TIBCO iProcess™ Client (JSP)

Customization Guide

Software Release 9.2
September 2009

TIBCO
The Power of Now®

Important Information

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. Please see the readme.txt 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.

Contents

Back to Library

Back to Library

About This Guide

This guide describes how to configure and customize the **TIBCO iProcess Client (JSP)**.

The TIBCO iProcess Client (JSP) is referred to throughout the rest of this document as the iProcess Client (JSP).

Back to Library

How to Use This Guide

You should read Chapter 1 first. This chapter describes the iProcess Client (JSP) architecture and provides an overview of how it works. You can then consult the following chapters/appendices as required:

- Chapter 2 describes how to configure the iProcess Client (JSP).
- Chapter 3 describes how to customize the iProcess Client (JSP).
- Appendix A shows how iProcess step elements are mapped to HTML elements and attributes when a step is converted to a JSP page.

Back to Library

Target Audience

This guide is aimed at anybody who wants to configure or customize the iProcess Client (JSP), such as web/application server administrators, web developers or integrators.

If you want to configure the iProcess Client (JSP), you should be familiar with the following subjects:

- JavaServer Pages (JSP) technology
- administration of your particular web/application server
- iProcess

If you want to customize the iProcess Client (JSP), or integrate it with other applications or technologies, you should be familiar with:

- Java
- JavaScript
- JavaServer Pages (JSP) technology
- TIBCO iProcess™ Objects

Back to Library

Changes from the Previous Issue of This Guide

Major changes from the information presented in the previous issue of this guide are:

Product Re-branding

Further product re-branding has been carried out in this release, as follows:

- TIBCO Staffware Process Suite™ has been renamed TIBCO iProcess Suite™
- TIBCO iProcess™ Client (Windows) has been renamed TIBCO iProcess™ Workspace (Windows)
- TIBCO iProcess™ Client (Browser) has been renamed TIBCO iProcess™ Workspace (Browser)
- The term "Staffware" has been replaced by "iProcess".

Until this work is complete you may still see references to Staffware and to the old product names within the software and in some documentation.

Back to Library

Where You Can Find More Information

You can find more information about the iProcess Client (JSP) software from the following sources:

- The *TIBCO iProcess Client (JSP) Installation* guide explains how to install the software.
- A **Readme** file, supplied with the software, provides any last-minute and version-specific information that could not be included in the main documentation. Please see the *TIBCO iProcess Client (JSP) Installation* guide for more information about this file.
- The *TIBCO iProcess Client (JSP): User's Guide* explains how to use the iProcess Client (JSP).
- The *TIBCO iProcess™ Step Definition Converter: User's Guide* explains how to convert iProcess step definitions into JSP for use with the iProcess Client (JSP).
- Online help is available within the iProcess Client (JSP).
- Detailed information about using the TIBCO iProcess Suite can be found on the TIBCO iProcess Suite: Documentation Library CD.
- There is also a useful resource, **http://power.tibco.com**, that delivers technical content to the TIBCO user community. This site has been developed to foster an open forum where users of TIBCO products can find valuable information, example projects and resources for those projects, and exchange ideas with other users. Entry to this site requires a username and password. If you do not have a username, you can request one.
- For the latest TIBCO iProcess Suite product information, please refer to the TIBCO Support web site at **http://www.tibco.com/services/support**.

Back to Library

Document Conventions

To avoid needless repetition of otherwise identical pathnames, separators in pathnames are generally shown in this guide using the UNIX convention of a forward-slash (/).

The equivalent pathname on a Windows system is the same, but using the backslash (\) as a separator character.

Windows pathnames are occasionally shown explicitly, using backslashes as separators, where a Windows-specific path, example or syntax is required.

Throughout this guide iProcess server indicates either the TIBCO Process™ Engine or TIBCO iProcess™ Engine depending on the version of iProcess you are using.

Back to Library

Installation Directories

The following variables are used throughout this guide to refer to directories where iProcess Client (JSP) components are installed.

SWWebClientInstallDir The directory where you installed the iProcess Client (JSP) files, either directly and/or as a Web ARchive (WAR) file.

Note - See the TIBCO iProcess Client (JSP) Installation guide for more information.

SWWebClientDir The directory where the iProcess Client (JSP) JSP (***.jsp**) files are installed.

The iProcess Client (JSP) Help (***.html**) and image (***.gif**, ***.jpg**) files are also installed in sub-directories below this directory.

If you chose to install the product:

- directly, this directory will be *SWWebClientInstallDir***/jsp**.
- as a WAR file, this directory will be wherever the web/application server stores web applications. Refer to the **Readme** file and/or your web/application server documentation to determine the location of this directory.

SWWebClientBeansDir The directory where the iProcess Client (JSP) beans (***.class** files) are installed.

If you chose to install the product:

- directly, this directory will be *SWWebClientInstallDir***/beans**.
- as a WAR file, this directory will be wherever the web/application server stores JavaBeans for a web application. Refer to the **Readme** file and/or your web/application server documentation to determine the location of this directory.

SWWebClientConfigDir The directory where the iProcess Client (JSP) configuration files (**webclient.properties**, **menu.xml** and **messages.properties**) are installed.

SWStepConvertDir The directory where the TIBCO iProcess Step Definition Converter is installed (on a computer running Windows).

Back to Library

Back to Library

Chapter 1 Overview

This chapter gives you a brief overview of the iProcess Client (JSP) - its architecture and how it works.

Topics

Back to Library

What is the iProcess Client (JSP)?

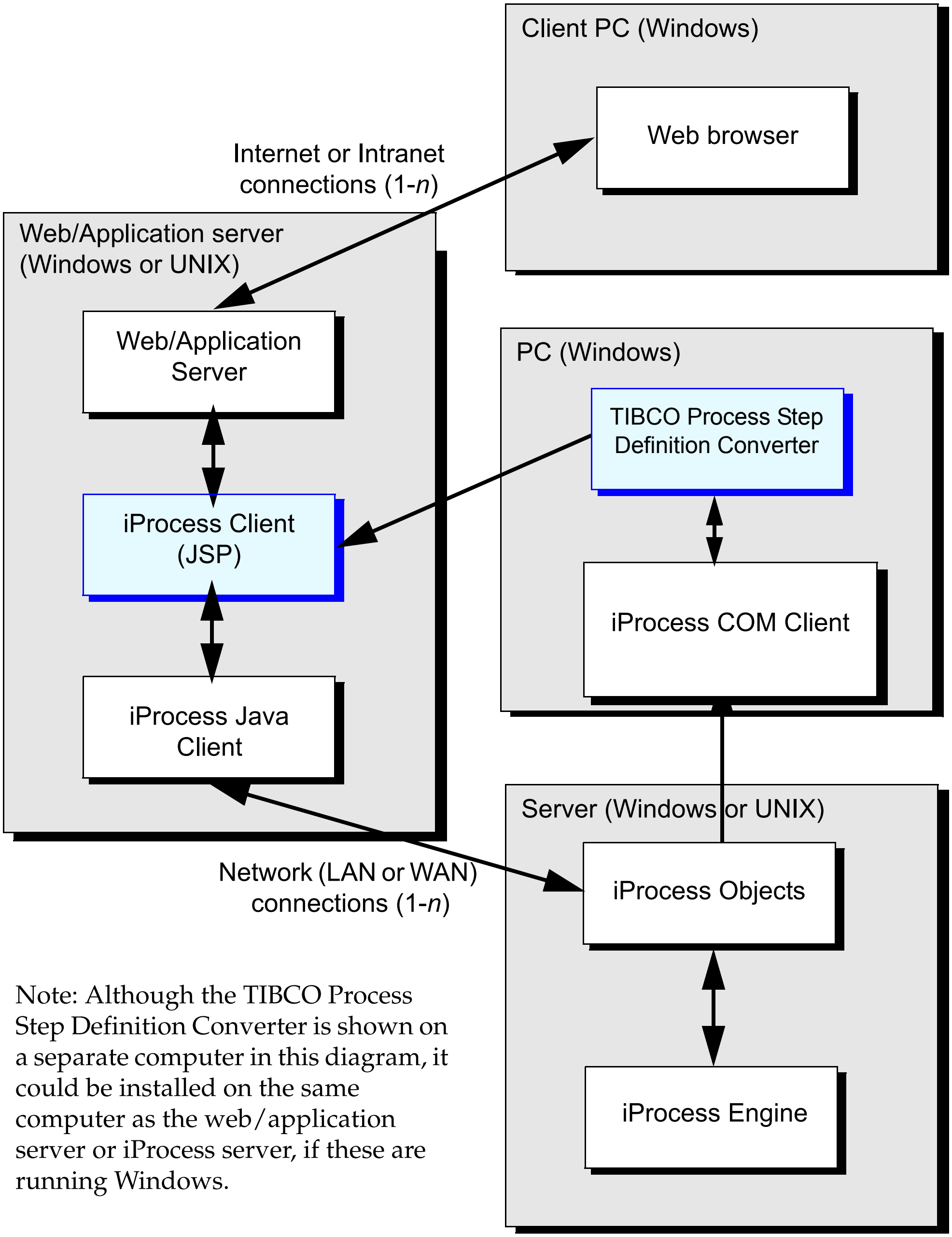
The iProcess Client (JSP) is a JSP-based web/application server application that allows organizations to extend their iProcess systems over the internet and intranets, enabling staff to participate in a procedure irrespective of where they are or what time it is.

The diagram on the following page shows the architecture of the iProcess Client (JSP), and how it is used to provide browser-based access to iProcess.

Note that the iProcess Client (JSP) uses:

- TIBCO iProcess Objects to communicate with iProcess (by calling TIBCO® iProcess Objects Java Client objects from code in JSP pages). For more information about TIBCO iProcess Objects please refer to the **Help** supplied with the TIBCO iProcess Objects Java Client.
- the TIBCO Process Step Definition Converter to generate the JSP representations of iProcess steps. For more information about the Step Definition Converter, please refer to the *TIBCO iProcess Step Definition Converter: User's Guide*.

Back to Library



Note: Although the TIBCO Process Step Definition Converter is shown on a separate computer in this diagram, it could be installed on the same computer as the web/application server or iProcess server, if these are running Windows.

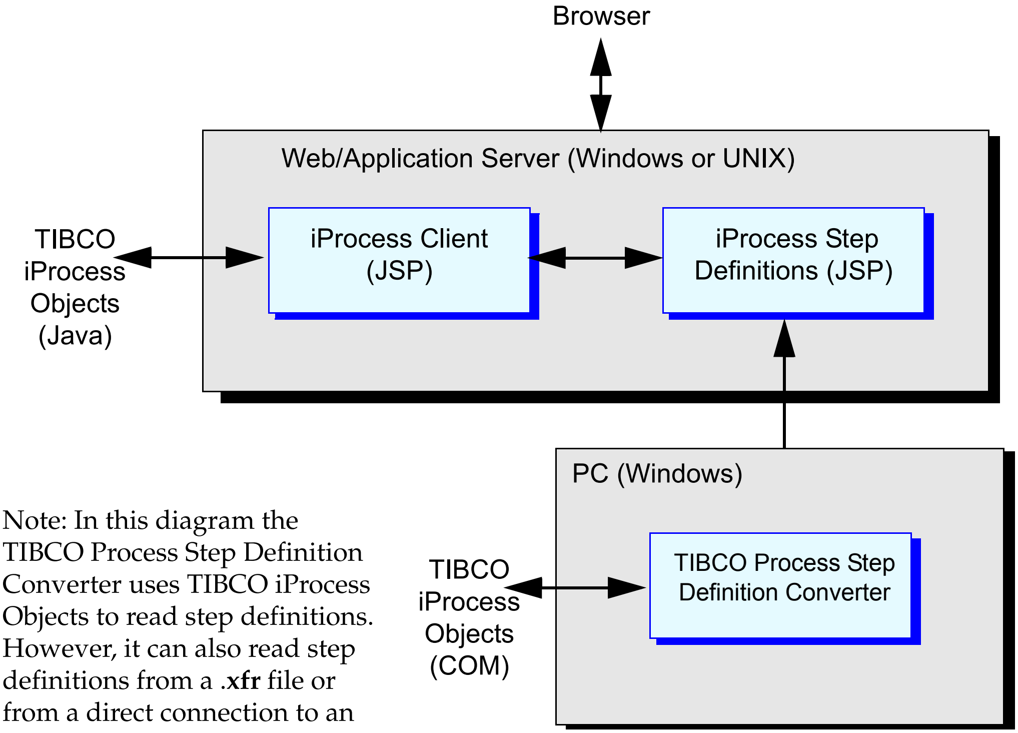
Back to Library

How the iProcess Client (JSP) Works

The iProcess Client (JSP) has two main components, as shown in the following diagram:

- a set of interrelated JSP pages containing HTML, Java and JavaScript, which provide the necessary logic and user interface.
- A step definition cache, containing JSP pages representing iProcess step definitions, which can be sent to the browser. The JSP pages must be generated using the Step Definition Converter and then copied into the step definition cache.

When the iProcess Client (JSP) receives a request to display a step definition from the browser, it looks for a step definition file of that name in the step definition cache.


Browser
Web/Application Server (Windows or UNIX)
TIBCO iProcess Objects (Java)
iProcess Client (JSP)
iProcess Step Definitions (JSP)
PC (Windows)
TIBCO iProcess Objects (COM)
TIBCO Process Step Definition Converter

Note: In this diagram the TIBCO Process Step Definition Converter uses TIBCO iProcess Objects to read step definitions. However, it can also read step definitions from a **.xfr** file or from a direct connection to an iProcess server.

Back to Library

What is the TIBCO Process Step Definition Converter?

The TIBCO Process Step Definition Converter is a separate product that runs on a Windows computer. Refer to the TIBCO iProcess Step Definition Converter: User's Guide for detailed information about how to use it.

The Step Definition Converter reads iProcess step definitions and converts them into JSP. Step definitions can be read either from TIBCO iProcess Objects, from a **.xfr** file, or from a direct connection to an iProcess server.

The Step Definition Converter uses a step definition template (*SWStepConvertDir***TemplateJSP.FC**) as a wrapper, and creates the following JSP pages for each step definition to be created, in the step definition cache specified by the user:

- a **step definition file**, *STEPNAME***.jsp**. When it is processed by the web/application server, the step definition file will show the step definition and its current field values in the browser.

  The following information is written into the file's header:

  — the version number of the iProcess procedure from which the step definition was generated.

  — the time-stamp of the step definition template from which the step definition was generated.

- a **persistence file**, *STEPNAME***Close.jsp**. The persistence file is called when the user clicks the **Keep** or **Release** button on the step definition file. It instructs TIBCO iProcess Objects to keep or release the work item, and to update any changed field values.

For more information about the step definition template, see Customizing the Step Definition Template on page 33.

Updating the Step Definition Cache

Because the JSP forms used by the iProcess Client (JSP) are generated externally, you must update them manually whenever the original iProcess step definitions are changed. To do this regenerate the required JSP forms, using the Step Definition Converter, overwriting the existing versions.

Back to Library

Security

You should note that the iProcess Client (JSP) does not encrypt the data it transmits.

If you intend to use the iProcess Client (JSP) to transmit sensitive data - either passwords or step data, you are advised to use secure protocols, such as Secure Socket Layer (SSL), to establish a secure connection between the browser and web/application server.

For more information about the use of secured connections, please consult your web/application server documentation.

Back to Library

Chapter 2 Configuring the TIBCO iProcess Client (JSP)

This chapter describes the ways in which you can configure the TIBCO iProcess Client (JSP) to suit your requirements.

Topics

Back to Library

Using the URL Login

In addition to logging in using the standard login dialog, you can log in to the iProcess Client (JSP) using a URL.

There are two methods you can use:

- The POST method where the log in information is passed inside a Form tag in the body of a JSP page.
- The GET method where the log in information is passed in the URL.

The HTML specifications define the difference between GET and POST. GET means that form data is encoded (by a browser) into a URL while the POST method means that the form data is within a message body. This means that, using POST hides the login information in the body of the page (so passwords are not visible on the URL).

Back to Library

Using the GET Method to Log in to the iProcess Client (JSP)

Using the GET method, the username, password and log in parameters are passed to the web server in the URL. This is useful if you do not want to allocate iProcess Usernames and Passwords to users. Users can go to the URL and go straight to the work queue you want them to work on.

To do this enter the following URL:

http://*web server*/?*username=username&password=password&nodetag=staffware server|nodename|spo server IP address|port number|spo director|instance*

where:

- *web server* is the name of the computer where your webserver\application is installed.
- *username* is your iProcess username.
- *password* is your iProcess password if the iProcess server is set up to require a password.

The password is visible in plain text as part of the URL.

- *nodetag* is the name of the machine where your iProcess server is installed.
- *nodename* is the name of your iProcess server.

- *SPO server IP Address* is the IP address of the machine where the TIBCO iProcess Objects server is installed.
- *port number* is the TCP port number where your TIBCO iProcess Objects server is listening to requests. The port number must be static. See "Configuring the TCP Port Number" in the *TIBCO iProcess Objects: Programmer's Guide* for information on how to do this.
- *spo_director* specifies whether or not the TIBCO iProcess Objects server is an TIBCO iProcess™ Objects Director. Enter:
  - **Y** if your TIBCO iProcess Objects server is a TIBCO iProcess Objects director, or
  - **N** if you TIBCO iProcess Objects server is not an TIBCO iProcess Objects director.
- *instance* specifies the TIBCO iProcess Objects instance number.

Below is an example URL:

http://uk_joannablap/?username=joannab&password=mypassword&nodetag=uk_joannablap|sw9010|127.0.0.1|55555|n|1

where the parameters are:

Parameter	Value
Web Server	uk_joannablap
iProcess Username	joannab
iProcess Password	mypassword
Nodetag	uk_joannablap
Nodename	sw9010
iProcess Objects IP Address	127.0.0.1
Port Number	55555
TIBCO iProcess Objects Director	n
Instance	1

Back to Library

Using the POST Method to Log in to the iProcess Client (JSP)

Using the POST method, the username, password and nodetag information is passed to the webserver via a separate JSP page. This is useful if you do not want to allocate iProcess Usernames and Passwords to users and you do not want the username and password information to be visible in a URL.

Below is an example of the log in information that would need to be passed to the iProcess Client (JSP) using the POST method:

```
<html>
<head>
<title>iProcess Client Redirect</title>
<SCRIPT lang=javascript>
function redirectImmediate()
{
document.rdirect.submit();
}
</SCRIPT>

</head>

<body>

<FORM name="rdirect" method="POST" action="default.jsp">
<INPUT type="hidden" size="80"name="username" value="swadmin">
<INPUT type="hidden" size="80"name="password" value="mypassword">
<INPUT type="hidden" size="80" name="nodetag"
value="swmachine|swnode|10.12.84.2|4567|N|1"> <!--INPUT
type="SUBMIT"-->
<input type="submit" value="OK" onclick="redirectImmediate();">
</FORM>

</body>
</html>
```

Back to Library

where the parameters are:

<table>
<tr><th>Parameter</th><th>Value</th></tr>
<tr><td>iProcess Username</td><td>swadmin</td></tr>
<tr><td>iProcess Password</td><td>mypassword</td></tr>
<tr><td>Nodetag</td><td>swmachine|swnode|10.12.84.2|4567|N|1
where:
• <b>swmachine</b> is the name of the computer where your web server is installed.
• <b>swnode</b> is the name of your iProcess server.
• <b>10.12.84.2</b> is the IP address of the machine where the TIBCO iProcess Objects server is installed.
• <b>4567</b> is the TCP port number where your TIBCO iProcess Objects server is listening to requests. The port number must be static. See “Configuring the TCP Port Number” in the <i>TIBCO iProcess Objects: Programmer’s Guide</i> for information on how to do this.
• <b>N</b> or <b>Y</b> specifies whether or not the TIBCO iProcess Objects server is an TIBCO iProcess Objects director.
• <b>1</b> is the TIBCO iProcess Objects instance number.</td></tr>
</table>

Back to Library

Setting the Language

If you want to configure the iProcess Client (JSP) to support one or more languages for the user interface, you must create a **messages.properties** file for each language you want the iProcess Client (JSP) to support. To do this:

1. Configure the language settings in Microsoft Internet Explorer. To do this:

    a. From Microsoft Internet Explorer, click **Tools** > **Internet Options**. The **Internet Options** dialog is displayed.

    b. From the **General** tab, click **Languages**. The **Language Preference** dialog is displayed.

    c. Click **Add**. The **Add Language** dialog is displayed.

    d. Select the language you want to add.

    > Make a note of the language code displayed in square parentheses beside the language name in the **Language Preference** dialog, as the **messages.properties** file must include the language code in the filename.

    e. Click **OK**.

2. In the *SWWebClientConfigDir*, copy and paste the **messages.properties** file. The copy of the **messages.properties** file must remain in the *SWWebClientConfigDir*.

3. Rename the **messages.properties** file using the language code that is displayed in square parentheses in the **Language Preference** dialog in Microsoft Internet Explorer. The last 2 digits of the language code must be in upper case. For example, **messages_es_US.properties** for American Spanish or **messages_DE.properties** for German.

4. Open the renamed file, for example **messages_DE.properties**, in a text editor.

5. Translate all the text strings to the new language. For example, in the **messages_DE.properties** file, the text strings should be translated to German.

6. Save and close the file.

7. Repeat steps 1-6 for every language you want the iProcess Client (JSP) to support.

Back to Library

Viewing sw_warn and sw_error messages

If you want to configure the iProcess Client (JSP) to view sw_warn and sw_error messages, you need to convert the [SYSTEM] procedure to JSP in the TIBCO Process Step Definition Converter. See the *TIBCO iProcess Step Definition Converter: User's Guide* for more information on converting iProcess procedures to JSP.

Once the [SYSTEM] procedure has been converted, any sw_warn and sw_error messages that are delivered as work items to the System Administrator's work queue can be opened and viewed. If you do not convert the [SYSTEM] procedure to JSP using the TIBCO Process Step Definition Converter then, although sw_warn and sw_error messages are still delivered as work items to the System Administrator's work queue, if you try to open them, a **404 File not found** message is displayed.

Back to Library

Modifying the TIBCO iProcess Client (JSP) Configuration File

The TIBCO iProcess Client (JSP) configuration file, *SWWebClientConfigDir*/**webclient.properties**, allows you to configure a number of different areas of the TIBCO iProcess Client (JSP).

The following sections describe the different entries that you can configure.

⚠ Do not change anything in the configuration file that is not documented in the following sections.

Setting the Number of Items Returned Per Block from the TIBCO iProcess Objects Server

The **blocksize** parameter determines the number of items returned per block from the TIBCO iProcess Objects server. The default is **20**. TIBCO recommend that you do not change the default setting.

Setting Display Options for the Work Queue Window

The **wqpage** entry specifies the number of work items that are displayed in the work queue window. (This makes display of large queues more efficient. The user can page through the work items if the queue has to span multiple pages.) A value of **18** (the default value) specifies that 18 work items are shown on a single page.

Setting the Number of Maximum Audit Trails Returned Per Block from the TIBCO iProcess Objects Server

The **maxauditlist** parameter configures the maximum number of audit trails returned from the TIBCO iProcess Objects Server. It is set to **100** by default although you can amend this, depending on your requirements.

When you select a procedure from the **Audit Trail** folder, the Audit Case List Filter Criteria window is displayed. If the **maxauditlist** parameter is set to 100, Audit List Filter Criteria that returns 100 cases or less will display the audit trail list. If the Audit List Filter Criteria will return an audit trail list greater than 100, the **Number of items exceeds limit. Please refine you case list filter criteria** message is displayed. The user should click **Access Criteria: Filter** to display the Audit Case List Filter Criteria window and re-define the audit case list criteria to reduce the number of audit trails that will be returned.

Back to Library

See the *TIBCO iProcess Client (JSP): User's Guide* for more information on setting Audit Case List Filter Criteria.

The lowest permissible value for the **maxauditlist** parameter is **1**.

Setting Display Options for the Audit Trail Window

The **auditpage** entry specifies the number of cases of a procedure that will be downloaded to and shown on a single page in the Audit Trail window. (This makes the display of large numbers of cases more efficient. The user can page through the list of cases that span multiple pages.) A value of **20** (the default value) specifies that 20 cases are shown on a single page.

Back to Library

Setting the Work Item Refresh Interval

This section determines how often your currently selected work queue is automatically refreshed. The default is set to 60 seconds.

```
RefreshInterval = 60
```

You can turn this feature off by setting this value to 0. If you do set this value to 0, it means that your work queue is only refreshed if you refresh it manually by clicking **Refresh Queue**.

Setting Forward Options for Work Items

The **forward** entry determines if it is possible to forward a work item. This will be either true or false. The default is true.

Setting Debug to Identify Problems

The **dbgstring** entry enables you to set debug on memory and object usage to help you identify any problems. If you set debug using the **dbgstring** entry, the resulting debug is logged in a log file. See *Defining the Log File* on page 16.

Defining the Debug Level

You can set the level of debug by specifying some or all of the debug levels in the **dbgstring** entry. The default is set to **WARNING|AUTH|CONNECT**.

To set debug, modify the **dbgstring** entry in the configuration file, using the following syntax:

dbgstring=*string*

where *string* is all, or a combination of, the following debug levels. (Each debug level should be separated by "|".):

DEBUG	Low level debugging.
MEMORY	Memory usage.
WARNING	Non-fatal errors. (The user's session is not terminated.)
CONNECT	An entry is written for every page that is visited.
AUTH	The logged in user that performed the operation.
TIMING	The length of time the JSP Web Pages take to execute.

DEBUG and MEMORY are for iProcess use only. You should only set these debug levels on the advice of TIBCO Support.

For example, the following entry sets the **dbgstring** parameter to capture non-fatal errors and which pages have been visited:

dbgstring=WARNING|CONNECT

Defining the Log File

The TIBCO iProcess Client (JSP) logs connection attempts, warnings, errors and debug to a log file. The **logfile** entry defines the name of this log file, either as a full pathname or as a simple filename. The default entry is **swclient.log**.

If **logfile** is specified as a simple filename, the location of the file will be determined by the web/application server.

Log File Format

Log file entries are in the format:

date|thread id|instance id|page|user|debug event id|message

where:

- *date* gives the date and time that the entry was created.
- *thread id* identifies the thread that generated the entry.

Back to Library

- *instance id* identifies the instance of the **beanSWSEO.class**.
- *page* is the filename of the page (without the filetype suffix) that a connection has been established to.
- *user* is the iProcess User name of the user who performed the operation.
- *debug event id* is the type of debug event depending on the dbstrings specified in the **dbgstring** entry. The following table describes the debug event id that corresponds to each debug string:

Debug Event ID	dbstring entry
DEBUG	DEBUG
MEMORY	MEMORY
WARN	WARNING
CONN	CONNECT
AUTH	AUTH
TIMING	TIMING

Back to Library

- *message* is a description of the event that occurred.

The following is an example of some log file entries:

```
12/05/2003 09:23:40|T2|I0|/dologinout.jsp|UNKNOWN|CONN|Page
/dologinout.jsp started

12/05/2003 09:23:40|T2|I0|/dologinout.jsp|UNKNOWN|DEBUG|URL for
page null

12/05/2003 09:23:40|T2|I0|/dologinout.jsp|UNKNOWN|AUTH|Login
attempt User: swadmin to node: 9010@uk_joannablap from 127.0.0.1

12/05/2003 09:23:40|T2|I0|/dologinout.jsp|UNKNOWN|MEMORY|CREATED
+SWENTERPRISE+ ACTIVE: 3
```

In the event of an error, the TIBCO iProcess Client (JSP) also returns information to the client to notify the user that there was a problem and that the error has been logged.

Defining if Locked Work Items can be Forced Open

Work items are marked as locked (by the TIBCO iProcess Objects Java Client) when they are opened. If a user's browser session or PC crashes while they have a work item open, the lock is not reset (even if the iProcess server is restarted), and the item continues to appear as locked when the queue is next accessed.

The **breaklocks** entry defines whether or not work items locked in this way can be forced open:

- If **breaklocks = false**, a locked work item cannot be unlocked (and opened) from the TIBCO iProcess Client (JSP) by any user.
- If **breaklocks = true**, a locked work item can be unlocked (and opened) by the user who locked the work item. The user will be prompted to confirm that they want to break the lock and open the item.

A user cannot unlock an item if the lock originated from the TIBCO Process™ Workspace (Windows).

Locks can also be reset using TIBCO iProcess Objects or the iProcess **swutil UNLOCKMAIL** utility. Refer to the TIBCO iProcess Objects **Help** or *TIBCO iProcess Engine: Administrator's Guide* for more information about these methods.

Back to Library

Adding iProcess and Process Engines to the Static Login List

One or more **staticserver** entries can be used to define TIBCO iProcess Engines that should always be displayed in the list of available servers on the login screen, irrespective of whether they can be contacted by the TIBCO iProcess ObjectsJava Client.

The drop down list of available servers, which is displayed on the login screen, is generated dynamically. The TIBCO iProcess Objects Java Client sends a User Datagram Protocol (UDP) broadcast message to detect TIBCO iProcess Objects servers. Each server that responds is added to the list.

Some servers cannot be contacted by this method - for example, those that are reached via routers that do not allow UDP broadcasts. To avoid reconfiguring the network or router, you can simply add these servers to the static server list to make them available to users.

For more information about the way the TIBCO iProcess Objects Java Client detects iProcess Objects servers on the network, see the **Help** supplied with the TIBCO iProcess Objects Java Client.

To add an iProcess server to the static list, add a **staticserver** entry to the file, using the following syntax:

```
staticserverX=serverName,port,ip_address,spo_director,instance
```

where:

- *X* is any number (0-*n*), uniquely identifying this entry.
- *serverName* is the name of the iProcess server, in the format *nodename@computername.*
- *port* specifies the port number of the computer hosting the iProcess server. The port number must be static.
- *ip_address* specifies the IP address of the computer hosting the iProcess server. The IP address must be static.
- *spo_director* specifies whether or not the TIBCO iProcess Objects server is an TIBCO iProcess Objects director. Enter:
  - **Y** if your TIBCO iProcess Objects server is an TIBCO iProcess Objects director, or
  - **N** if your TIBCO iProcess Objects server is not an TIBCO iProcess Objects director.
- *instance* specifies the TIBCO iProcess Objects instance number.

There must be no spaces in the **staticserver** entry.

For example, to add an iProcess server called **SWEnt1** on computer **PURCH1** to the static login list:

```
staticserver0=SWEnt1@PURCH1,2556,10.10.2.21,N,1
```

Enabling Automatic Login to iProcess Servers

One or more **autologin** entries can be used to define iProcess servers that will be automatically logged into when the TIBCO iProcess Client (JSP) is accessed.

To enable automatic login to a server, add an **autologin** entry to the file, using the following syntax:

```
autologinX=serverName,machineName,ip_address,port,spo_director,instance,name[,passw
ord]
```

where:

- *X* is any number (0-*n*), uniquely identifying this entry.
- *serverName* is the name of the iProcess server.

- *machineName* is the name of the machine where your iProcess server is installed.
- *ip_address* specifies the IP address of the computer hosting the iProcess server. The IP address must be static.
- *port* specifies the port number of the computer hosting the iProcess server. The port number must be static.
- *spo_director* specifies whether or not the TIBCO iProcess Objects server is an TIBCO iProcess Objects director. Enter:
  - **Y** if your TIBCO iProcess Objects server is an TIBCO iProcess Objects director, or
  - **N** if your TIBCO iProcess Objects server is not an TIBCO iProcess Objects director.
- *instance* specifies the TIBCO iProcess Objects instance number.
- *name* is the iProcess user name to be used to log in.
- *password* is the iProcess password for this login. This parameter can be omitted if login password checking is disabled on the iProcess server.

There must be no spaces in the **autologin** entry.

For example, the following entry enables automatic login to the iProcess server **i9server** on computer **sales1**. The username used to log in is **swadmin**, with no password supplied. (Password checking must therefore be disabled on the iProcess server for the login to succeed.)

```
autologin0=sales1|i9server|10.12.41.150|20007|N|1,swadmin,swadmin
```

Login Dialog Options

The following entries can be used to control the login options available to the user.

Entry	Description
require_password	Defines whether or not the Password field is displayed in the login dialog. This should only be set to **false** if login password checking is disabled on *all* iProcess servers which can be accessed from this client.

Back to Library

Entry	Description
logon_user	Defines whether a user can log in with a username other than their current username. If set to **true**, this option: • sets the contents of the Username field in the login dialog to be the current username. • makes the Username field read-only.

Setting Forced Garbage Collection

The **gcinterval** parameter determines the number of milliseconds between Web Client forced Garbage Collection. The default is **120000**. TIBCO recommend that you do not change the default setting.

Back to Library

Defining Custom Menu Items

You can define one or more new menu options to be appended to the bottom of the list of folders on the left hand side of the iProcess Client (JSP) window. This is useful, if for example, you wanted to add a new menu section called **Links** that gave a list of links to other websites that are frequently used.

You must create a file(s) that contains the code for the new menu item. See *Customizing TIBCO iProcess Client (JSP) JSP Pages* on page 35 for more information.

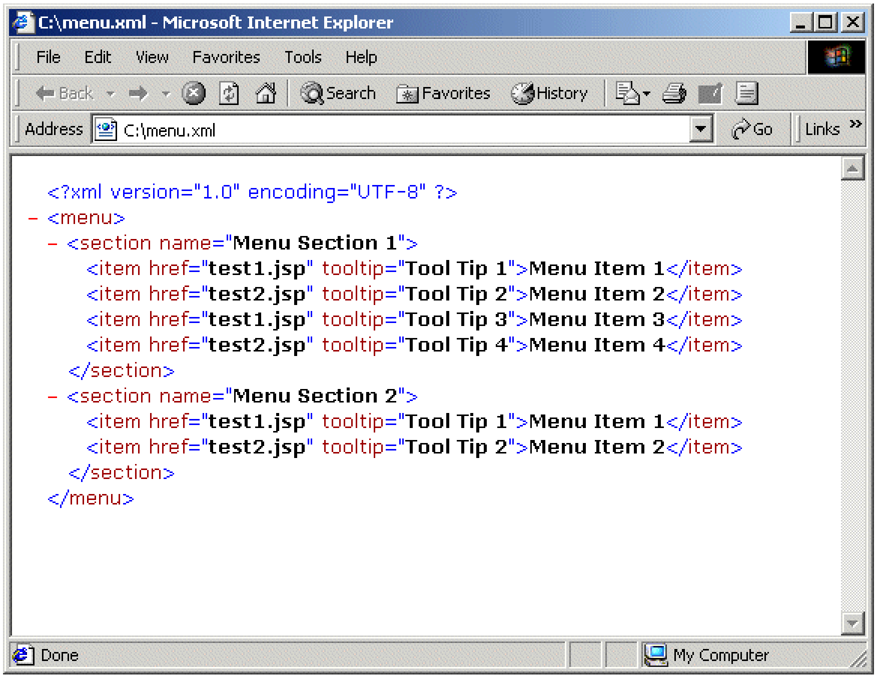
Once this file(s) has been created, you must define the new menu option(s) in the *SWWebClientConfigDir*/**menu.xml** file. The **menu.xml** file specifies the filename that contains the code for the new menu item, the menu name and the tool tip of the new menu item. You can rename **menu.xml** to a new filename.

If you want the iProcess Client (JSP) user interface to support multiple languages, you must create a **menu.xml** file for each language you want the iProcess Client (JSP) to support, in the same way that you must create a message properties file for each language you want the iProcess Client (JSP) to support. See *Creating Multiple menu.xml Files to Support Multiple Languages* on page 22.

Once the **menu.xml** file is amended, you need to uncomment the **menusection** parameter in the configuration file. You can rename the **menu.xml** file to another name, for example, **links.xml** as long as you specify the correct file name in the **menusection** parameter.

Format of menu.xml File

The entries in the **menu.xml** file are in the following format:


C:\menu.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address C:\menu.xml
Go Links
Done
My Computer

```
  <?xml version="1.0" encoding="UTF-8" ?>
- <menu>
  - <section name="Menu Section 1">
      <item href="test1.jsp" tooltip="Tool Tip 1">Menu Item 1</item>
      <item href="test2.jsp" tooltip="Tool Tip 2">Menu Item 2</item>
      <item href="test1.jsp" tooltip="Tool Tip 3">Menu Item 3</item>
      <item href="test2.jsp" tooltip="Tool Tip 4">Menu Item 4</item>
    </section>
  - <section name="Menu Section 2">
      <item href="test1.jsp" tooltip="Tool Tip 1">Menu Item 1</item>
      <item href="test2.jsp" tooltip="Tool Tip 2">Menu Item 2</item>
    </section>
  </menu>
```

where:

- *item href* is the file that is opened when you click on the menu item. This file can be in any format, for example, **.jsp** or **.txt**.
- *tooltip* is the tool tip text you want displayed when the cursor is rested on the menu item.
- *menu item* is the menu name to be displayed.

Creating Multiple menu.xml Files to Support Multiple Languages

If the iProcess Client (JSP) supports one or more languages then a **menu.xml** file needs to be created for each language supported. To do this:

1. In the *SWWebClientConfigDir*, copy and paste the **menu.xml** file.
2. Rename the **menu.xml** file using the language code that is displayed in square parentheses in the **Language Preference** dialog in Microsoft Internet Explorer. The last two digits of the language code must be in upper case. For example, **menu_es_US.xml** for American Spanish or **menu_DE.xml** for German. See *Setting the Language* on page 12.

Back to Library

3. Open the renamed file, for example **menu_DE.xml** file in a text editor.
4. Translate the text strings in the file to the new language.
5. Save and close the file.
6. Repeat steps 1-5 for every language the iProcess Client (JSP) supports.

Adding the menusection Parameter

To add a custom menu, uncomment the **menusection** entry in the configuration file. For example,

```
menusection=menu.xml
```

Back to Library

Configuring Dynamic Lists

This section defines whether sub-lists (for example, work queues) are pre-loaded when a user logs in to the iProcess Client (JSP). You can specify:

- **true** to pre-load the lists when a user logs in to the iProcess Client (JSP).
- **false** to stop the lists pre-loading when a user logs in to the iProcess Client (JSP).

Pre-loading lists can affect performance, for example, when you have a large number of users logging in to the iProcess Client (JSP) or if you have one user that has a large number of work queues and/or audit trails. By setting the **preloadlists** variable to **false**, the lists are only loaded, for example, when a user clicks on the **Work Queue** option. The default is **false**. For example:

```
preloadlists = false
```

Sorting the Procedure List

The **procdisplayorder** entry determines how procedures are sorted in the iProcess Client (JSP). To change the sort order for procedures, amend the **procdisplayorder** entry in the configuration file.

The options are as follows:

Option	Description
A	Alphabetical
N	Procedure Number
R	Reverse Alphabetical

The default is **A**.

Changing the Default Folder Display

Back to Library

You can change the default folders that are displayed for individual users.

To change the default folder display, uncomment and amend or add the **menuoption** entries in the configuration file. You need a **menuoption** entry for each user whose default folder display you want to configure.

If no **menuoption** entry is configured, the default folder display is used. The following folders are displayed by default:

- Work queues
- Audit trail
- Case Start
- Tools
- Nodes

The **menuoption** entries are in the following format:

```
#menuoption0=swadmin|all,workqueue=true,audittrail=true,casestart=
true,tools=true,node=true
```

where:

- **menuoption*n*** is the menuoption for each user whose default folder display you want to configure. *n* is a numeric and must be incremented for each **menuoption** entry you add for each user.
- **swadmin** | **all** is either the iProcess Username of the user whose default folder display you want to configure (for example, **swadmin**) or **all** which is a global setting that enables you to set the same default folder display for all iProcess users.

- **workqueue=true,audittrail=true,casestart=true,tools=true,node=true** are entries for each of the folders that can be displayed with a value of either **true** or **false**, depending on your requirements. A value of **true** means the folder is displayed and a value of **false** means the folder is not displayed.

Note that:

- There must be no spaces in the **menuoption** entries.
- You can set a **menuoption** entry to use the global setting of **all** and add other **menuoption** entries for individual users. This is because the **menuoption** entries for the individual users will override the global setting of **all**. This is useful if, for example, you want to configure one user to have access to all folders while all other users have access to just one folder.
- If no **menuoption** entry is configured, the default folder display is used. This means if you have a temporary user who only needs access to one folder, you can configure a **menuoption** entry for that user only while all other users will automatically have the default folder display.

Back to Library

Examples

1. This example gives the iProcess Administrator user (**swadmin**) access to all folders and enables all other users to have access to the **Work Queues** folder only:

```
menuoption0=swadmin,workqueue=true,audittrail=true,casestart=true,
tools=true,node=true
```

```
menuoption1=all,workqueue=true,audittrail=false,casestart=false,to
ols=false,node=false
```

2. This example gives a user with the iProcess Username of **temp001** access to the work queues folder only. As there is no other **menuoption** entry, all other iProcess users will have the default folder display.

```
menuoption1=temp001,workqueue=true,audittrail=false,casestart=fals
e,tools=false,node=false
```

Changing the Default Work Queue Display

The following iProcess fields are displayed in the work queue by default:

- **SW_STEPDESC**
- **SW_CASEREF**
- **SW_CASEDESC**
- **SW_DEADLINE**
- **SW_QPARAM1**.

To change the default work queue display, uncomment and amend the **default display** entry in the configuration file. For example, to display the iProcess fields SW_CASEDESC, SW_LOCKER, SW_CASEREF, SW_DEADLINE, amend the **default display** entry as follows:

```
defaultdisplay=SW_CASEREF,SW_CASEDESC,
SW_DEADLINE,SW_LOCKER,
```

Back to Library

You can choose all or a combination of the following iProcess fields:

iProcess Field	Description
SW_ARRIVALTIME	Date and time when the work item arrived in the queue, in the format *dd/mm/yyyy hh:mm*.
SW_CASEDESC	Text describing the case (up to 24 characters).
SW_CASENUM	Case number of the case which the work item is part of.
SW_CASEREF	Unique case reference, in the form *procedure number-case number*, of the case which the work item is part of.
SW_DEADLINE	Date and/or time when the deadline on the work item expires (if one is set), in the format *dd/mm/yyyy hh:mm*.
SW_HOST	Node name of the system hosting the procedure which generated the work item.
SW_LOCKER	User name of the person who currently has the work item open (displayed only if the item is currently locked.
SW_PRIORITY	Priority value of the work item.
SW_PRODESC	Description of the procedure which generated the work item, supplied by the procedure definer.
SW_PRONAME	Name of the procedure which generated the work item, supplied by the procedure definer.
SW_QPARAM*n*	Work queue field.
SW_STARTER	User name of the person who started the case the work item belongs to.
SW_STATUS	Work item status. For example, opened or unopened.
SW_STEPDESC	Description of the step which is displayed when the work item is opened, supplied by the procedure definer.
SW_STEPNAME	Name of the step which is displayed when the work item is opened, supplied by the procedure definer.

Back to Library

If an iProcess User has set the Filter, Display or Sort Options then these settings will override the **default display** entry in the **webclient.properties** file.

Changing the Default View

When you first login to the iProcess Client (JSP), the Queue Summary is the first page that is displayed by default.

To change the default view, add a **firstpage** entry to the configuration file in the following format:

```
firstpage=filename
```

where *filename* is the filename of the page you want to display. For example:

```
firstpage=firstpage.htm
```

If there is no **firstpage** entry in the configuration file, the Queue Summary is displayed by default.

Improving Performance of the Work Queue Search

You can improve the performance of the **Work Queue Search** by specifying:

- the type of cases you search for. For example, only searching for open cases.
- the amount of work items that are returned by the search.

For more information about the **Work Queue Search**, see “Searching for Work Items” in the *TIBCO iProcess Client (JSP): User’s Guide*.

Specifying the Type of Cases to Search For

The **inactivecasesearch** variable defines the type of cases to search for when performing a work queue search. You can specify:

- **false** to only search for open cases, or
- **true** to search for both open and closed cases.

The default is **false**. For example:

```
inactivecasesearch = false
```

Back to Library

Limiting the Amount of Work Items That Are Returned by the Work Queue Search

The **maxworkqsearch** variable defines the maximum number of active and inactive cases that are returned by a work queue search. For example, if you have the **maxworkqsearch** variable set to 100 and the **inactivecasesearch** variable set to true, the maximum number of cases that can be returned is 200, 100 active cases and 100 inactive cases. The default is set to **100**. You can change this depending on your requirements.

If the **maxworkqsearch** parameter is set to 100, a work queue search that returns 100 or less work items is returned. If the work queue search returns a work item list greater than 100, a list of 100 work items is displayed and a **Number of items exceeds limit. Please refine your case description filter criteria** message is displayed. You should re-define the work queue search to reduce the number of work items that are returned.

Back to Library

Setting Case Start Precedence

This section determines the precedence order used to select which version of any sub-procedure is started by the case.

Value	Description	Value
swPrecedenceR	Released version only	0
swPrecedenceUR	Unreleased > Released version	1
swPrecedenceMR	Model > Released version	2
swPrecedenceUMR	Unreleased > Model > Released version	3
swPrecedenceMUR	Model > Unreleased > Released version	4

Using a Released version starts a live sub-case. Using an Unreleased or Model version starts a test sub-case.

By default, the precedence variable used is **0** (**swPrecedenceR**).

You must set the integer value in the code. For example, the following code sets a precedence order of Unreleased > Model > Released.

```
xsubprocprecedencetype = 3
```

For more information about procedure versions, see the *TIBCO iProcess Modeler - Procedure Management Guide*.

Configuring the Width of the Menu

The **menuwidth** entry determines the width of the menu on the left hand side of the window in the iProcess Client (JSP). To configure the width, amend the **menuwidth** entry in the configuration file, depending on your requirements. The default is **23%**.

Back to Library

Chapter 3 Customizing the TIBCO iProcess Client (JSP)

This chapter describes the ways in which you can customize the TIBCO iProcess Client (JSP).

Topics

Back to Library

Changing Styles and Images

It is easy to customize the basic style of, and images used in, the TIBCO iProcess Client (JSP):

- The *SWWebClientDir*/**global.css** file defines the basic styles used throughout the TIBCO iProcess Client (JSP), and can be easily modified to suit your preferences. For example, if you want to change the color of new work items, change the characteristics of the HIGHLIGHTED class.

  The **global.css** file is an external style sheet, which is linked:

  — from the TIBCO iProcess Client (JSP) JSP files that require it.

  — to the JSP versions of the iProcess step definitions, when they are generated, from the *SWStepConvertDir*\TemplateJSP.fc file. See the following section for more information.

- All image files used by the TIBCO iProcess Client (JSP) are contained in the *SWWebClientDir*/**images** directory, and referenced from the appropriate JSP pages.

Back to Library

Customizing the Step Definition Template

The step definition template file is actually part of the TIBCO Process Step Definition Converter software.

The file *SWStepConvertDir***TemplateJSP.fc** is a step definition template which is used by the Step Definition Converter when it generates a new JSP version of an iProcess step definition. (See What is the TIBCO Process Step Definition Converter? on page 5.)

The step definition template is a JSP file. As well as basic HTML and scripts, it can contain a number of special **step definition converter tags**, which can be used to insert iProcess step (and other step) data into the generated JSP page.

Step definition converter tags are delimited using the characters <$ and $>. The following table describes the tags that can be used.

Tag	Description
<$STAFFWAREFORM$>	Defines the location on the page where the iProcess step definition will be inserted. **Note: This tag must be placed within a BODY tag and not within any other grouping tags such as DIV or SPAN. Failure to do this can cause the step to behave incorrectly.**
<$SWCASEDESC$>	Will be replaced with code which contains the (text) value of the iProcess case description SW_CASEDESC.
<$SWCASENUM$>	Will be replaced with code which contains the (numeric) value of the iProcess case number SW_CASENUM.
<$SWSTEPNAME$>	Will be replaced with code which contains the (text) value of the iProcess step name SW_STEPNAME.
<$SWSTEPDESC$>	Will be replaced with code which contains the (text) value of the iProcess step description SW_STEPDESC.

Back to Library

Tag	Description
<$SWSTEPKEY$>	Will be replaced with code which contains the TIBCO iProcess Objects object key used to find the **SWStep** object. (For more information about this object see the TIBCO iProcess Objects Java Client **Help**.)
<$SWPROCNAME$>	Will be replaced with code which contains the (text) value of the iProcess procedure name SW_PROCNAME.
<$SWPROCDESC$>	Will be replaced with code which contains the (text) value of the iProcess procedure description SW_PROCDESC.
<$SWPROCKEY$>	Will be replaced with code which contains the TIBCO iProcess Objects object key used to find the **SWProc** object. (For more information about this object see the TIBCO iProcess Objects Java Client **Help**.)
<$SWNODENAME$>	Will be replaced with code which contains the (text) value of the iProcess node name SW_NODENAME.
<$SWNODEKEY$>	Will be replaced with code which contains the TIBCO iProcess Objects object key used to find the **SWNode** object. (For more information about this object see the TIBCO iProcess Objects Java Client **Help**.)

Except for <$STAFFWAREFORM$>, the step definition converter tags can be used anywhere in the step definition template - even within SCRIPT blocks.

Back to Library

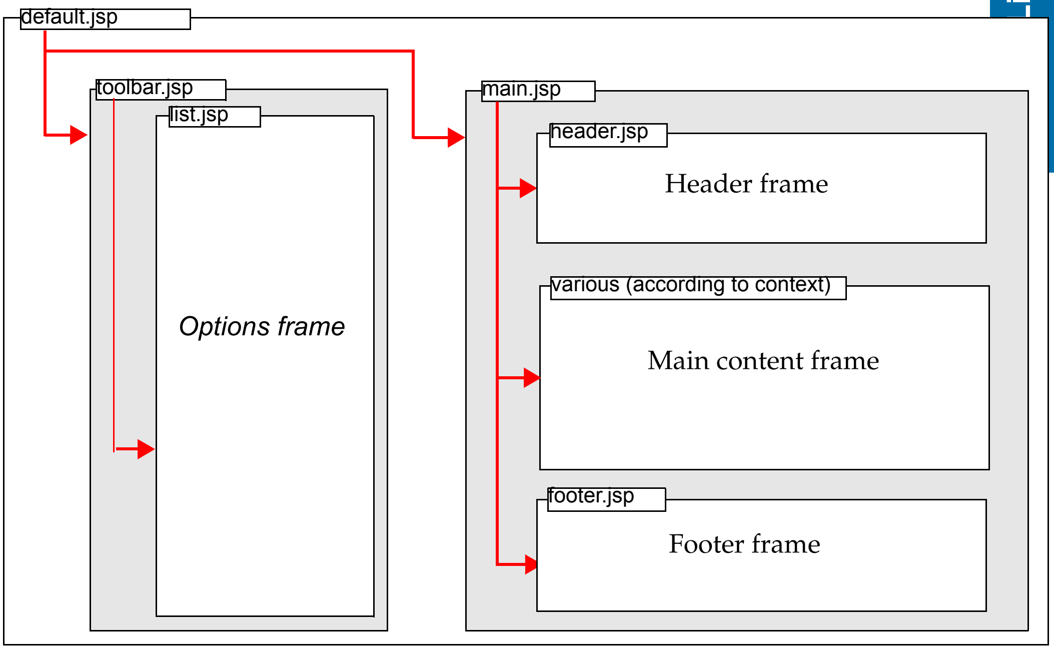
Customizing TIBCO iProcess Client (JSP) JSP Pages

The TIBCO iProcess Client (JSP) application consists of a number of JSP files (*SWWebClientDir*/***.jsp**). By modifying, replacing or extending these JSP files, you can customize the TIBCO iProcess Client (JSP) to your own requirements.

TIBCO iProcess Client (JSP) JSP Page Structure

The *SWWebClientDir*/***.jsp** files contain the necessary server-side (Java) and client-side (JavaScript) scripting to provide the TIBCO iProcess Client (JSP) functionality.

The following diagram gives a high-level overview of the JSP structure of the TIBCO iProcess Client (JSP).



For more detailed information about what the TIBCO iProcess Client (JSP) JSP pages do and the ways in which they interact, please refer to the files themselves

The following table lists the TIBCO iProcess Client (JSP) JSP files and provides a brief summary of what each one does.

.jsp File	Includes...	Description
auditcaselist.jsp	webclient.jsp	Displays the Audit Trail case list.
auditdisplay.jsp	webclient.jsp	Displays the Procedure Case List Display Criteria.
auditframe.jsp	none	Displays the audit trails as per the **auditpage** entry in the webclient.properties file.
auditpagecnt.jsp	none	Displays the audit trails as per the **auditpage** entry in the webclient.properties file.
audittraillist.jsp	webclient.jsp	Displays the audit trail for a particular case.
casefilter.jsp	webclient.jsp	Displays the Audit Case Filter criteria window.
casepredictlist.jsp	webclient.jsp	Displays the Case Prediction window.
casestart.jsp	webclient.jsp	Handles case starts (displays a step definition if appropriate, or calls **doCaseStart.jsp**).
datehandler.jsp	none	Provides date handling and the calendar object.
default.jsp	none	Performs iProcess Client (JSP) connection.
display.jsp	webclient.jsp	Provides display option for access criteria.
doauditdisplay.jsp	webclient.jsp	Displays the customized audit case list.
docasefilter.jsp	webclient.jsp	Builds the audit case list filter criteria.
docasestart.jsp	webclient.jsp	Performs case starts. This file is called from **CaseStart.jsp**.
dodefview.jsp	none	Provides the default view for access criteria.
doforward.jsp	webclient.jsp	Performs forwarding of work items. This file is called from Forward.jsp.
dologinout.jsp	webclient.jsp	Performs login/logouts. This file is called from **login.jsp** and **logout.jsp**.
error.jsp	none	Error handler.
filter.jsp	webclient.jsp	Displays filter option for access criteria

Back to Library

.jsp File	Includes...	Description
footer.jsp	none	Displays the footer area in the right-hand pane (when a queue is not displayed).
form.jsp	none	Handles attempts to open a work item.
formhandler.jsp	none	Provides client-side step handling functions (field validation and so on).
forward.jsp	webclient.jsp	Provides the forwarding page which includes a list of Work Queues that the selected items can be forwarded to.
head.jsp	webclient.jsp	Displays the graphics for the default html
header.jsp	webclient.jsp	Displays the header area in the right-hand pane.
info.jsp	webclient.jsp	Displays the current login status.
list.jsp	webclient.jsp	Displays the options available to the user (Work Queues, Audit Trail and Case Start).
list_at.jsp	webclient.jsp	Displays the Audit Trail list.
list_cs.jsp	webclient.jsp	Displays the Case Start list.
list_functions.jsp	webclient.jsp	Contains a set of helper functions used by the other lists.
list_wq.jsp	webclient.jsp	Displays the Work Queue list.
login.jsp	webclient.jsp	Performs a login.
logout.jsp	webclient.jsp	Performs a logout.
main.jsp	none	Displays the right-hand pane of the window.
memo.jsp	webclient.jsp	Displays memos.
openform.jsp	none	Displays third party forms.
participation.jsp	webclient.jsp	Provides participation setup.
qfooter.jsp	webclient.jsp, footer.jsp	Displays the footer area in the right-hand pane when a work queue is displayed.
queue.jsp	webclient.jsp	Displays the work items list.

Back to Library

.jsp File	Includes...	Description
queueadmin.jsp	webclient.jsp	Displays the list of participants of a work queue.
queueadminlist.jsp	webclient.jsp	Displays the list of users being supervised.
queueframe.jsp	none	Contains queue.jsp and qfooter.jsp.
queuesummary.jsp	webclient.jsp	Displays the work queue summary.
redirection.jsp	webclient.jsp	Provides redirection set up.
sort.jsp	webclient.jsp	Provides sort option for access criteria
status.jsp	none	Performs a search of work items (search by case description). Displays the priority and the audit trail for the selected work item.
strapline.jsp	webclient.jsp	Displays the bottom of the page, queue summary and access criteria.
toolbar.jsp	none	Displays the left-hand pane of the window.
tools.jsp	webclient.jsp	Provides the tools available to the user (Login, Logout, Help and Refresh).
webclient.jsp	none	Provides a core set of client-side functions. See page 39 for more information.
workitem.jsp	none	Provides Open Next Workitem.

The following table lists the files that are generated by the Step Definition Converter, and which must then be copied to the *SWWebClientDir*/*nodename*/*PROCNAME* step definition cache. See What is the TIBCO Process Step Definition Converter? on page 5 for more information.

.jsp File	Includes...	Description
*stepName***.jsp**	datehandler.jsp, formhandler.jsp, webclient.jsp	The step definition file for a particular step, as generated by the Step Definition Converter.
*stepName***Close.jsp**	webclient.jsp	The persist file (which handles the keep and release functions) for a particular step, as generated by the Step Definition Converter.

Back to Library

Core Scripting Functions

Some core functions, which are used to carry out the more complex tasks needed, have been centralized into a specific file:

- Server-side scripting functions are provided in the file *SWWebClientBeansDir*/**BeanSWSEO.class**.
- Client-side functions are provided in the file *SWWebClientDir*/**webclient.jsp**.

The following sections describe the functions provided by these files in more detail.

If you modify the TIBCO iProcess Client (JSP) JSP pages you are strongly recommended to use the standard functions provided (in webclient.jsp) rather than write new functions from scratch.

Back to Library

Server-Side Functions

For more detailed information, please refer to the beans documentation, which is available in the *SWWebClientConfigDir* / **docs/index.html** file.

Client-Side Functions

The following table summarizes the client-side scripting functions provided in the file *SWWebClientDir*/**webclient.jsp**. (For more information, please refer to the *SWWebClientDir*/**webclient.jsp** file.)

Function(s)	Description
showMain()	Displays the right hand pane of the TIBCO iProcess Client (JSP) window (the header, main panel and footer). It is recommended that this function is always used when altering the display.

Integrating the TIBCO iProcess Client (JSP) with Other Applications

The TIBCO iProcess Client (JSP) can be integrated with other technologies, either by using its server-based JSP pages to access other JavaBeans or Enterprise JavaBeans (EJBs), or by using client-side integration.

The following sections discuss some features which can be used in client-side integrations.

Adding Code to the Step Definition Template

You can carry out integration by adding appropriate script to the TIBCO Process Step Definition Converter's Step Definition Template (*SWStepConvertDir***TemplateJSP.fc**). If you do this, you should note the following points:

- Step definition converter tags (see Customizing the Step Definition Template on page 33) can be used to get commonly used iProcess data into scripting. This can be used in JavaScript to determine which step is being accessed.
- The TIBCO iProcess Client (JSP) stores all fields used within a step as an associative array. The array contains objects of type **field**, which have 3 main properties: **Name**, **Value** and **Type**. You can use this array to dynamically access field data from TIBCO iProcess Client (JSP) forms.
  - Your code can access this array (after the **onLoad** event has completed) using the construct **fields["***fieldname***"]**, where *fieldname* is the name of the field you are accessing.
  - If you want to update a field value, use the function **fields["***fieldname***"].updatefield(null)**.

For more information about the **Field** object, see the *SWWebClientDir***formhandler.jsp** file.

Controlling Step Flow

If you are modifying code within the Step Definition Template (*SWStepConvertDir***TemplateJSP.fc**), you should note the following points:

- The step will override some event handlers when the conversion is carried out - notably the **Body.onLoad** event handler. This event handler is often used within web pages, but must be left blank so that the Step Definition Converter can make use of it.

Back to Library

To work around this problem, you can call other functions from the **init()** function in the **formhandler.jsp** file. (The **init()** function is called by the **onLoad** event handler.)

- You are recommended to add any other functions that are called at the very end of the **init()** function. This ensures that the **fields** array and other important objects are fully created before any external access is carried out.

Adding Code to an iProcess Step

You can add HTML or JavaScript code to the iProcess step definition (for example, using the iProcess Step Definer). Any HTML tagged text will be preserved intact when the JSP form is generated by the TIBCO Process Step Definition Converter.

You can use this technique to generate "per-step" HTML code.

Back to Library

Back to Library

Appendix A Step Definition Converter Mappings

The following tables show how iProcess step definition elements are mapped to HTML elements and attributes when the step is converted to a JSP file.

Back to Library

General Mappings

iProcess step definition element	HTML elements and attributes
Normal text	Shown like normal text, using Courier or Courier New font.
Field markings (optional)	Optional and calculated fields are shown like HTML form fields, using the <input type="text" …> element and attribute. They have the 'class=OPT' attribute. The width of the HTML field is equal to the width of the Staffware field, with a maximum of 40 characters.
Field markings (required)	Required fields are shown like optional fields, but they have the 'class=REQ' instead of the 'class=OPT' attribute, resulting in a red line around the box. Additionally, JavaScript code checks if the field has been given a value.
Field markings (display)	Display fields are shown in the same way as optional fields, with additionally the 'readonly' HTML attribute.
Field markings (calculated)	For calculated fields the same code is generated as for optional fields.
Field markings (hidden)	No HTML code is generated for hidden fields.
Field markings (embedded)	Embedded fields are shown like normal text.
Field markings (numeric)	Numeric fields are given an initial value of '0', when applicable. In addition, JavaScript code checks if the field contains only numerical data.
Field markings (date)	Date fields are given an initial value of the current date, when applicable.
Field markings (time)	Time fields are given an initial value of the current time, when applicable.
Field markings (memos and attachments)	Memos and attachments are not supported by SPO. However, the same HTML code is generated as for normal text fields.
Field markings (help)	A 'title=…' attribute (Internet Explorer specific) is used to specify pop-up help for the field.

Back to Library

iProcess step definition element	HTML elements and attributes
Field markings (validations)	Drop down lists can have the following class attributes: • ‘class=REQ’ • ‘class=OPT’ • ‘class=DIS’
Field markings (commands)	Commands are not supported, no HTML code is generated.
Fonts	For every font change a <font …> HTML element is used.
Font names	Different fonts are supported using the ‘face=…’ attribute. Some fonts are mapped (see below).
Font sizes	Font sizes are supported using the ‘size=…’ (old style, see below) and ‘style=“font-size: …;”’ (new style) HTML attributes.
Font colors	Font colors are supported using the ‘color=…’ attribute. No named colors are used.
Font attributes	Font attributes are supported using <b>, </b>, <i>, </i>, <u> and </u> tags. For each font change, either an opening or a closing tag is inserted for each attribute.

Back to Library

Font Mappings

Input font	HTML font
TimesRoman	TimesRoman, Times Roman, Times New Roman
Courier	Courier New
Arial	Helvetica, Swiss, Arial

Back to Library

Font Size Mappings

Point size (pt.)	HTML size (old style)
0-9	1
10-11	2
12-13	3
14-16	4
17-21	5
22-30	6
31+	7

Back to Library

Back to Library