMessages_`*`ll_CC`*`.properties` file. For example, if your filename is `exampleMessages_`*`ll_CC`*`.properties` then the `common` property must be `common=example`.

The Logger gadget enables you to quickly identify any words that are not translated in a locale pack. Specify the level of logging in the Logger gadget at INFO level, and an error displays if there are any words that are not translated. Words that are not translated default to American English. See *Openspace User's Guide* for more information.

Overview of the Steps Required to Manually Localize Openspace

To make Openspace available in different languages and localizations, you must manually translate the Openspace application files and add the new language/country combination to the `locale.properties` file.

1. See Manually Translating Openspace Application Resource Bundles for information on how to manually translate Openspace application files.
2. Add the new language/country combination as a property to the `locale.properties` file. The `locale.properties` file contains the list of locales that Openspace supports. Adding the new language/country combination as a property to the `locale.properties` file enables the new language to be available for selection from the **Locale Selector** box within Openspace. See Adding the New Language as a Property to the locale.properties File.
3. Create your language pack as a composite application. To do this, perform the following tasks in order:
   a. Create an SOA Project
   b. Configure the Composite
   c. Create the Distributed Application Archive (DAA)
   d. Deploying the DAA

Manually Translating Openspace Application Resource Bundles

You can manually translate the Openspace application resource bundles.

The location of the resource bundles are:

- **Openspace** - There are several Openspace application plugins that contain resource bundles.
  - `\`*`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment-adminservername`*`\data_`*`n`*`\host\plugins\com.tibco.openspace.login_`*`n`*`\resources\locale`.
  - `\`*`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment-adminservername`*`\data_`*`n`*`\host\plugins\com.tibco.openspace.login_`*`n`*`\resources\CalendarGadget\locale`.
  - `\`*`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment-adminservername`*`\data_`*`n`*`\host\plugins\com.tibco.openspace.login_`*`n`*`\resources\openform\locale`
  - `\`*`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment-adminservername`*`\data_`*`n`*`\host\plugins\com.tibco.os.data_`*`n`*`\resources\data\locale`.
  - `\`*`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment-adminservername`*`\data_`*`n`*`\host\plugins\com.tibco.bpm.casemanagement_`*`n`*`\resources\dataview\locale`.
  - `\`*`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment-adminservername`*`\data_`*`n`*`\host\plugins\com.tibco.os.org_`*`n`*`\resources\org\locale`.
  - `\`*`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment-adminservername`*`\data_`*`n`*`\host\plugins\com.tibco.openspace.adhoc.app_`*`n`*`\resources\adhoc\locale`.
  - `\`*`CONFIG_HOME`*`\tibcohost\`*`adminenvironment-bpmenvironment-adminservername`*`\data_`*`n`*`\host\plugins\com.tibco.os.businessservice.app_`*`n`*`\resources\bizapp\locale`
- **Accessible Openspace** - These are available in

- *CONFIG_HOME*\tibcohost*adminenvironment-bpmenvironment-adminservername*\data_*n*\host\plugins\com.tibco.os.a11y.app_*n*\accessibility\locale.
- *CONFIG_HOME*\tibcohost*adminenvironment-bpmenvironment-adminservername*\data_*n*\host\plugins\com.tibco.os.a11y.app_*n*\accessibility\webcomponent\locale.

- **MobileSpace** - These are available in *CONFIG_HOME*\tibcohost*adminenvironment-bpmenvironment-adminservername*\data_*n*\host\plugins\com.tibco.openspace.login_*n*\resources\mobile\advanced\locale.

See Openspace Configuration Overview for information.

Procedure

1. For each Openspace application, copy the base properties files. The base properties files are the files that do not have locale key already specified in the filename.
2. Once you have translated them, you must:
   - Rename the files so that the file names now include the new locale key, see Locales Overview. For example, if you are translating from American English to Spanish, then `LoggerMessages.properties` must now be called `LoggerMessages_es.properties`. Properties files should be saved in UTF-8 encoding.
   - Copy the files back to their original location.

- All forms that you are using in your process must also be translated. For information on how to translate forms, see *TIBCO Business Studio Forms User's Guide*.
- You cannot translate a default TIBCO form. You must create a custom form for all your steps and manually translate these. Once you have translated the TIBCO Forms, you must redeploy your application to BPM for the new language to take effect.
- Openspace cannot translate any third party OpenSocial gadgets. If these gadgets are not available from the third party in the new language, they display in Openspace in their original language.
- The Logger gadget enables you to quickly identify any words that are not translated in a locale pack. Specify the level of logging in the Logger gadget at INFO level, and an error displays if there are any words that are not translated. Words that are not translated default to American English. See "Adding gadgets to a Tab" in *TIBCO Openspase User's Guide*.
- You can configure your Openspace Login page to use a different language using the `?locale` query parameter. See "Adding a locale code" in *TIBCO Openspace User's Guide*.
- See "Logging into Openspace" in the *TIBCO Openspace User's Guide* for information on selecting another language from the Openspace Login page.

Adding the New Language as a Property to the locale.properties File

To make the new language available in the **Locale Selector** drop-down list within Openspace, add the new language to the `locale.properities` file.

The `locale.properities` file is located in *CONFIG_HOME*\tibcohost*adminenvironment-bpmenvironment-adminservername*\datan\host\plugins\com.tibco.openspace.login_*n*\resources\locale. See Openspace Configuration Overview for more information.

Add the property in the following format:

`locale.`*n*`=`*locale_code* where:

- *n* is an incremental number that increases each time you add a new language, for example locale.0, locale.1 and so on.
- *locale key* is the ISO standard language and country code combination, see Locales Overview.

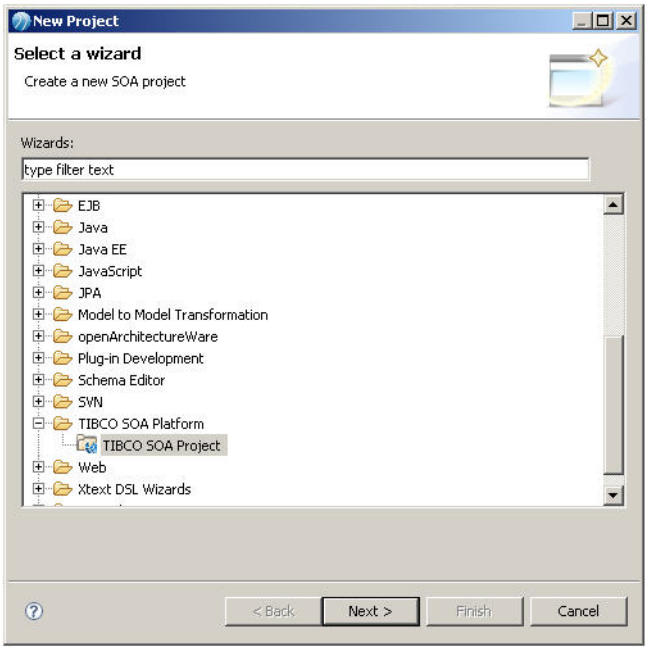
For example, `locale.1=en_GB`.

Create an SOA Project

You need to create an SOA project in TIBCO Business Studio to contain the composite that is deployed to the BPM runtime. Open Studio for Designers.

Procedure

1. On the toolbar, click the drop-down next to **Enable/Disable Business Studio Capabilities** and ensure the **TIBCO SOA Platform Extension** capability is selected.
2. Click **File** > **New** > **Project** .
3. Select **TIBCO SOA Platform** > **TIBCO SOA Project** and click **Next**.

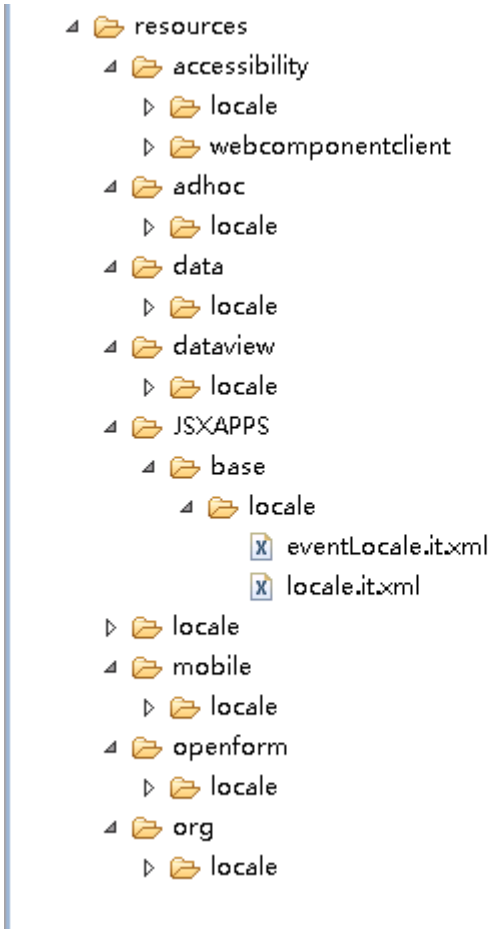


4. In the **Project** box, type the name of your SOA project and click **Next**.
5. From the Asset Type Selection dialog, deselect **Mediation Flows** but accept all the other defaults by clicking **Next**.
6. From the Composite Project dialog, select **Empty SOA Project** and click **Finish** to save your data and exit the wizard.

   A message is displayed asking you if you want to open the associated modeling perspective for this project. Click **Yes**.
7. If you open or amend any of the properties files in TIBCO Business Studio, the files will be corrupted unless you change TIBCO Business Studio to use UTF-8 encoding. Therefore, TIBCO recommend that you change TIBCO Business Studio to use UTF-8 encoding. To do this:
   a) Click **Window** > **Preferences** .
   b) Expand **General** and click **Content Types**. The content types are listed in the upper right corner of the pane.
   c) Expand **Text** and click **Java Properties Files**.
   d) In the **Default Encoding** box, replace the existing value with **UTF-8** and click **Update**.

e) Click **OK** to exit the dialog.

8. In the root of your SOA project directory in your workspace, create a `reources` folder as the root of folder of the Openspace application resource bundles whose properties files you are translating. You must create a folder structure that matches the folder structure of your ActiveMatrix BPM instance. For example,

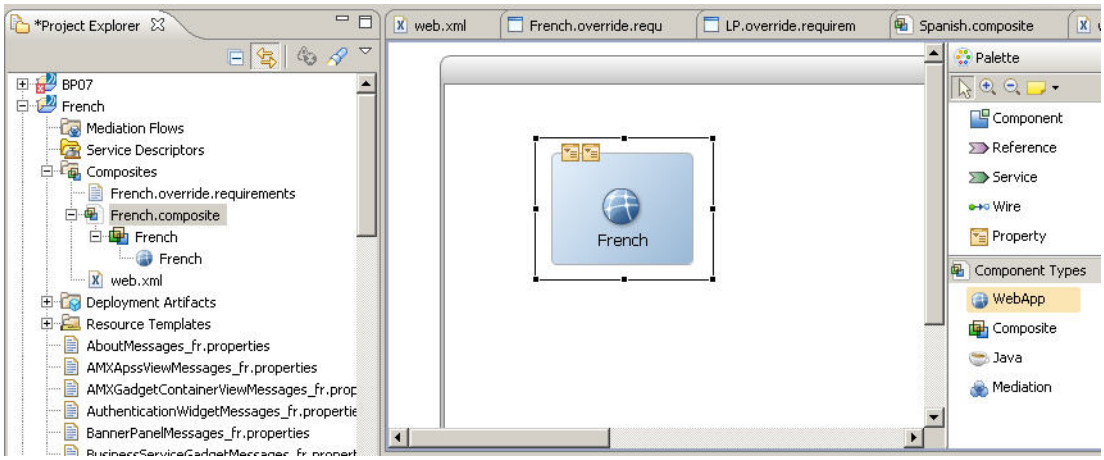


Configure the Composite

You need to configure the composite created above so that it can use the Openspace properties files from the language pack.

Procedure

1. Select ***projectname*** > **.Composite** where *projectname* is the name of your SOA project.
2. From the **Component Type** palette, select **WebApp** and drag it to your composite.
3. In the **Name** box, type a name for your web application



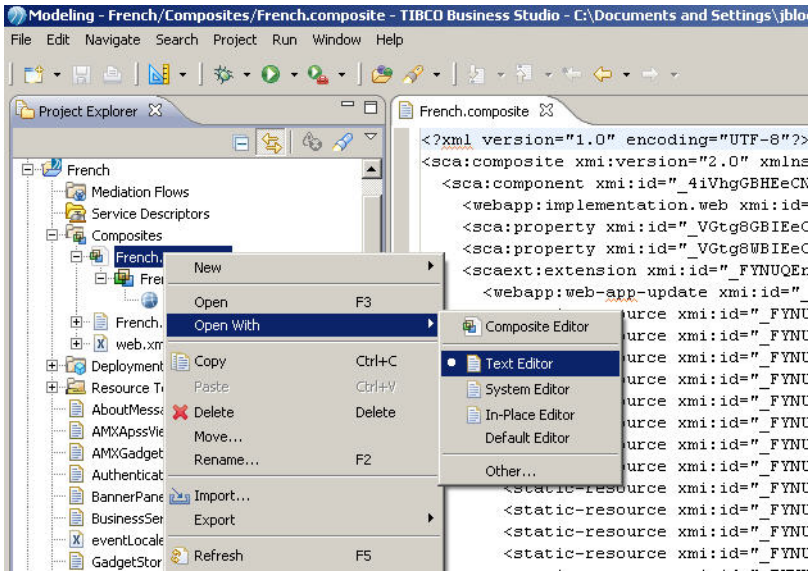
4. Click **Properties**.
5. From the **Basic** tab, select **Plugin Project**.
6. Click **Properties** and set the values for the following properties as described below:

Property	Value
contextRoot	LanguagePackContextRoot_CODE
defaultConnector	httpConnector

7. You will see that you have a **Component** ***webappname*** **is not configured** error displayed against your web application. You can use the quick fix to resolve this error so click **Generate servlet implementation**. The **Generate servlet implementation** dialog is displayed. Click **Next** to accept the defaults and **Finish** to apply the fix. This fix is required because TIBCO Business Studio has some validation rules that require a generated servlet implementation.
8. Once the servlet implementation has generated, select your web application and click **General**.
9. From **Administrator Requirements**, select **Override**.
   The **Application Administration Override** tab is displayed.
10. From the **Application Administration Override** tab, select your SOA project, select the **Resources** tab.
11. From the **Resource** tab, click **Browse**.
    The Open Resource dialog displays.
12. From the **Select an item to open** box, type ***.properties**. A list of all your properties files displays. Select all of these and click **OK**.

    If no files are shown, type **?** to display the list.

13. From the **Select an item to open** box, type `*.xml`. The `locale.xx.xml` and `eventLocale.xx.xml` display where *xx* the locale code. Select both of these and click **OK**.
14. Click **OK** to close the dialog.
15. Right-click *`workspace`*`\projectname\composites\name.composite` file (where *workspace* is the location of your workspace) and select **Open with** > **Text Editor** .



16. Locate the following line:

```
</sca:component>
```

17. Copy the XML similar to that shown in os-scaext-extension.txt.

    Note that:

    - The file containing the XML cannot be downloaded using the link above from the PDF document. This link must be used from the HTML document -- see TIBCO ActiveMatrix BPM documentation.
    - The folder structure under resources for `event.xx.xml` and `eventLocale.xx.xml` is `JSXAPPS\base\locale` as shown in the screenshot above.
    - For each `static-resource` element, the `xmi:id` must be unique.
      - *location* is the path of the resource bundle in SOA project
      - *path* is the path of the matching bundle in the corresponding plugin

      For example, `OrgBrowserMessages_zh_CN.properties` is located in the plugin named `com.tibco.os.org_n` and under the folder of `resources/org/locale`. The value of *path* is `/org/locale/OrgBrowserMessages_zh_CN.properties`. However, the *path* values for `event.xml` and `eventLocale.xml` are `/JSXAPPS/base/locale/locale.zh_CN.xml` and `/JSXAPPS/base/locale/localeLocale.zh_CN.xml`, respectively.

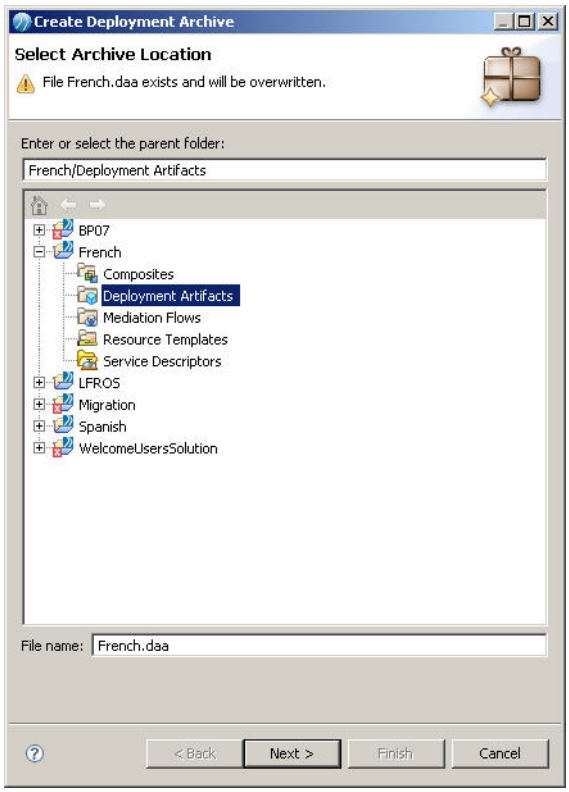
18. Save the file.

Create the Distributed Application Archive (DAA)

As the SOA project only contains a composite, you must package the composite in a DAA, before it can be deployed.

Procedure

1. From TIBCO Business Studio in Project Explorer, navigate to ***name*** > **.composite** where *name* is the name of the composite you created in Configure the Composite.
2. Right-click **name** > **.composite** and select **Create DAA**. The Create Deployment Archive wizard displays.



3. From the Select Archive Location dialog, click **Next** to accept the defaults.
4. From the Select Distribution dialog, click **Next** to accept the default.
5. From the Create Deployment Archive dialog, click **Finish**.

Deploying the DAA

To make Openspace available in different languages and localizations, you must manually translate the Openspace application files and add the new language/country combination to the `locale.properties` file. Then you must create your language pack as a composite application. Once you have created your DAA, you must deploy it.

Procedure

1. From the Deployment Server view, right-click the deployment server you want, and click **Deploy Module...**. The Deploy Module wizard displays.
2. From the Select Module Type dialog, select **Deploy DAA...**.
   The Application Setup dialog displays.
3. Select **Workspace**. The DAA Selection dialog displays. Navigate to ***name*** > **.daa** where *name* is the name of the DAA you created in Create the Distributed Application Archive (DAA). Select it and click **OK** to close the dialog.
4. Click **Next** to exit the **Application Setup** dialog.
   The Administrator Setup dialog displays.
5. Make sure **BPMEnvironment** is selected and click **Next**.
   The Distribution dialog displays.
6. Select **Single Node** and select **BPMNode** from the drop-down list. Click **Next**.
   The Property Configuration dialog displays.
7. Make sure the values of the properties match the ones you defined and click **Finish**.

8. Once the Deployment Complete dialog is displayed, click **Close** to close the dialog. The application is now deployed.

If you wish, you can go to TIBCO Administrator to check that the application is running.

Result

Now the application is running, the next time you login to Openspace it is displayed in the new language.

If Openspace was running before the language pack was deployed, you have to reload Openspace to be able to use the new language.

Configuring What Happens on Openspace Logout

When a user logs out of TIBCO Openspace, usually the current Openspace session is cleared and the current URL is reloaded. However, you can specify a different URL to load, using the Openspace property `logout.path`. If you are using CA SiteMinder, for example, you can specify the URL of a custom logout page that clears the SiteMinder session.

The property `logout.path` is in the Openspace `config.properties` file, in the TIBCO ActiveMatrix BPM configuration directory. For example:

- Openspace:

  ```
  C:\ProgramData\amx-bpm\tibco\data\tibcohost\Admin-AMX BPM-AMX BPM Server
  \data_3.2.x\host\plugins\com.tibco.openspace.login_1.7.1.00n\resources
  \config.properties
  ```

- Accessible Openspace:

  ```
  C:\ProgramData\amx-bpm\tibco\data\tibcohost\Admin-AMX BPM-AMX BPM Server
  \data_3.2.x\host\plugins\com.tibco.os.a11y.app_1.1.1.005\accessibility
  \config.properties
  ```

Prerequisites

TIBCO recommends you back up the `config.properties` file before amending it.

Procedure

1. Open the `config.properties` file in a text editor.
2. Configure the property `logout.path` as follows:

Value	Description
`logout`	The default value causes the ActiveMatrix BPM server to clear the current Openspace session and then the current URL is reloaded, which results in the Openspace login page being displayed.
Empty string (or property is commented out or missing)	This causes the Openspace client browser to clear the current Openspace session and then the current URL is reloaded, which results in the Openspace login page being displayed.
A URL starting `http`	This value causes the ActiveMatrix BPM server to clear the current Openspace session and then this URL is loaded. If you are using SiteMinder, you can specify the URL of a custom logout page that clears the SiteMinder session and then redirects to your preferred login page: either the Openspace login page or a custom login page.

3. Save and close the `config.properties` file.
4. Log out and log back into Openspace for the changes to take effect.