

# **TIBCO iProcess™ Workspace Lite**

## **Customization Guide**

*Software Release 1.2*  
*October 2010*

## 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 LICENSE.PDF) 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.

TIB, TIBCO, TIBCO Adapter, Predictive Business, Information Bus, The Power of Now, TIBCO ActiveMatrix BusinessWorks, and TIBCO iProcess are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or 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 © 2006-2010 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

# Contents

<b>Preface</b> .....	<b>v</b>
Related Documentation .....	vi
TIBCO iProcess Workspace Lite Documentation .....	vi
How to Contact TIBCO Support .....	vii
 <b>Chapter 1 Introduction</b> .....	 <b>1</b>
Customizing the iProcess Workspace Lite Application .....	2
Architectural Overview .....	3
Logging .....	5
 <b>Chapter 2 Customizing Skins</b> .....	 <b>7</b>
Introduction .....	8
Adding a Skin .....	9
Configuring the Default Skin .....	11
 <b>Chapter 3 Customizing Sorting and Filtering</b> .....	 <b>13</b>
Overview .....	14
Action Request Files .....	15
Customizing Sort Order .....	16
Customizing the Sort Order on the Case List .....	16
Customizing the Sort Order on the Work Item List .....	22
Adding a Custom Filter .....	27
Filter Request .....	27
Customizing Filter Request Procedure .....	32
 <b>Chapter 4 Adding an Additional Language</b> .....	 <b>35</b>
Introduction .....	36
Adding Language Procedure .....	37
 <b>Chapter 5 Customizing Layout</b> .....	 <b>39</b>
Customizing Layout .....	40
Workspace Lite XSLT Library .....	41
User Customization XSLT .....	43

Adding CDQPs to the Work Item List ..... 45

Removing a Column From the Work Item List ..... 51

**Chapter 6 Additional Customizations..... 55**

Time-outs ..... 56

Using Custom Forms ..... 57

    Deploying Custom Forms ..... 59

How To Embed TIBCO iProcess Workspace Lite in a Portal ..... 66

Customizing the Number of Cases Returned from the Server ..... 69

# Preface

This guide describes how to customize the TIBCO iProcess Workspace Lite product.

## Topics

---

- [Related Documentation, page vi](#)
- [How to Contact TIBCO Support, page vii](#)

## Related Documentation

---

This section lists documentation resources you may find useful.

### TIBCO iProcess Workspace Lite Documentation

The following documents comprise the TIBCO iProcess Workspace Lite documentation set:

- *TIBCO iProcess Workspace Lite Installation Guide* - Read this manual for information about installing TIBCO iProcess Workspace Lite.
- *TIBCO iProcess Workspace Lite Release Notes* - Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for each release.
- *TIBCO iProcess Workspace Lite User's Guide* - Read this manual for instructions on using the iProcess Workspace Lite application.
- *TIBCO iProcess Workspace Lite Customization Guide* - This manual provides information about customizing iProcess Workspace Lite.

## How to Contact TIBCO Support

---

For comments or problems with this manual or the software it addresses, please contact TIBCO Support as follows.

- 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.





## Chapter 1      **Introduction**

This chapter provides an introduction to customizing the TIBCO iProcess Workspace Lite product.

### Topics

---

- [Customizing the iProcess Workspace Lite Application, page 2](#)
- [Architectural Overview, page 3](#)
- [Logging, page 5](#)

## Customizing the iProcess Workspace Lite Application

---

The TIBCO iProcess Workspace Lite application can be customized in the following ways:

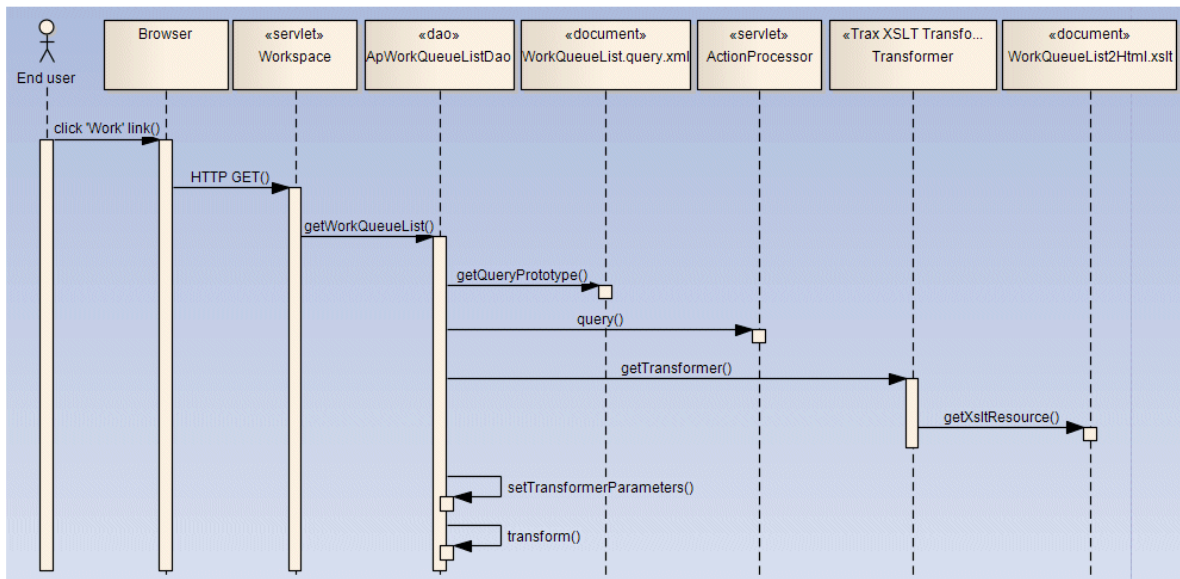
- Customizing skins (adding a new skin and configuring a default skin) — see [Customizing Skins on page 7](#).
- Customizing sort order — [Customizing Sort Order on page 16](#).
- Adding a custom filter — [Adding a Custom Filter on page 27](#).
- Adding an additional language — see [Adding an Additional Language on page 35](#).
- Customizing the application layout — see [Customizing Layout on page 40](#).
- Adding CDQPs to the work item list — [Adding CDQPs to the Work Item List on page 45](#).
- Removing a column from the work item list — [Removing a Column From the Work Item List on page 51](#).
- Modifying time-out periods — see [Time-outs on page 56](#).
- Using custom forms — see [Using Custom Forms on page 57](#).
- Embedding TIBCO iProcess Workspace Lite in a portal — [How To Embed TIBCO iProcess Workspace Lite in a Portal on page 66](#).
- Customizing the number of cases returned from the server — [Customizing the Number of Cases Returned from the Server on page 69](#).

The intended audience for this guide is web developers who are familiar with JSP, XML, XSLT, HTML, and CSS. Knowledge of these subjects is necessary to perform some of these customizations.

## Architectural Overview

The following sequence diagram illustrates one example of how requests from the browser are handled — this diagram uses the work queue list query (WorkQueueList.query.xml) as an example. All requests are handled in a similar fashion, with one Data Access Object (DAO)<sup>1</sup>, one query file, and one XSLT per request.

Figure 1 Example Request Sequence Diagram



The following describes the elements in the sequence diagram:

- **End user** - The user of the iProcess Workspace Lite application. The request begins with the end user making a selection in the application (e.g., clicking on the **Work** link to display the work queue list).
- **Browser** - The officially supported browsers are Microsoft Internet Explorer 7 and 8, as well as Mozilla Firefox 3.0, 3.5, and 3.6.
- **ApWorkQueueListDao** - This is the heart of the system. The DAO's purpose is to make Action Processor requests based on the query XML file and the

1. DAO is a Sun 'blueprint' pattern. For more information, see the following website: <http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>.

parameters received in the HTTP request, and then apply the specified XSLT to the result before returning it to the user via the HTTP response stream.

- **WorkQueueList.query.xml** - The XML document containing the Action Processor request into which user parameters are substituted. There are certain ways the user may wish to change this, though they are relatively limited — for more information, see [Customizing Sorting and Filtering on page 13](#).
- **Action Processor** - A standard, unmodified, Action Processor from TIBCO iProcess Workspace (Browser) v11.1 must be installed. It can be either a Java or .NET Action Processor.
- **Transformer** - This is a standard Trax-compliant XSLT engine provided by the web application server, such as the Xalan XSLT processor included in the JDK running the Tomcat server.
- **WorkQueueList2Html.xslt** - This is the user-modifiable XSLT file that controls the way the action request is displayed.

## Logging

---

The default logging system for TIBCO iProcess Workspace Lite is **log4j**. A `log4j.properties` file is included in the following directory:

*InstallDir*\WEB-INF\classes\

where *InstallDir* is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: `C:\Tomcat\webapps\ipe-web`.

For information about configuring the log4j logging system, see the Configuration section in the following document:

<http://logging.apache.org/log4j/1.2/manual.html>



## Chapter 2 Customizing Skins

This chapter provides the procedure to perform to add additional skins to your iProcess Workspace Lite application, as well as how to configure a default skin to be used in situations when the user is not presented with a Login screen.

### Topics

---

- [Introduction, page 8](#)
- [Adding a Skin, page 9](#)
- [Configuring the Default Skin, page 11](#)

## Introduction

---

Applying a “skin” allows the user of the iProcess Workspace Lite application to choose the desired “look-and-feel”.

There are four skins provided with the application:

- Accessible
- Carbonated
- Popular
- TIBCO

This chapter describes how to include additional skins.

There are numerous web sites available from which you can obtain CSS templates that can be used to create skins. Two of the skins provided in the iProcess Workspace Lite application (carbonated and popular) were downloaded from the following web site, which provides free CSS templates:

<http://www.freecsstemplates.org>

One of the benefits of using the sample skins from this site is that many of the available samples have `<div>` elements corresponding to the elements required in iProcess Workspace Lite, so you can simply copy them.



## Adding a Skin

---

To add an additional skin to the iProcess Workspace Lite application, follow these steps:

1. Create a new XML file in the following location:

*InstallDir*\WEB-INF\classes\skins\SkinName.xml

where:

- *InstallDir* is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: C:\Tomcat\webapps\ipe-web.
- *SkinName* is the name of the skin you are adding.

2. In the XML file created in step 1, include the following elements:

---

```
<skin>
  <header></header>
  <sidebar></sidebar>
  <footer></footer>
</skin>
```

---

3. Within each of the elements added in step 2, include one or more **<div>** elements, and within those, include whatever HTML you want in the header, sidebar, and footer.

If you download one of the skins available from the web site mentioned above, you can copy the **<div>** sections from the downloaded file, then paste them into your new XML file.

4. Modify the **<div>** sections to include iProcess Workspace Lite-specific items.

Using one of the provided skins as a guide, include all of the appropriate hrefs and actions needed to provide links/buttons for displaying the process list, work queue list, skip navigation feature, etc.

Note that for some skins, one or more of the elements shown in step 2 may be left empty.

5. Create a Cascading Style Sheet (CSS) file named `default.css` in the following directory:

*InstallDir*\skins\SkinName\css\

where:

- *InstallDir* is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: C:\Tomcat\webapps\ipe-web.
  - *SkinName* is the name of the skin you are adding.
6. Include in the `default.css` file all of the appropriate output styles (fonts, color, spacing, etc.) needed for the skin you are adding.
- If you downloaded a skin from the web site mentioned above, it will include a CSS file that you can use intact, or modify it as desired.
7. Open the *InstallDir*\login.jsp file and locate the **<option>** elements for the provided skins. They will appear as follows:

---

```
<select id="skin" title="skin" name="skin" >
  <option value="accessible" value="" <% if ("accessible".equals(style)) ...
  <option value="carbonated" value="" <% if ("carbonated".equals(style)) ...
  <option value="popular" value="" <% if ("popular".equals(style)) ...
  <option value="tibco" value="" <% if ("tibco".equals(style)) ...
</select>
```

---

8. Add an additional **<option>** element for the new skin you are adding.
- This adds the new skin name to the **Skin** drop-down list on the **Login** screen so that the user can select it before logging in.
9. (Optional) Add a thumbnail image to the Home page in the iProcess Workspace Lite application:
    - a. Create the thumbnail image and save it with the others in the following directory:
 

*InstallDir*\images
    - b. Open the following file:
 

*InstallDir*\WEB-INF\classes\content\index.content.xml
    - c. Add a new **<img>** element that points to the image you saved in step 9a above. Use one of the existing **<img>** elements as a guide.
    - d. Save and close the `index.content.xml` file.

## Configuring the Default Skin

---

In scenarios where the user is not presented with the Login dialog (such as when using single sign-on), the user is not able to choose the desired skin. Because of this, you can specify which of the available skins will be used in the application.

To configure the default skin:

1. Ensure that your application server is stopped.

2. Open the following file:

*InstallDir*\WEB-INF\web.xml

where:

- *InstallDir* is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: C:\Tomcat\webapps\ipe-web.

3. Locate the **defaultSkin** parameter:

---

```
<context-param>
  <description>Default skin to be used if none is specified on URL
</description>
  <param-name>defaultSkin</param-name>
  <param-value>tibco</param-value>
</context-param>
```

---

4. Change the **<param-value>** value to the name of the new default skin. This value is case sensitive, and the skin must exist in the *InstallDir*\WEB-INF\classes\skins directory.

5. Save and close the web.xml file.

6. Restart your application server.

7. Run the iProcess Workspace Lite application.

For the application to use the default skin specified in the **defaultSkin** parameter, the skin cannot be specified on the URL.

Also, note that this does not affect the skin that is selected by default in the **Skin** field on the Login dialog.



## Chapter 3      **Customizing Sorting and Filtering**

This chapter describes how to customize the sorting and filtering of lists.

### Topics

---

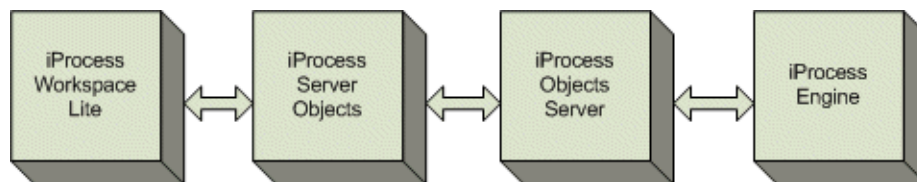
- [Overview, page 14](#)
- [Customizing Sort Order, page 16](#)
- [Adding a Custom Filter, page 27](#)

## Overview

---

When a user performs any function in the iProcess Workspace Lite application (for example, clicking on the **Work** link to display the work queue list), it causes an "action" request to be sent to the Action Processor, which in turn creates a method call to the TIBCO iProcess Server Objects. The iProcess Server Objects pass the call on to the TIBCO iProcess Objects Server, which sends it to the TIBCO iProcess Engine. This is illustrated below:

*Figure 2 Action Request Flow*



The action request sent to the Action Processor must contain the appropriate parameters in the proper order so that the Action Processor can create an iProcess Server Objects call. Because of this, to modify the action requests generally requires an understanding of the structure of the information in iProcess Server Objects method calls.

Rather than requiring you to obtain all of the iProcess Server Objects documentation and gain an understanding of that product, this section describes very specific steps that you need to perform to accomplish specific customization tasks. (If you have an understanding of the iProcess Server Objects product, you are free to exercise that knowledge and customize any of the available action requests.)

The customization tasks that are described here are:

- Customizing the order in which work items and cases are sorted in the work item and case lists, respectively — see [Customizing Sort Order on page 16](#).
- Adding a custom filter to the work item and case lists — see [Adding a Custom Filter on page 27](#).

## Action Request Files

An action request is in the form of an XML file. The action request XML files can be manually modified to customize the request to the Action Processor.

When you install iProcess Workspace Lite, a JAR file is installed that contains all of the action request XML files. This file is installed in the following location:

*InstallDir*\WEB-INF\lib\com.tibco.ipe.workspace.lib-version.jar

where:

- *InstallDir* is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: C:\Tomcat\webapps\ipe-web.
- *version* is the version number of your release of iProcess Workspace Lite.

Although the JAR contains all of the action request files, you will only be dealing with ones that display the case and work item lists. They are:

- CaseList.query.xml
- WorkItemList.query.xml

## Customizing Sort Order

---

By default, the items listed in the work item and case lists are sorted as follows:

- Work item list: process description (**Process** column in the work item list)
- Case list: case number (**Id** column in the case list)

If desired, you can change the default sort order of the work item and cases lists. For example, you might want to specify that work items in all work item lists are sorted by priority, in descending order, so the highest priority work items are displayed first.

Note that prior to this release, changing the default sort order for both the work item list and case list required modifying the action request sent to the Action Processor, which can be an involved process. In this release, however, an improved method of changing the default sort order on the work item list has been implemented. This involves changing only three parameters in the `web.xml` file, rather than changing the action request.

Because the method of customizing the sort order is different depending on the type of list, it is described in two separate sections, as follows:

- [Customizing the Sort Order on the Case List on page 16](#)
- [Customizing the Sort Order on the Work Item List on page 22](#)

### Customizing the Sort Order on the Case List

To specify a custom sort order on the case list, you will modify the following action request file:

- `CaseList.query.xml`

The following subsections describe the elements of the customized sort request, followed by the step-by-step procedure to add a customized sort order to the case list.

#### Sort Request

A customized sort request is accomplished by adding the following elements to the action request:



---

```
<sso:vSortField>
  <sso:FieldName>SW_ARRIVAL</sso:FieldName>
  <sso:IsAscending>true</sso:IsAscending>
  <sso:SortTypeAs>swDateTimeSort</sso:SortTypeAs>
</sso:vSortField>
```

---

To sort on a single column in the case list, you will add one set of these elements. To sort on multiple columns, you can add one set of these elements for each column on which you want to sort.

The sort elements consist of the following:

- **FieldName** - Identifies the field/column on which you want to sort.
- **IsAscending** - Specifies whether the sort order will be ascending or descending.
- **SortTypeAs** - Specifies the type of data the sort order is based on.

These are described in more detail below.

### FieldName

You can sort on most of the columns in the case list. Note, however, that when you specify the column(s) to sort on, you do not use the column header label shown in the list — you must use the "system field" that represents that column information. For example, instead of using "Description" in the query, you use the SW\_CASEDESC system field:

---

```
<sso:vSortField>
  <sso:FieldName>SW_CASEDESC</sso:FieldName>
  <sso:IsAscending>true</sso:IsAscending>
  <sso:SortTypeAs>swTextSort</sso:SortTypeAs>
</sso:vSortField>
```

---

The following table lists the system fields used to sort on the columns on the case list:

Table 1: System Fields Used to Sort the Case List

Column	System Field	Data Type
Id	SW_CASENUM	Numeric

---

Table 1: System Fields Used to Sort the Case List

Column	System Field	Data Type
Description	SW_CASEDESC	Text
Started	SW_STARTED	DateTime
Completed	SW_TERMINATED	DateTime
Manage <sup>1</sup>	n/a	n/a

1. You cannot sort on the Manage column.

**IsAscending**

The `<sso:IsAscending>` element specifies whether the sort order is ascending or descending. Pass "true" in the `<sso:IsAscending>` element to sort in ascending order; pass "false" to sort in descending order.

```
<sso:vSortField>
  <sso:FieldName>SW_CASEDESC</sso:FieldName>
  <sso:IsAscending>true</sso:IsAscending>
  <sso:SortTypeAs>swTextSort</sso:SortTypeAs>
</sso:vSortField>
```

**SortTypeAs**

The way in which the system sorts the values in the sort fields depends on the type of data in that field (e.g., text data is sorted alphabetically, numeric data is sorted numerically, etc.).

You can, however, tell the server to sort the data in a field using a data type that is different than the field’s default data type. For instance, you can tell it to sort the data in the SW\_CASEDESC system field (which is a text field) as numeric data. (For the default data types, see the tables in [FieldName on page 17](#).)

Note, however, that if the sort field does not contain something readily convertible to the specified type, the sort results may not be what you expect. For example, if sorting text as a numeric field but some of the text fields contain non-numeric data, the results may not be what you expected.

The `<sso:SortTypeAs>` element is used to specify the sort type for the specified system field. You can pass one of the following constants for the sort type:

Table 2: SortAsType Constants for Sorting the Case List

Constant	Description
swDateTimeSort	Sort as date/time
swNumericSort	Sort as real number
swTextSort	Sort as text

For example:

```
<sso:vSortField>
  <sso:FieldName>SW_CASEDESC</sso:FieldName>
  <sso:IsAscending>true</sso:IsAscending>
  <sso:SortTypeAs>swNumericSort</sso:SortTypeAs>
</sso:vSortField>
```

You can specify a different sort type for any of the columns on the case list.

To sort using the default data type for the specified system field, you can either include the **<sso:SortTypeAs>** element with the default data type specified, or merely leave out the **<sso:SortTypeAs>** element. For example:

```
<sso:vSortField>
  <sso:FieldName>SW_CASEDESC</sso:FieldName>
  <sso:IsAscending>true</sso:IsAscending>
</sso:vSortField>
```

## Procedure for Customizing the Sort Request for the Case List

The following provides step-by-step instructions for customizing the sort order of a case list.

Assume the following for this example:

- You are customizing the sort on the case list — therefore, you will modify the `CaseList.query.xml` file.
- You are specifying that the case list be sorted on the case description (i.e., the **Description** column) — therefore, you will sort on the `SW_CASEDESC` system field.
- The descriptions should appear in alphabetical order — therefore, you will specify that the sort be in ascending order.

- As the description is text, you must specify a "sort type" of **swTextSort**.

Using the information in the preceding sections, you should be able to modify this example to customize the sort for any of the columns on the case list.

To customize the action request for the example described above, follow these steps:

1. Ensure that your application server is stopped.
2. Extract the files from the JAR file containing the Action Processor queries. This JAR file is located as follows:  
`InstallDir\WEB-INF\lib\com.tibco.ipe.workspace.lib-version.jar`  
 where:
  - *InstallDir* is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: `C:\Tomcat\webapps\ipe-web`.
  - *version* is the version number of your release of iProcess Workspace Lite.
3. Make a copy of the `CaseList.query.xml` file.
4. Rename of the copy. In this example, rename it `CustomCaseList.query.xml`.
5. Copy your custom query file to the following directory (you will need to create the `\query` directory if this is your first custom query):

`InstallDir\WEB-INF\classes\query`

The query file can actually be placed anywhere within the class path. This example shows where to copy the query file when using Tomcat; its class loader looks for resource files in `\WEB-INF\classes`. If using a different application server, change the path accordingly.

6. Open the query file you have renamed.
7. Locate the **<sso:SortFields>** element. Within this element, you can add one or more **<sso:vSortField>** elements, one for each sort field you want to specify.

The following shows the elements to add to sort on the **Description** column, in ascending order (with the added elements shown in bold):

---

```

      :
      :
    <sso:SortFields>
      <sso:vSortField>
        <sso:FieldName>SW_CASEDESC</sso:FieldName>
        <sso:IsAscending>true</sso:IsAscending>
        <sso:SortTypeAs>swTextSort</sso:SortTypeAs>
      </sso:vSortField>
    </sso:SortFields>
      :
      :

```

---

You can add additional sort fields by adding additional `<sso:vSortField>` element sets under the `<sso:SortFields>` element.

8. Save and close the query file.

9. Open the following file:

`InstallDir\WEB-INF\web.xml`

10. Locate the **listCasesQuery** parameter name:

---

```

<init-param>
  <param-name>listCasesQuery</param-name>
  <param-value>/query/CaseList.query.xml</param-value>
</init-param>

```

---

11. Change:

`/query/CaseList.query.xml`

to:

`/query/CustomCaseList.query.xml`

This causes your custom query to be used instead of the one provided with the application.

Note - It's important that the path begin with '/', and that the path be relative to the `classes` directory.

12. Save and close the `web.xml` file.

13. Restart your application server.

14. Run the iProcess Workspace Lite application.

## Customizing the Sort Order on the Work Item List

To specify a custom sort order on the work item list, you will modify the following file:

– web.xml

The web.xml file contains the following parameters that can be modified to change the default sort order on the work item list:

---

```
<!-- parameters for defining the default work item list sort -->
<!-- settings can be commented out -->
<init-param>
  <param-name>defaultSortFieldName</param-name>
  <param-value>SW_CASENUM</param-value>
</init-param>
<init-param>
  <param-name>defaultSortIsAscending</param-name>
  <param-value>true</param-value>
</init-param>
<init-param>
  <param-name>defaultSortType</param-name>
  <param-value>swNumericSort</param-value>
</init-param>
```

---

Using this method, you can only sort on a single column at one time.

The sort parameters consist of the following:

- **defaultSortFieldName** - Identifies the field/column on which you want to sort.
- **defaultSortIsAscending** - Specifies whether the sort order will be ascending or descending.
- **defaultSortType** - Specifies the type of data the sort order is based on.

These are described in more detail below.

### defaultSortFieldName

You can sort on most of the columns in the work item list. Note, however, that when you specify the column to sort on, you do not use the column header label shown in the list — you must use the "system field" that represents that column information. For example, instead of using "Process" in the **defaultSortFieldName** parameter, you use the SW\_PRODESC system field:

---

```
<init-param>
  <param-name>defaultSortFieldName</param-name>
  <param-value>SW_PRODESC</param-value>
</init-param>
```

---

The following table lists the system fields used to sort on the columns in the work item list:

Table 3: System Fields Used to Sort the Work Item List

Column	System Field	Data Type
Locked	SW_LOCKER	Text
Id	SW_CASENUM	Numeric
Process	SW_PRODESC	Text
Case	SW_CASEDESC	Text
Task	SW_STEPDESC <sup>1</sup>	Text
Priority	SW_PRIORITY	Numeric
Arrived	SW_ARRIVAL	DateTime
Deadline	SW_DEADLINE	DateTime

1. "Tasks" and "Steps" are synonymous — in an older process designer, they were called "steps"; in TIBCO Business Studio, they are called "tasks".

### defaultSortIsAscending

The **defaultSortIsAscending** parameter specifies whether the sort order is ascending or descending. Pass "true" to sort in ascending order; pass "false" to sort in descending order.

---

```
<init-param>
  <param-name>defaultSortIsAscending</param-name>
  <param-value>true</param-value>
</init-param>
```

---

defaultSortType

The **defaultSortType** parameter specifies the data type to use when sorting on the column specified in the **defaultSortFieldName** parameter. You need to specify the constant shown in the table below for the type of data in the column you are sorting — see the table on [page 23](#) that lists the data types for each column.

The following are the available constant values for the **defaultSortType** parameter:

Table 4: Sort Type Constants for Sorting Work Item List

Constant	Description
swDateTimeSort	Sort as date/time
swNumericSort	Sort as real number
swTextSort	Sort as text

For example:

```
<init-param>
  <param-name>defaultSortType</param-name>
  <param-value>swNumericSort</param-value>
</init-param>
```

Case Column Exception

The **Case** column on the work item list contains text data (i.e., the SW\_CASEDESC system field is a text field). You can, however, specify that the data in the **Case** column be sorted using a different data type. For example, you can sort the data in the **Case** column as numeric data (by specifying swNumericSort in the **defaultSortType** parameter).

Note, however, that if the **Case column** does not contain something readily convertible to the specified data type, the sort results may not be what you expect.

Procedure for Customizing the Sort Request for the Work Item List

The following provides step-by-step instructions for customizing the sort order of a work item list.



Assume the following for this example:

- You are specifying that the work item list be sorted on the case description (i.e., the **Case** column) — therefore, you will sort on the SW\_CASEDESC system field.
- The descriptions should appear in alphabetical order — therefore, you will specify that the sort be in ascending order.
- As the description is text, you must specify a "sort type" of **swTextSort**.

Using the information in the preceding sections, you should be able to modify this example to customize the sort for any of the columns on the work item list.

1. Ensure that your application server is stopped.

2. Open the following file:

*InstallDir*\WEB-INF\web.xml

where:

— *InstallDir* is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: C:\Tomcat\webapps\ipe-web.

3. Locate the **defaultSortFieldName** parameter:

---

```
<init-param>
  <param-name>defaultSortFieldName</param-name>
  <param-value>SW_CASENUM</param-value>
</init-param>
<init-param>
  <param-name>defaultSortIsAscending</param-name>
  <param-value>true</param-value>
</init-param>
<init-param>
  <param-name>defaultSortType</param-name>
  <param-value>swNumericSort</param-value>
</init-param>
```

---

4. In the value parameter for the **defaultSortFieldName** parameter, specify the system field that represents the column on which you want to sort the work item list.

For a list of the available system fields, and the columns to which they apply, see [defaultSortFieldName on page 22](#).

5. In the value parameter for the **defaultSortIsAscending** parameter, specify whether the sort should be in ascending or descending order — "true" for ascending; "false" for descending.
6. In the value parameter for the **defaultSortType** parameter, specify the constant for the desired data type to be used in the sort.

For a list of the default data types for each of the columns/system fields, see the table in [defaultSortFieldName on page 22](#).

For a list of the available data types and their associated constants, see the table in [defaultSortType on page 24](#).

7. Save and close the `web.xml` file.
8. Restart your application server.
9. Run the iProcess Workspace Lite application.

## Adding a Custom Filter

---

By default, all work items and cases to which the user has access are listed in the work item and case lists.

If desired, you can customize the appropriate action requests to include a filter expression to specify which work items and cases to display in the work item and case lists, respectively. For example, you may only want the cases that were started earlier than a certain date to be displayed — you can filter the case list so that only those cases are displayed.

Only the work item and case lists can be filtered. To specify a custom filter, you will modify the appropriate action request file, as follows:

- Work item list: `WorkItemList.query.xml`
- Case list: `CaseList.query.xml`

The following subsections describe details of specifying a custom filter.



The work item list contains filter fields that allow the user to enter a filter expression from the UI so that only the desired work items are displayed. The case list does not contain filter fields. Therefore, the user cannot filter the case list.

Adding a custom filter using the procedure described here allows you to force a filter on the work item or case list. The custom filter is automatically imposed on the list when it is displayed.

### Filter Request

A customized filter request is accomplished by including a filter expression inside the `<sso:FilterExpression>` element in the action request:

---

```
<sso:FilterExpression>SW_CASEDESC = "BR*"</sso:FilterExpression>
```

---

The filter expression consists of the following components:

- **System field** - Identifies the data on which you want to filter. For example, the `SW_CASEDESC` system field contains the case description.
- **Operator** - Specifies how to compare the value in the system field with the provided expression. For example, `"="` specifies to return the work items or cases where the value in the system field "equals" the value passed in the filter expression.

- **Filter expression** - The value to compare with the value in the specified system field to determine if each work item or case should be displayed in the list.

These are explained in more detail in the following subsections.

System Field

A system field contains information about a work item or case. For example, the SW\_STARTED system field contains the date and time the case was started, and SW\_ARRIVAL contains the date and time the work item arrived in the work queue.

The following tables show the system fields used to filter on each of the columns in the work item and case lists:

Table 5: System Fields Used to Filter Work Item Lists

Column	System Field	Data Type
Locked	SW_LOCKER	Text
Id	SW_CASENUM	Numeric
Process	SW_PRODESC	Text
Case	SW_CASEDESC	Text
Task	SW_STEPDESC <sup>1</sup>	Text
Priority	SW_PRIORITY	Numeric
Arrived	SW_ARRIVAL	DateTime
	SW_ARRIVALDATE	Date
Deadline	SW_DEADLINE	DateTime
	SW_DEADLINEDATE	Date
	SW_EXPIRED	Numeric <sup>2</sup>

1. "Steps" and "Tasks" are synonymous — in an older process designer, they were called "steps"; in TIBCO Business Studio, they are called "tasks".
2. 1 = has an expired deadline; 0 = has a deadline that has not expired, or a deadline had not been specified for the work item.

Table 6: System Fields Used to Filter Case Lists

Column	System Field	Data Type
Id	SW_CASENUM	Numeric
Description	SW_CASEDESC	Text
Started	SW_STARTED	DateTime
	SW_STARTEDDATE	Date
Completed	SW_TERMINATED	DateTime
	SW_TERMINATEDDATE	Date
Manage <sup>1</sup>	n/a	n/a

1. You cannot filter on the **Manage** column.



Although it doesn't appear on the case list as a column, another very useful system field on which you can filter is the SW\_STATUS system field. This system field indicates whether or not the case is currently *active* (SW\_STATUS = "A") or *closed* (SW\_STATUS = "C").

## Operator

The operator specifies how to evaluate the filter expression. The following operators are allowed:

Table 7: Operators Used in Filter Expressions

Operator	Description
=	Equals
<&gt;	Not equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to

For example, the following filter expression specifies to return all cases that were started on or before 13-Aug-2008:

```
<sso:FilterExpression>SW_STARTEDDATE &lt;= !13/08/2008!</sso:FilterExpression>
```

Filter Expression

The filter expression is the value to which the system compares the value in the system field to determine if the work item or case should be returned by the server and shown in the list.

The data type (which is shown for each system field in the tables in [System Field on page 28](#)) dictates the way in which the filter expression must be passed in the Action Processor query, as follows:

Table 8: Filter Expression Data Types

Data Type	Description
Text	Consists of any number of letters, numbers, or special characters (e.g., #, \$, etc.).  Text values must be enclosed in double quotes.  Case is not significant when entering a text filter value.  Example: SW_CASEDESC = "Carriage bolts 66-UF7"
Numeric	Consists of one or more of the numbers 0-9.  Do not enclose numeric values in quotes.  Example: SW_PRIORITY &gt; 10
DateTime	Consists of both the date and time, separated by a space.  DateTime values must be enclosed in double quotes.  The format of the date (MM/DD/YYYY or DD/MM/YYYY) depends on the setting on the iProcess Engine to which you are connected.  Example: SW_STARTED &lt; "14/03/2009 17:00:00"

Table 8: Filter Expression Data Types

Data Type	Description
Date	<p>Although the Arrived and Deadline columns on the work item list, and the Started and Completed columns on the case list, display DateTimes, you can filter the lists using only the date portion by using the appropriate Date system fields (see the tables in <a href="#">System Field on page 28</a>).</p> <p>Date values must be enclosed in ! characters.</p> <p>The format of the date (MM/DD/YYYY or DD/MM/YYYY) depends on the setting on the iProcess Engine to which you are connected.</p> <p>Example: SW_STARTEDDATE &amp;lt;= !14/03/2009!</p>

Wild Card Characters

The "\*" wild card character can be used in text (string) filter expressions. It matches zero or more characters.

For example, to list all work item whose task name begins with "R", pass this filter expression in the query:

– SW\_STEPDESC = "R\*"

And to list all cases whose case description contains the value "300D", pass this filter expression in the query:

– SW\_CASEDESC = "\*300D\*"

Note that case number (SW\_CASENUM) is an anomaly in that it is a numeric value, and it can be passed as follows:

– SW\_CASENUM = 51

However, you can use the wild card character in your filter expression that includes case number, but if you do, you must enclose the number in double quotes (i.e., treat it like text). For example:

– SW\_CASENUM = "5\*"

Current Date System Field

There is a special system field that represents the current date: SW\_DATE. For example, the following would cause only the cases that were started today to be shown in the case list:

<sso:FilterExpression>SW\_STARTEDDATE = SW\_DATE</sso:FilterExpression>

### Filtering on Empty Fields

A special field is available that you can use to test for empty fields. The `SW_NA` field is used to determine if a value is assigned to the field to which you are comparing it.

For example, to return all work items that are *not* locked (i.e., there is no value in the `SW_LOCKER` system field), pass the following in the filter expression:

— `SW_LOCKER = SW_NA`

To return all work items that *are* locked (i.e., there is a value in the `SW_LOCKER` system field), pass the following in the filter expression:

— `SW_LOCKER &lt;&gt; SW_NA`

## Customizing Filter Request Procedure

The following provides step-by-step instructions for passing a custom filter in the Action Processor query. Assume the following for this example:

- You are implementing a custom filter on the case list — therefore, you will modify the `CaseList.query.xml` file.
- You are specifying that only cases that were started prior to today be listed, therefore, you will filter on the `SW_STARTEDDATE` system field.

Using the information in the preceding sections, you should be able to modify this example to fit the list and system fields for which you are specifying a custom filter.

To customize the action processor request for the example described above, follow these steps:

1. Ensure that your application server is stopped.
2. Extract the files from the JAR file containing the Action Processor queries to a temporary directory. This JAR file is located as follows:

`InstallDir\WEB-INF\lib\com.tibco.ipe.workspace.lib-version.jar`

where:

— `InstallDir` is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: `C:\Tomcat\webapps\ipe-web`.

— `version` is the version number of your release of iProcess Workspace Lite.

3. Make a copy of the appropriate query file (either `CaseList.query.xml` or `WorkItemList.query.xml`) (the query files are located in the `\query` directory).



For this example, you will make a copy of the `CaseList.query.xml` file.

4. Rename of the copy. In this example, you will rename it `CustomCaseList.query.xml`.
5. Copy your custom query file to the following directory (you will need to create the `\query` directory if this is your first custom query):

`InstallDir\WEB-INF\classes\query`

The query file can actually be placed anywhere within the class path. This example shows where to copy the query file when using Tomcat; its class loader looks for resource files in `\WEB-INF\classes`. If using a different application server, change the path accordingly.

6. Open the query file you have renamed.
7. Locate the `<sso:FilterExpression />` element.
8. Insert your filter expression within the `<sso:FilterExpression />` element (note that you will need to add the closing element).

For this example, add the following:

---

```
<sso:FilterExpression>
    SW_STARTEDDATE < SW_DATE
</sso:FilterExpression>
```

---

This specifies to return all cases whose start date is earlier than today.



Note that if you were customizing a filter request for the work item list, you would find that the `<sso:FilterExpression />` element already has a value. For information about this value, see [Work Item List Custom Filter Query on page 34](#).

9. Save and close your query file. In this example, you are saving the `CustomCaseList.query.xml` file.
10. Open the following file:
 

`InstallDir\WEB-INF\web.xml`
11. Locate the `listCasesQuery` parameter name:

---

```
<init-param>
    <param-name>listCasesQuery</param-name>
    <param-value>/query/CaseList.query.xml</param-value>
</init-param>
```

---

12. Change:

```
/query/CaseList.query.xml
```

to:

```
/query/CustomCaseList.query.xml
```

This causes your custom query to be used instead of the one provided with the application.

Note - It's important that the path begin with '/', and that the path be relative to classes.

13. Save and close the web.xml file.

14. Restart your application server.

15. Run the iProcess Workspace Lite application.

### Work Item List Custom Filter Query

If you customize a filter query for the work item list (i.e., you are adding a custom query to the WorkItemList.query.xml file), the `<sso:FilterExpression />` element will already contain a value, as follows:

---

```
<sso:FilterExpression>__FILTER_EXPRESSION__</sso:FilterExpression>
```

---

This value represents the value entered by a user into one of the filter fields on the work item list. It cannot be removed, otherwise the user loses the capability to filter via the UI.

To add a custom filter query to the work item list, while maintaining UI filtering capability, add your custom query before the "\_\_FILTER\_EXPRESSION\_\_" notation, and AND it with the UI-entered value, as follows:

---

```
<sso:FilterExpression>SW_EXPIRED=1 AND __FILTER_EXPRESSION__</sso:FilterExpression>
```

---

This example causes the work item list to contain only work items that have an expired deadline, and that also satisfy a value the user may enter in one of the filter fields on the work item list.

## Chapter 4 **Adding an Additional Language**

This chapter describes how to add an additional language to the TIBCO iProcess Workspace Lite application.

### Topics

---

- [Introduction, page 36](#)
- [Adding Language Procedure, page 37](#)

## Introduction

---

The user of the iProcess Workspace Lite application can choose which language to display the application in by making a language selection from within the browser. This is done as follows:

- From Internet Explorer: **Tools > Internet Options > General Tab > Languages**
- From Firefox: **Tools > Options > Advanced > General Tab > Choose**

The iProcess Workspace Lite application uses the Accept-Language header from the browser to determine which language it should use when displaying the application. Based on this information, the application will use a specific "properties" file that contains strings that are used throughout the application.

The iProcess Workspace Lite application is distributed with the English version of the properties file, as follows:

```
InstallDir\WEB-INF\classes\workspaceLite.properties
```

where

- *InstallDir* is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: C:\Tomcat\webapps\ipe-web.

If the user selects a different language from within the browser, iProcess Workspace Lite will look for the properties file for that language. If there is a properties file for that language, the application will include the translated strings from that file. The names of the properties files (other than the English one provided), must be in the following format:

```
workspaceLite_XX.properties
```

where:

- *xx* 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>

For example, if the user selects any type of French (fr, fr-CA, fr-MC, etc.), the application will look for a `workspaceLite_fr.properties` file, and it will display the application using the translated strings in that file.

## Adding Language Procedure

---

To add an additional language so that the user can display the iProcess Workspace Lite application in that language, follow these steps:

1. Make a copy of the English properties file, which is located as follows:

`InstallDir\WEB-INF\classes\workspaceLite.properties`

2. Rename the properties file to include the language, as follows:

`workspaceLite_xx.properties`

where:

- `xx` 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>

3. For each of the strings in the newly created properties file, translate it to the appropriate language.
4. Save the file in the `InstallDir\WEB-INF\classes\` directory.
5. Stop your application server (if it's not already stopped), then restart it.
6. From the browser, select the language for the properties file that was added.
7. Run the iProcess Workspace Lite application.



## Chapter 5 Customizing Layout

This chapter provides an introduction to customizing the TIBCO iProcess Workspace Lite application.

### Topics

---

- [Customizing Layout, page 40](#)
- [Adding CDQPs to the Work Item List, page 45](#)
- [Removing a Column From the Work Item List, page 51](#)

## Customizing Layout

---

The layout of the TIBCO iProcess Workspace Lite application is accomplished with the use of XSLT files. You are free to customize the existing XSLT files, or create your own to get the desired layout.

XSLTs allow a vast amount of control in customizing layouts. However, this section does not provide a tutorial in using XSLTs. It merely describes how they are used in iProcess Workspace Lite.

The XSLT to use for each request is called out in the *InstallDir*\WEB-INF\web.xml file. For example, the following shows the XSLT used for displaying the work item list:

---

```
    .
    .
    .
    <init-param>
        <param-name>workItemListQuery</param-name>
        <param-value>/query/WorkItemList.query.xml</param-value>
    </init-param>
    <init-param>
        <param-name>workItemList2HtmlXslt</param-name>
        <param-value>/xslt/WorkItemList2Html.xslt</param-value>
    </init-param>
    .
    .
    .
```

---

The path to the XSLT references are relative to the *InstallDir*\WEB-INF\classes\ directory.

Note that in this release, it is only possible to register one XSLT per request type, therefore any changes will apply to all users.

The basic structure of the XSLT is as follows:



---

```

<xsl:stylesheet version="1.0"

<!-- This is general configuration, such as xsl:param, -->
<!-- xsl:output, etc. You should not change this. -->

<xsl:template match="/">

<html>

<!-- The HTML for the page, with XSL elements embedded. -->

</html>

</xsl:template>

<xsl:template match="ap:Status">

<!-- Other xsl:templates called from the root template. -->

</xsl:template>

```

---

Essentially, XSLTs provide a means of embedding data from the Action Processor request into your own HTML. To customize the layout using XSLTs, the following procedure is recommended:

1. Design a static HTML page that has the layout you want with hard-coded data that will later be retrieved by the application.
2. Make a copy of the provided XSLT for the desired query. For example, to customize the XSLT for displaying the case audit trail, copy the `Audit2Html.xslt` file.
3. In the copied XSLT, replace the HTML in the root and other templates with your own.
4. Save the XSLT with a new name, then reference that new name in the `InstallDir\WEB-INF\web.xml` file for the query to which it applies.

## Workspace Lite XSLT Library

Some of the functionality in the provided XSLTs is obtained from the following XSLT library:

```
InstallDir\WEB-INF\classes\xslt\workspace.xslt
```

The functionality available in this library may also be useful in customizations. You can use the functions in this library by including the following in your XSLT file:

---

```
<xsl:include href="xslt/workspace.xslt"/>
```

---

In particular, the templates for date and timestamp formatting may be useful:

- dateFormatter
- timestampFormatter

These are used for formatting the **Arrived** column timestamp and **Deadline** column date in the work item list, as follows:

---

```
<td>
  <xsl:call-template name="timestampFormatter">
    <xsl:with-param name="dateString">
      <xsl:value-of select="sso:Arrived"/>
    </xsl:with-param>
    <xsl:with-param name="locale">
      <xsl:value-of select="$locale"/>
    </xsl:with-param>
  </xsl:call-template>
</td>
<td>
  <xsl:choose>
    <xsl:when test="sso:Deadline = '3000-12-31T23:15:00.0000000'">
      <xsl:text> - </xsl:text>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name="dateFormatter">
        <xsl:with-param name="dateString">
          <xsl:value-of select="sso:Deadline"/>
        </xsl:with-param>
        <xsl:with-param name="locale">
          <xsl:value-of select="$locale"/>
        </xsl:with-param>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</td>
```

---

## User Customization XSLT

To provide a separation of user customization of XSLT from the iProcess Workspace Lite's standard XSLT library, the following XSLT file is provided:

`InstallDir\WEB-INF\classes\xslt\UserCustomization.xslt`

where:

- *InstallDir* is the installation directory for iProcess Workspace Lite. For example: `C:\Tomcat\webapps\ipe-web`.



All user customization of XSLT should be done in `UserCustomization.xslt`, not in the iProcess Workspace Lite standard XSLT library (`workspace.xslt`). This allows Technical Support to separate customer configuration from the standard product.

The `UserCustomization.xslt` file contains Boolean variables used to control the behavior of other XSLTs (`workspace.xslt` includes `UserCustomization.xslt`). It contains the following variables:

Table 9: UserCustomization.xslt File Variables

Variable	Description
<b>hideStatus</b>	<p>Setting this variable to true hides the Action Processor status information that is displayed on the top of each page. This includes all information enclosed in the box entitled "Status of the last request".</p> <p>Default = false.</p>
<b>hideLinks</b>	<p>Setting this variable to true hides the following links that are displayed on various pages in iProcess Workspace Lite:</p> <ul style="list-style-type: none"> <li>• The <b>View Work Queue:</b> <i>name</i> and <b>View All Work Queues</b> links that are displayed after a work item is cancelled, closed or submitted.</li> <li>• The <b>View All Work Queues</b> link that is displayed after starting a case of a process.</li> <li>• The <b>View processes</b> link displayed on the bottom of the case list.</li> <li>• The <b>Show cases of</b> <i>name</i> and <b>View processes</b> links that are displayed on the case audit trail page.</li> <li>• The <b>View cases of</b> <i>name</i> and <b>View processes</b> links that are displayed on the bottom of the page after closing a case of a process.</li> </ul> <p>Default = false.</p>

Table 9: UserCustomization.xslt File Variables

Variable	Description
hideFilters	Setting this variable to true to hides the filter fields on all work item lists, preventing the user from being able to filter work item lists.  Default = false.


## Adding CDQPs to the Work Item List

Columns can be added to the work item list that show the values in CDQPs (data fields) for each of the work items in the list.

For example, you may want to include a column that shows the values in a CustName field, which would allow you to filter and sort the work item list on a customer name.

This section provides step-by-step instructions to add a new column to the work item list that shows the value in a CDQP field. In this example, a "Purpose" column will be added between the current Task and Priority columns, and the value in the "PURPOSE" CDQP field will be shown in the new column for each work item, as follows:

Figure 3 Adding CDQPs to the Work Item List example



Locked? ^ v	Id ^ v	Process ☒ v	Case ^ v	Task ^ v	Purpose ^ v	Priority ^ v	Arrived ^ v	Deadline ^ v
No	1353	Company Car Allocation	Ford van 66298	Car service request	Weekly car pool	50	29/06/2009 15:26:00	-
No	1403	Company Car Allocation	Saturn 88309	Get Manager's authority	Company picnic	50	30/06/2009 10:10:00	-
No	1453	Company Car Allocation	Honda civic 88910	Get Manager's authority	Just for fun	50	30/06/2009 15:28:00	-

To add the example Purpose column to the work item list, follow these steps:

1. Ensure the desired field (PURPOSE in this example) has been added as a CDQP and mapped to the desired work queue in which you want to view it.

For information about adding and mapping CDQPs, see the *TIBCO iProcess swutil and swbatch Reference Guide*.

2. Open the following file with a text editor:

`InstallDir\WEB-INF\classes\workspacelite.properties`

where *InstallDir* is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: C:\Tomcat\webapps\ipe-web.

This file contains all text strings used in the application, which allows for easier localization of the application.

Note that the `workspacelite.properties` file is the English version; if your iProcess Workspace Lite application is localized, open and update the appropriate file in the same directory (e.g., `workspacelite_de.properties` for German; `workspacelite_fr.properties` for French, etc.).

3. Locate the section that contains strings for the work item list (it is labeled "# Messages used by WorkItemList2Html / WorkItemList2Atom").
4. In the work item list string section, add an entry for the new column, plus an entry that will be used as "alt" text for the filter field you will be adding later. For example:

```
# Messages used by WorkItemList2Html / WorkItemList2Atom
workItemList_title = Work Item List
workItemList_label = Work Item List:
workItemId_title = Id
workItemIdFilter_title = Filter on work item id
processDescFilter_title = Filter on process description
caseDesc_title = Case
caseDescFilter_title = Filter on case description
taskDesc_title = Task
taskDescFilter_title = Filter on task description
CDQP_title = Purpose
CDQPFilter_title = Filter on CDQP field
.
.
.
```

This will cause the new column header to be entitled "Purpose", and the new column filter field "alt" text will be "Filter on CDQP field".

5. Save and close the `workspacelite.properties` file.
6. Open the following file:

```
InstallDir\WEB-INF\classes\xslt\WorkItemList2Html.xslt
```

where *InstallDir* is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: C:\Tomcat\webapps\ipe-web.

7. Locate the table header elements (<th>) that cause the **Task** column to appear in the work item list (you want to place our new column directly after the **Task** column). It will look like the following:

---

```
<th width="35%" scope="col">
  <xsl:value-of select="java:com.tibco.ipe.internal.l10n.Messages.get($locale,
    'taskDesc_title') " />
  <br/>
  <xsl:call-template name="sortControls">
    <xsl:with-param name="nextSortFieldName">SW_STEPDESC</xsl:with-param>
    <xsl:with-param name="nextSortFieldType">swTextSort</xsl:with-param>
  </xsl:call-template>
</th>
```

---

8. Copy the table header entry for the **Task** column (as shown above), then paste it in the appropriate location to insert your new column.

In this example, you will paste it between the table header elements for the **Task** and **Priority** columns.

9. In the table header entry you just pasted, modify the variable for the column title to match the one you added to the `workspacelite.properties` file in [step 4](#) (“CDQP\_title” in this example).
10. Change the “nextSortFieldName” value to the name of your CDQP field (“PURPOSE” in this example).
11. Change the “nextSortFieldType” value to the data type for the CDQP field you specified in “nextSortFieldName”. The available values are `swTextSort`, `swNumericSort`, `swDateTimeSort`, `swDateSort` and `swTimeSort`. If the data type specified doesn’t match the type for the CDQP field specified, the sort results will be unpredictable.

The new table header entry should look like the following (with the changes shown in bold type):

---

```
<th width="35%" scope="col">
  <xsl:value-of select="java:com.tibco.ipe.internal.l10n.Messages.get($locale,
    'CDQP_title') " />
  <br/>
  <xsl:call-template name="sortControls">
    <xsl:with-param name="nextSortFieldName">PURPOSE</xsl:with-param>
    <xsl:with-param name="nextSortFieldType">swTextSort</xsl:with-param>
  </xsl:call-template>
</th>
```

---

12. Optionally modify the column width percentages to accommodate the new column you are adding. Note that the browser will dynamically adjust the column widths as much as it can so that the table stays within the browser window.
13. Near the beginning of the `WorkItemList2Html.xslt` file, locate the "Filter" parameter declarations, and add a new one for the CDQP field you are adding to the work item list. The naming convention for this parameter is:

`<CDQPFieldName>Filter`

For example, since you are adding one for the PURPOSE CDQP field, the new entry would look as follows:

---

```

        :
        :
<xsl:param name="caseDescFilter"/>
<xsl:param name="taskDescFilter" />
<xsl:param name="PURPOSEFilter"/>
<xsl:param name="workItemPriorityFilter" />
<xsl:param name="workItemArrivedFilter" />
        :
        :

```

---

14. Locate the table header entry that causes the **Task** column filter field to appear in the work item list (you want to place our new filter field directly after the **Task** column filter field). It will look like the following:

---

```

<th width="35%" scope="col" style="padding-right: 5px;">
  <xsl:element name="input">
    <xsl:attribute name="name">taskDescFilter</xsl:attribute>
    <xsl:attribute name="alt">
      <xsl:value-of select="java:com.tibco.ipe.internal.ll0n.Messages.get($locale,
        'taskDescFilter_title')"/>
    </xsl:attribute>
    <xsl:attribute name="value">
      <xsl:value-of select="$taskDescFilter"/>
    </xsl:attribute>
    <xsl:attribute name="style">width: 100%;</xsl:attribute>
  </xsl:element>
</th>

```

---

15. Copy the table header entry for the **Task** column filter field (as shown above), then paste it in the appropriate location to insert your new filter field.



In this example, you will paste it between the table header elements for the **Task** and **Priority** column filter fields.

16. In the table header entry you just pasted, modify the following:
  - a. Change the name of the filter control on the third line to the name you specified in [step 13](#). In this example, it's "PURPOSEFilter".
  - b. On the fifth line, change the "alt" text value to the parameter name you specified in the `workspacelite.properties` file in [step 4](#) (CDQPFilter\_title in this example). (The "alt" text can be used by screen readers to identify the filter field.)
  - c. Change the name of the filter control on the eighth line to the name you specified in [step 13](#). In this example, it's "PURPOSEFilter".

The new table header entry should look like the following (with the changes shown in bold type):

---

```
<th width="10%" scope="col" style="padding-right: 5px;">
  <xsl:element name="input">
    <xsl:attribute name="name">PURPOSEFilter</xsl:attribute>
    <xsl:attribute name="alt">
      <xsl:value-of select="java:com.tibco.ipe.internal.l10n.Messages.get($locale,
        'CDQPFilter_title')"/>
    </xsl:attribute>
    <xsl:attribute name="value">
      <xsl:value-of select="$_PURPOSEFilter">
    </xsl:attribute>
    <xsl:attribute name="style">width: 100%;</xsl:attribute>
  </xsl:element>
</th>
```

---

17. Optionally modify the column width percentages to accommodate the new filter field you are adding. Note that the browser will dynamically adjust the column widths as much as it can so that the table stays within the browser window.
18. Locate the table data elements (<td >) that populate the work item list table with values. The table data entry for the **Task** column (which displays the step description for the work item), appears as follows:

---

```
<td>
  <xsl:value-of select="sso:StepDescription"/>
</td>
```

---

19. Copy the table data entry for the **Task** column (as shown above), then paste it in the appropriate location to populate the table with data.

In this example, you will paste it between the table data elements for the **Task** and **Priority** columns.

20. In the table data entry you just pasted, change the value so that the value in the CDQP field is displayed in the table. It must be in the form:

`sso:CDQPs/sso:vCDQP[sso:FieldName=' CDQPFieldName']/sso:Value`

where *CDQPFieldName* is the name of the CDQP field whose value you want displayed in the work item list.

In this example, the name of the CDQP field is "PURPOSE".

The new table data entry should now appear as follows (with the changes shown in bold type):

---

```
<td>
  <xsl:value-of select="sso:StepDescription"/>
</td>
<td>
  <xsl:value-of select="sso:CDQPs/sso:vCDQP[sso:FieldName=' PURPOSE']/sso:Value"/>
</td>
<td>
  <xsl:value-of select="sso:Priority"/>
</td>
```

---

21. Save and close the `WorkItemList2Html.xslt` file.
22. Start your application server.
23. Start the iProcess Workspace Lite application and view the work item list.

## Removing a Column From the Work Item List

---

This section provides step-by-step instructions on how to remove a column from the work item list.

In this example, you will remove the **Priority** column, as well as its respective filter field.

To remove a column from the work item list, follow these steps:

1. Stop your Web Application Server.
2. Open the following file:

`InstallDir\WEB-INF\classes\xslt\WorkItemList2Html.xslt`

where *InstallDir* is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: `C:\Tomcat\webapps\ipe-web`.

3. Locate the table header elements (<th >) that cause the **Priority** column to appear in the work item list. It will look like the following:

---

```
<th width="10%" scope="col">
  <xsl:value-of select="java:com.tibco.ipe.internal.110n.Messages.get($locale,
    'workItemPriority_title') " />
  <br/>
  <xsl:call-template name="sortControls">
    <xsl:with-param name="nextSortFieldName">SW_PRIORITY</xsl:with-param>
    <xsl:with-param name="nextSortFieldType">swNumericSort</xsl:with-param>
  </xsl:call-template>
</th>
```

---

4. Either delete, or comment out, the table header entry for the **Priority** field (which is shown above).
5. Locate the table header entry that causes the **Priority** column filter field to appear in the work item list. It will look like the following:

---

```
<th width="10%" scope="col" style="padding-right: 5px;">
  <xsl:element name="input">
    <xsl:attribute name="name">workItemPriorityFilter</xsl:attribute>
    <xsl:attribute name="alt">
      <xsl:value-of select="java:com.tibco.ipe.internal.l10n.Messages.get($locale,
        'workItemPriorityFilter_title')"/>
    </xsl:attribute>
    <xsl:attribute name="value">
      <xsl:value-of select="$workItemPriorityFilter"/>
    </xsl:attribute>
    <xsl:attribute name="style">width: 100%;</xsl:attribute>
  </xsl:element>
</th>
```

---

6. Either delete, or comment out, the table header entry for the **Priority** filter field (which is shown above).
7. Locate the table data elements (<td >) that populate the work item list table with priority values. It will look like the following:

---

```
<td>
  <xsl:value-of select="sso:Priority"/>
</td>
```

---

8. Either delete, or comment out, the table data entry for the **Priority** field (which is shown above).
9. Save and close the WorkItemList2Html.xslt file.
10. Start your application server.
11. Start the iProcess Workspace Lite application and view the work item list.

The work item list will now appear as follows:

Figure 4 Removing a Column from the Work Item List Example

Locked?	Id ^ v	Process ☒ v	Case ^ v	Task ^ v	Arrived ^ v	Deadline ^ v
No	152	Brand orders	Carriage mounts 12CA00	Application	31/03/2009 11:49:00	-
No	102	Brand orders	Lipid grease AS77	Application	31/03/2009 13:40:00	-
No	1	Company Car Allocation	Ford van 29976	Request for Vehicle	31/03/2009 11:34:00	-



There are some other remnants of the **Priority** column, such as string values in the `workspacelite.properties` file, as well as parameter declarations in the `WorkItemList2Html.xslt` file. However, they do not need to be commented out nor deleted to successfully remove the **Priority** column from the work item list.



## Chapter 6      **Additional Customizations**

This chapter provides information about additional iProcess Workspace Lite customization tasks.

### Topics

---

- [Time-outs, page 56](#)
- [Using Custom Forms, page 57](#)
- [How To Embed TIBCO iProcess Workspace Lite in a Portal, page 66](#)
- [Customizing the Number of Cases Returned from the Server, page 69](#)

## Time-outs

---

By default, the iProcess Workspace Lite application times out of its session with the server in 20 minutes if there is no activity by the user. Also, a dialog is displayed after 18 minutes of inactivity, warning the user that their session with the server is about to time-out. This dialog provides screen-reader users the ability to read the time-out warning rather than experiencing an unexpected session timeout.

Both the time-out and the warning times can be modified as follows:

1. Open the following file:

```
InstallDir\WEB-INF\web.xml
```

where:

- *InstallDir* is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: C:\Tomcat\webapps\ipe-web.

2. Locate the **sessionTimeoutWarning** parameter name:

---

```
<init-param>
  <description>
    Milliseconds before warning session is about
    to timeout (1080000 = 18 mins)
  </description>
  <param-name>sessionTimeoutWarning</param-name>
  <param-value>1080000</param-value>
</init-param>
```

---

3. Change the number in the **<param-value>** element to the number of milliseconds of inactivity before the warning dialog is displayed.
4. Locate the **<session-timeout>** element:

---

```
<session-config>
  <session-timeout>20</session-timeout>
</session-config>
```

---

5. Change the number in the **<session-timeout>** element to the number of minutes of inactivity before a session timeout.
6. Save and close web.xml.
7. Restart the iProcess Workspace Lite application.

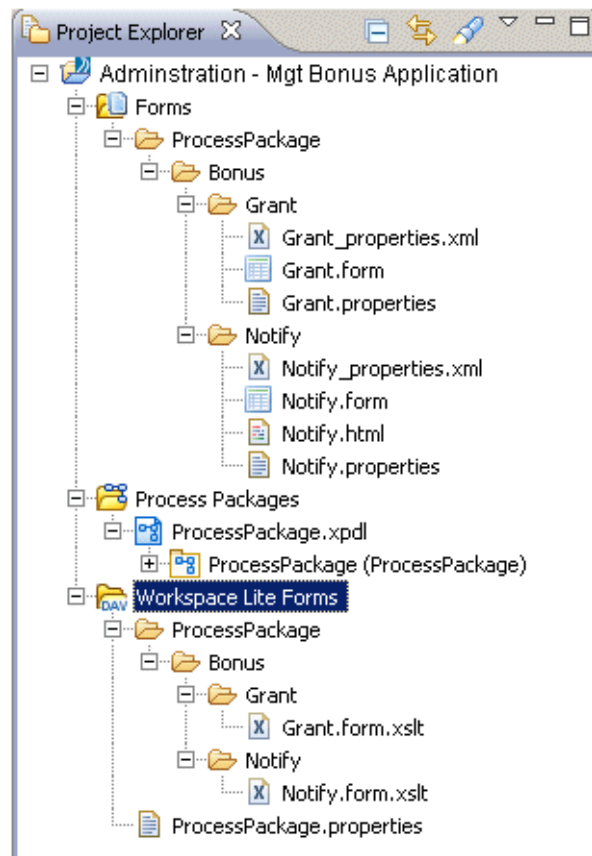


## Using Custom Forms

This section describes using custom TIBCO Forms (i.e., TIBCO Business Studio-produced forms with custom features) in the iProcess Workspace Lite application.

When a process is created in TIBCO Business Studio, *default* forms can be produced for each user task in the process. When default forms are created, a version of those forms are produced that are specifically configured to work in iProcess Workspace Lite. They are shown in the Project Explorer after you create default forms, as follows:

Figure 5 Default Forms in Project Explorer



The default forms can then be customized, using TIBCO Business Studio, to include enhanced functionality and look-and-feel features. For information about customizing forms, see the *TIBCO Business Studio Forms User's Guide*.

Note, however, if the customized forms are going to be used in an iProcess Workspace Lite application, the functionality and look-and-feel features that can be added are limited to those listed below because of the accessible nature of the iProcess Workspace Lite application.

### Supported Forms Elements

- **Controls:**
  - Button
  - Checkbox - works for booleans.
  - Date
  - Horizontal Pane
  - Hyperlink
  - Image
  - Label
  - Option List - static bound only.
  - Pass-through
  - Radio group - static bound only - works for text-type controls.
  - Text Area
  - Text Field
  - Time
  - Vertical Pane
- **Properties:**
  - Display Length
  - Maximum Length
  - Enabled
  - Label Visibility
  - Required
  - Visible

### Forms Elements Not Supported

- **Controls:**
  - Date-Time - the iProcess Engine only has Date and Time separately.

- **Properties:**
  - Background Color<sup>1</sup> - is not supported for accessibility reasons.
  - Font properties (all)<sup>1</sup> - is not supported for accessibility reasons.
  - Hint
  - Numeric - would require JavaScript.
  - Tab index
  - Value - to set a default value, use a script task.
- **Functionality:**
  - Validations
  - Rules



Note that custom form data field and parameter names (as opposed to labels) must meet the iProcess length restrictions rather than relying on truncation as part of the deployment.

## Deploying Custom Forms

After forms are customized using TIBCO Business Studio, they must be deployed to a WebDAV server.

When iProcess Workspace Lite is installed, a default WebDAV server is also installed in the following location:

```
ServerRoot\ipe-web\Forms\
```

where *ServerRoot* is the root directory for your Web Application Server. For example, if you are using Tomcat, it might be something like `C:\Tomcat\webapps`.

You can deploy your custom forms to this default WebDAV server, or you can deploy them to a different WebDAV server, if desired.



Note that it's important you consult your security personnel concerning the method of deploying forms. For further information, see [Security Issues Concerning Deployment on page 63](#).

The following provides the procedure for deploying custom forms from TIBCO Business Studio for use in iProcess Workspace Lite applications:

1. Generate default forms for each of the user tasks in your business process.

---

1. Skins can be used for customizing colors and fonts — for information about adding a skin, see [Adding a Skin on page 9](#).

For information about generating default forms, see the *TIBCO Business Studio iProcess Implementation Guide*.

2. Using the forms designer in TIBCO Business Studio, customize the forms to the extent possible for iProcess Workspace Lite applications.

For information about customizing forms, see the *TIBCO Business Studio Forms User's Guide*.

3. Deploy the custom forms as follows:
  - a. In the Deployment Server pane, right-click **Deployment Servers** and select **New > Server**.
  - b. Enter the desired name in the **Name** field.
  - c. Select "WebDAV Server" from the **Runtime** field drop-down list, then click **Next**.

The **Runtime Server Parameters** dialog is displayed.

- d. In the **Site URL** field, enter the URL to the WebDAV server that will be hosting your custom forms. This can be one of the following:
  - The built-in WebDAV server that is installed when you install iProcess Workspace Lite.
  - Another WebDAV server that you have installed somewhere else.

If you are using the built-in WebDAV server, its URL must be in the form:

`http(s)://<host>:<port>/ipe-web/Forms/`

where *<host>* is the name of the machine on which the WebDAV server is installed, and *<port>* is the port number used by the Web Application Server on *<host>*. (The */Forms/* path is relative to the root of the web application (\*WAR-FILE-ROOT\*)).



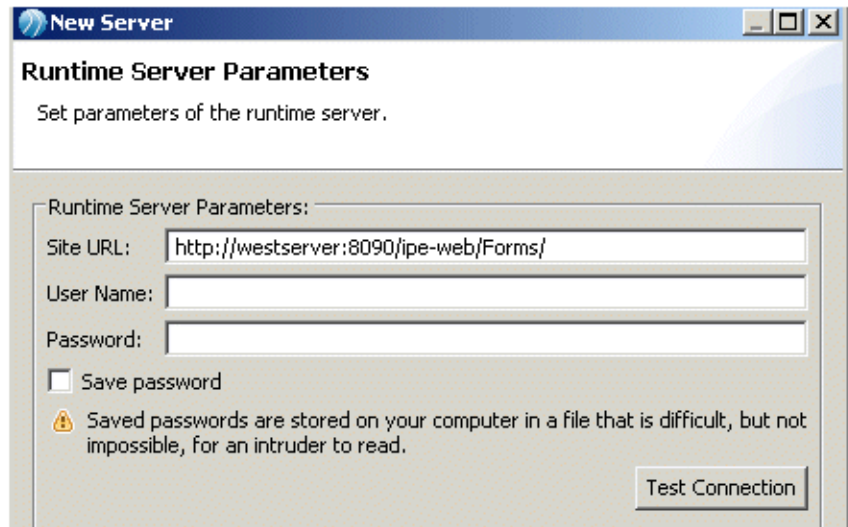
If you are using a Web Application Server that does not expand the .WAR file into a directory when the server is started (e.g., WebLogic), you will have an issue deploying forms to the built-in WebDAV server because, by default, a parameter in the `web.xml` file is pointing to a directory which assumes an expanded .WAR file.

You can work around this issue by either using a different WebDAV server, or by changing the value of the parameter in the `web.xml` file.

To change the `web.xml` parameter value, locate the **rootpath** parameter, under the **webdavforms** servlet record, and change it from "`*WAR-FILE-ROOT*/Forms/`" to the directory in which the forms were deployed. (Windows paths must be escaped, e.g., `c:\\temp\\forms`. UNIX paths do not need to be escaped, e.g., `/var/forms`.)

For example:

Figure 6 WebDAV Server URL Example



- e. Enter a user name and password in the **User Name** and **Password** fields, if required.

By default, a user name and password are not required if you are using the built-in WebDAV server.

Note that if a security constraint has been added to the iProcess Workspace Lite application's deployment descriptor (web.xml file), a user name and password must be supplied (for more information, see [Security Issues Concerning Deployment on page 63](#)).

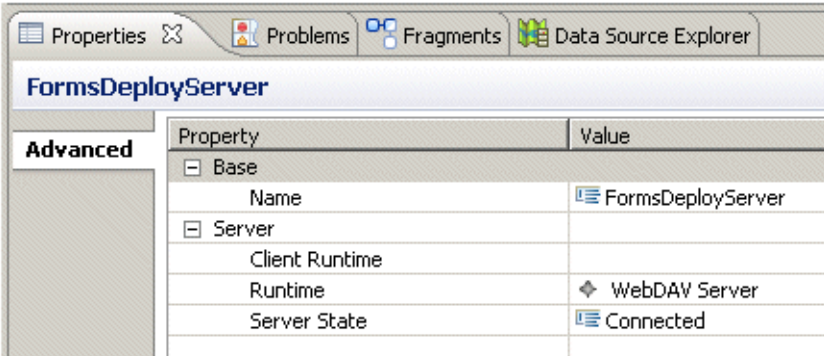
- f. Optionally click the **Test Connection** button to test connectivity with the WebDAV server. If you do not get a successful connection, ensure the specified URL is correct (and that the web application server is running on host).
- g. Click **Finish**.

The newly created deployment server is displayed in the Deployment Server pane.

- h. In the Deployment Server pane, right-click on the forms deployment server you just created and select **Connect**.

The Properties tab for the forms deployment server will indicate that TIBCO Business Studio has connected to the WebDAV server, as follows:

Figure 7 Forms Deployment Server Properties Tab

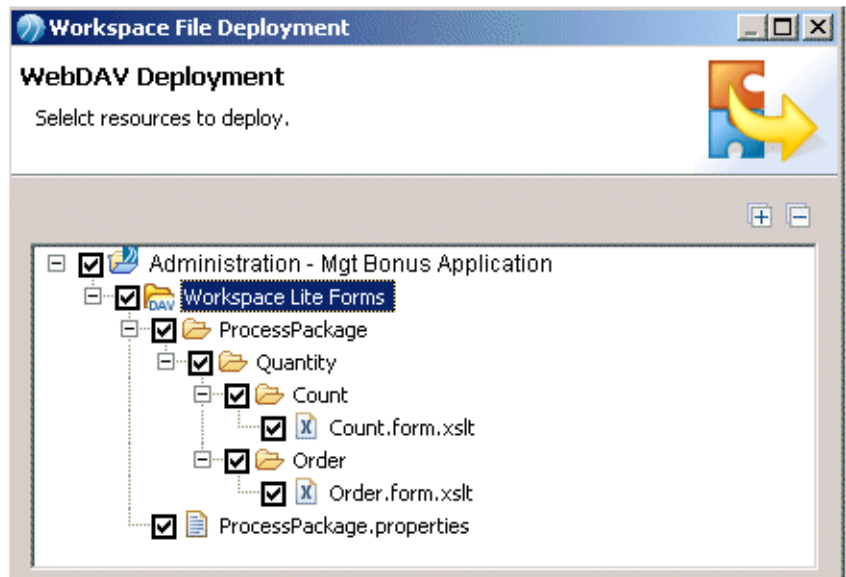


- i. In the Deployment Server pane, right-click on the forms deployment server you just created and select **Deploy Module ....**

The **Workspace File Deployment** dialog is displayed. This dialog presents a collapsible graphical display of all WebDAV 'Special' Folders, of which "Workspace Lite Forms" is one. This is the same Workspace Lite Forms Special Folder that was created in the Project Explorer when you created forms. If you drill down through the graphical display, the forms that will be deployed are shown.

- j. Click the **Select All** button to select all of the forms in the structure for deployment. For example:

Figure 8 Selecting Forms for Deployment



k. Click **Finish**.

Your custom forms are copied to the WebDAV server in the same structure as shown on the **Workspace File Deployment** dialog.

You should be able to now start cases and open work items in iProcess Workspace Lite using the custom forms.

## Security Issues Concerning Deployment

As described in [Deploying Custom Forms on page 59](#), a WebDAV server for forms is configured by default in iProcess Workspace Lite.

This default WebDAV server is configured in the application's deployment descriptor:

```
ServerRoot\ipe-web\WEB-INF\web.xml
```

where *ServerRoot* is the root directory for your Web Application Server. For example, if you are using Tomcat, it might be something like  
C:\Tomcat\webapps.

This method of deployment may be reasonable for development/testing environments, but not for production environments. It is highly recommended that your IT security personnel review your form deployment methodology.

The following are options that are available for forms deployment:

- You can enable security constraints on the URLs that are documented in the `web.xml` file for the WebDAV server. For details, consult your Web Application Server documentation — for Tomcat users, see:  
<http://tomcat.apache.org/tomcat-6.0-doc/realm-howto.html>
- You can also disable the WebDAV server altogether and deploy the forms ahead of time. This is the most secure approach and fits with the common enterprise practice of ensuring that all forms pass through a user acceptance environment before being deployed to a production environment. This can be done as follows:
  - a. Comment out (or delete) the **webdavForms** servlet entry, as well as the **servlet-mapping** elements in the `web.xml` file.
  - b. Copy all of your iProcess Workspace Lite form files (`.properties` and `.form.xslt` files) to the Forms sub-directory of the iProcess Workspace Lite application.

Also refer to the following website for additional information about security when deploying into a runtime environment:

<http://java.sun.com/javaee/5/docs/tutorial/doc/bncbe.html#bncbj>

### WebDAV Repository Root for Forms

The location on the file system at which forms content deployed via WebDAV is stored can be customized in the application's deployment descriptor file, `web.xml`, as follows:

1. Open the following file:

`InstallDir\WEB-INF\web.xml`

where:

— *InstallDir* is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: `C:\Tomcat\webapps\ipe-web`.

2. Locate the **webdavForms** servlet, and within that servlet, locate the **rootpath** parameter:



---

```

<servlet>
  <servlet-name>webdavForms</servlet-name>
  <servlet-class>com.tibco.ipe.webdav.WebdavServlet</servlet-class>
  <init-param>
    <description>
      name of the class that implements net.sf.webdav.WebdavStore
    </description>
    <param-name>ResourceHandlerImplementation</param-name>
    <param-value>
      com.tibco.ipe.webdav.LocalFileStore
    </param-value>
  </init-param>
  <init-param>
    <description>
      place where to store the webdavcontent on the filesystem
    </description>
    <param-name>rootpath</param-name>
    <param-value>*WAR-FILE-ROOT*/Forms/</param-value>
  </init-param>
  .
  .
  .

```

---

3. For the **rootpath** parameter, change the path in the **<param-value>** element to the location on the file system in which you want to store WebDAV content.

The special variable **\*WAR-FILE-ROOT\*** can be used (as it is in the default path shown above), which points to the web application root. Or you can specify an absolute path (with slashes in either direction) to the directory where WebDAV files are deployed.

If your web application server does not extract the web archive (.war file) into a directory (unlike Tomcat), then you will need to specify an absolute file system location instead of using the **\*WAR-FILE-ROOT\*** syntax.

4. Save and close web.xml.

## How To Embed TIBCO iProcess Workspace Lite in a Portal

---

TIBCO iProcess Workspace Lite can be embedded in a portal. For example, you may want to use something like iGoogle, which provides an environment where different data and applications can be brought together in a single application, or maybe an implementation of the Java Portlet specification such as WebLogic Portal. In either case the portal will provide the application container functionality, such as menus and navigation that is typically provided by TIBCO iProcess Workspace Lite itself.



For the sake of this description, the term “gadget” refers to TIBCO iProcess Workspace Lite functionality exposed in a portal. In certain cases this will equate to a Java Portlet, in others to a different technology.

To embed TIBCO iProcess Workspace Lite in a portal:

1. Decide what, if any, navigation menu should be provided. In the simplest case of no menu, you can use the embedded skin shipped with TIBCO iProcess Workspace Lite. In other cases, you may wish to define your own menu using the description in [Customizing Skins on page 7](#).
2. Decide which TIBCO iProcess Workspace Lite Actions (e.g., `openItem`, `listWorkItems`, etc.) should be exposed by a gadget (or gadgets).
3. Decide how the necessary configuration will be passed to the gadget.

### Approaches

There are two principal approaches you can employ:

- Externalize the assembly of the configuration data into a TIBCO iProcess Workspace Lite URL, and then pass it to a single TIBCO iProcess Workspace Lite-embedding gadget. In this case the gadget need not be in any way specific to TIBCO iProcess Workspace Lite since its only task is to invoke the URL.
- Internalize the URL creation inside each gadget. In this case the gadget is specific to a single TIBCO iProcess Workspace Lite Action, but has the advantage of being much more readily invoked by external clients.

### Example URLs

You can view examples of URLs through the links in TIBCO iProcess Workspace Lite, as well as viewing Atom feeds.

Example 1:

The following is an example that invokes the **listWorkItems** Action, returns 25 work items (returnCount), starting at the fifth item in the list (startIndex), and displays using the carbonated skin:

<http://corpserver:7979/ipe-web/Workspace?action=listWorkItems&returnCount=25&startIndex=5&skin=carbonated&workQueueTag=liberty|broker|R&username=susieq&password=mypassword>

Example 2:

This example starts a case of the CARPOOL procedure with a description of “Chevy Volt”:

<http://corpserver:7979/ipe-web/Workspace?action=startCase&skin=accessible&procTag=v11|CARPOOL|1|0&description=Chevy%20Volt>

Available Actions

The following table lists the parameters common to all TIBCO iProcess Workspace Lite URLs:

Table 10: Parameters in all TIBCO iProcess Workspace Lite URLs

Parameter	Comments
action	One of the actions listed in the next table, or a custom action.
locale	Optional. Allows you to force the locale. The locale specified in the browser preferences is used by default.
skin	Optional. The “accessible” skin is used by default. Also see <a href="#">Configuring the Default Skin on page 11</a> .
username	All actions must either supply a username, or already hold an authenticated user session.
password	All actions must either supply a password, or already hold an authenticated user session.

The following table lists the available Actions, as well as the parameters for each Action:

Table 11: Available Actions and Parameters

Action	Parameters
audit	procTag caseNum
listCases	procTag startIndex (optional) returnCount (optional)
closeCase	procTag caseTag
forwardWorkItem	workQueueTag workItemTag destWorkQueueTag
forwardWorkItemForm	workQueueTag workItemTag
keepItem	workQueueTag workItemTag fieldNames
listProcesses	None
releaseItem	workQueueTag workItemTag fieldNames
startCase	procTag description
unlockItem	workQueueTag workItemTag
getUserAttributes	None
openItem	workQueueTag workItemTag fieldNames
listWorkItems listWorkItemsAtomFeed	workQueueTag startIndex (optional) returnCount (optional)

## Customizing the Number of Cases Returned from the Server

---

By default, a maximum of 500 cases are returned from the server when a case list is displayed. This allows the user to page through the cases, 20<sup>1</sup> at a time, up to a maximum of 500 cases.

If desired, you can increase the maximum number of cases returned from the server. Note, however, that if you increase it to a very large number, it can have an adverse effect on performance of the system.

To change the maximum number of cases to return from the server, follow these steps:

1. Ensure that your application server is stopped.
2. Extract the files from the JAR file containing the Action Processor queries to a temporary directory. This JAR file is located as follows:

`InstallDir\WEB-INF\lib\com.tibco.ipe.workspace.lib-version.jar`

where:

- *InstallDir* is the installation directory for iProcess Workspace Lite. For example, if you are using Tomcat for the application server, the installation directory might be something like: C:\Tomcat\webapps\ipe-web.
- *version* is the version number of your release of iProcess Workspace Lite.

3. Make a copy of the `CaseList.query.xml` (the query files are located in the \query directory).
4. Rename the copy, for example, `CustomCaseList.query.xml`.
5. Copy your custom query file to the following directory (you will need to create the \query directory if this is your first custom query):

`InstallDir\WEB-INF\classes\query`

The query file can actually be placed anywhere within the class path. This example shows where to copy the query file when using Tomcat; its class loader looks for resource files in `\WEB-INF\classes`. If using a different application server, change the path accordingly.

6. Open the query file you have renamed.

---

1. This is also a customizable value. To change the number of items to display by default per page in lists, change the value of the **defaultReturnCount** parameter in the `InstallDir\WEB-INF\web.xml` file. Note that this controls the number of items per page in *all* lists.

7. Locate the `<sso:MaxCnt />` element.
8. Change the default value of 500 to the desired number of cases you would like returned from the server.

For example:

---

```
<sso:MaxCnt>1000</sso:MaxCnt>
```

---

This specifies to return 1000 cases from the server when the case list is displayed.

9. Save and close your query file. In this example, you are saving the `CustomCaseList.query.xml` file.
10. Open the following file:

`InstallDir\WEB-INF\web.xml`

11. Locate the **listCasesQuery** parameter name:

---

```
<init-param>
  <param-name>listCasesQuery</param-name>
  <param-value>/query/CaseList.query.xml</param-value>
</init-param>
```

---

12. Change:

`/query/CaseList.query.xml`

to:

`/query/CustomCaseList.query.xml`

This causes your custom query to be used instead of the one provided with the application.

Note - It's important that the path begin with '/', and that the path be relative to classes.

13. Save and close the `web.xml` file.
14. Restart your application server.
15. Run the iProcess Workspace Lite application.