TIBCO iProcess™ Client (ASP)

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 (ASP)**.

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

Back to Library

How to Use This Guide

You should read Chapter 1 first. This describes the iProcess Client (ASP) 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 (ASP).
- Chapter 3 describes how to customize the iProcess Client (ASP).
- Appendix A shows how iProcess step elements are mapped to HTML elements and attributes when a step is converted to an ASP page.

Target Audience

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

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

- Active Sever Pages (ASP)
- General principles of web server administration
- Internet Information Server (IIS)
- TIBCO iProcess Suite™.

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

- VBScript
- JavaScript
- 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 TIBCO iProcess Client (ASP) software from the following sources:

- The *TIBCO iProcess Client (ASP) 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 (ASP) Installation Guide* for more information about this file.
- The *TIBCO iProcess Client (ASP): User's Guide* explains how to use the iProcess Client (ASP).
- Online help is available within the iProcess Client (ASP).
- 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

Documentation Conventions

The directory where the iProcess Client (ASP) is installed is referred to in this guide as *SWWebClientDir*.

The TIBCO Process™ Engine and the TIBCO iProcess™ Engine are referred to in this guide by the generic term, iProcess server.

Back to Library

Back to Library

Chapter 1 Overview

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

Topics

Back to Library

What is the TIBCO iProcess Client (ASP)?

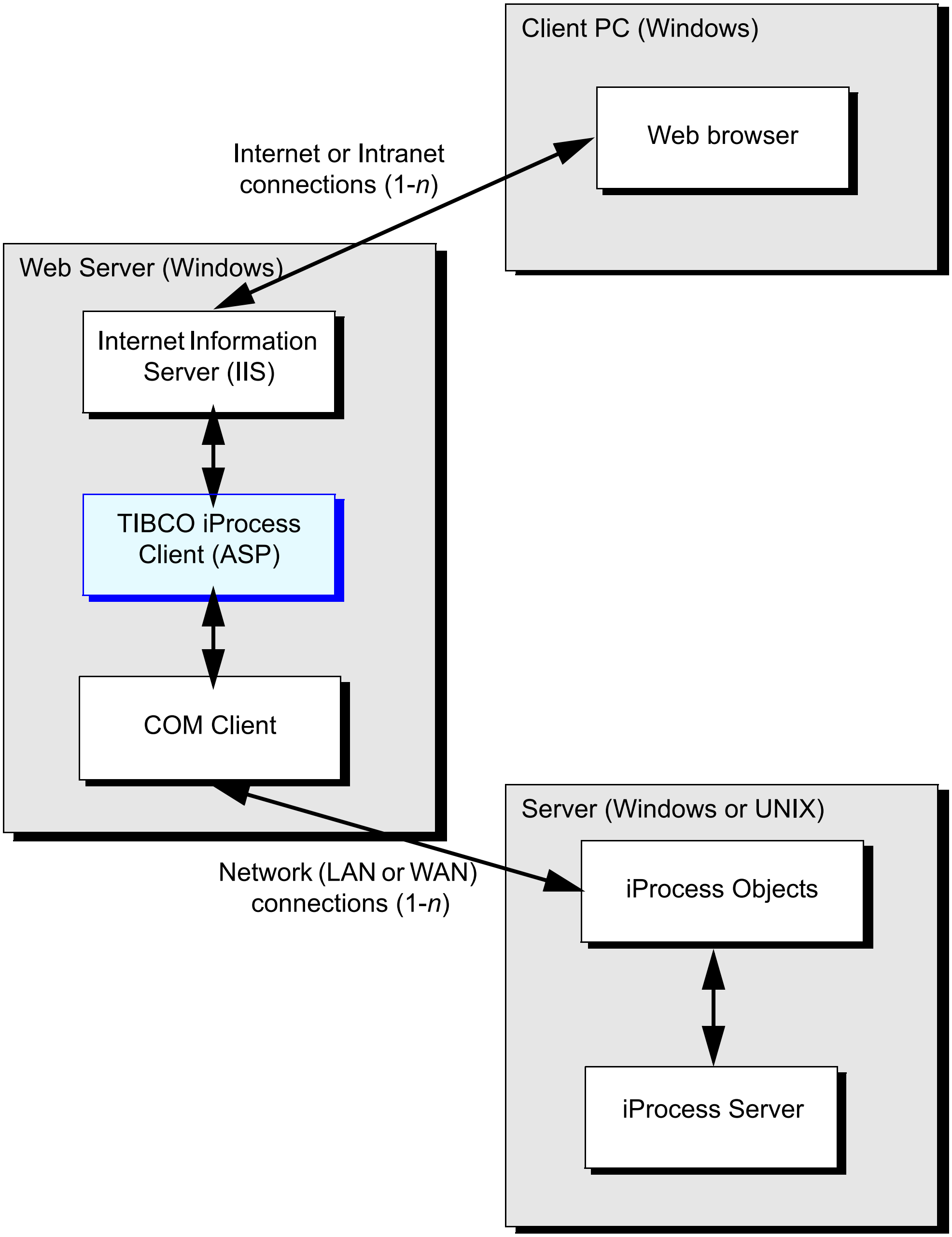
The iProcess Client (ASP) is an ASP-based web server application that enables organizations to extend their iProcess systems over the internet and intranets, enabling staff to participate in a procedure irrespective of where they are or what time it is.

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

The iProcess Client (ASP) uses iProcess Objects to communicate with iProcess (by calling TIBCO iProcess™ Objects COM Client objects from VBScript code in ASP pages). For more information about TIBCO iProcess Objects please refer to the **Help** supplied with the TIBCO iProcess Objects COM Client.

Back to Library

The Architecture of the TIBCO iProcess Client (ASP)


Client PC (Windows)
Web browser
Internet or Intranet connections (1-*n*)
Web Server (Windows)
Internet Information Server (IIS)
TIBCO iProcess Client (ASP)
COM Client
Network (LAN or WAN) connections (1-*n*)
Server (Windows or UNIX)
iProcess Objects
iProcess Server

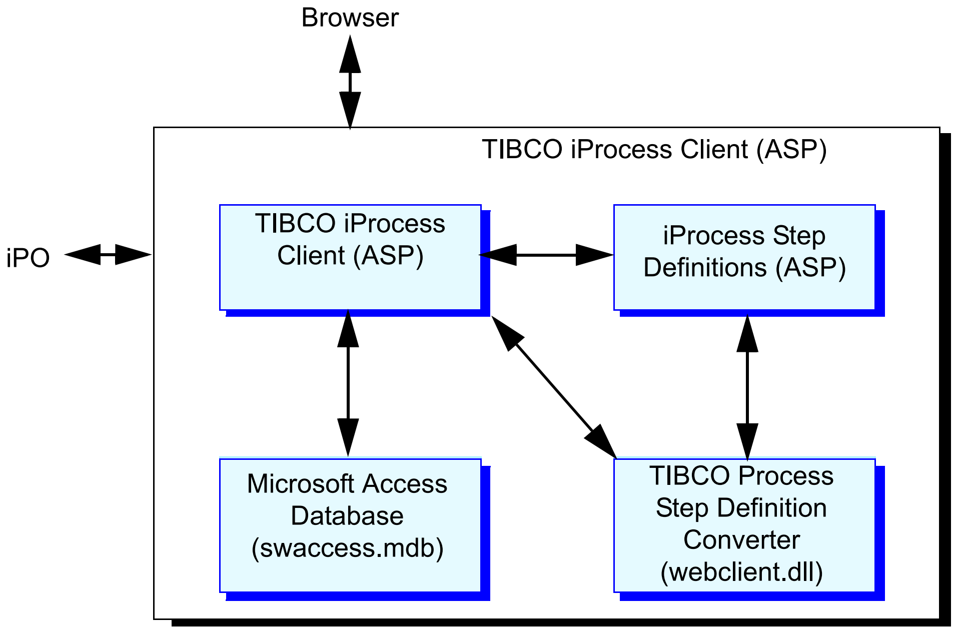
Back to Library

How the iProcess Client (ASP) Works

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

- A set of interrelated ASP pages containing HTML, VBScript and JavaScript, which provide the necessary logic and user interface.
- A Step Definition Converter. The Step Definition Converter reads an iProcess step definition, obtained from TIBCO iProcess Objects (iPO), and converts it into ASP pages which can be sent to the browser.
- A step definition cache, containing ASP pages representing iProcess steps, generated dynamically by the Step Definition Converter.

Back to Library


Browser
TIBCO iProcess Client (ASP)
iPO
TIBCO iProcess Client (ASP)
iProcess Step Definitions (ASP)
Microsoft Access Database (swaccess.mdb)
TIBCO Process Step Definition Converter (webclient.dll)

When the iProcess Client (ASP) receives a request to display a step definition from the browser, it looks for a step definition file (see below) of that name in the step definition cache, and checks to see if the file is fresh. If the file does not exist, or is stale, the iProcess Client (ASP) calls the Step Definition Converter to (re)generate the step definition in the step definition cache.

The Step Definition Converter

The Step Definition Converter is a **.dll** file (called **webclient.dll**) with COM interfaces that convert iProcess forms into ASP pages. The **webclient.dll** file can be found in the Windows system directory (for example, **winnt****system32****webclient.dll**).

The Step Definition Converter reads an iProcess step definition (passed to it from TIBCO iProcess Objects) and converts it into ASP. Using a step definition template called **template.fc** (also found in the Windows system directory, for example, **\Winnt\System32\template.fc**) as a wrapper, it creates the following ASP pages in the step definition cache:

- a **step definition file**, *SWWebClientDir*\NodeName\ProcedureName\*StepName***.asp**. When it is processed by the web server, the step definition file will show the step 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 form was generated.

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

  When the iProcess Client (ASP) accesses the step definition file, it uses this information to check whether the file is fresh, or whether it should be regenerated.

- a **persistence file**, *SWWebClientDir*\NodeName\ProcedureName\*StepName***Close.asp**. 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.

The **webclient.dll** file is called from the **GotoForm()** function in the **standard.asp** file. For more information see Customizing TIBCO iProcess Client (ASP) ASP Pages on page 35. For more information about the form template, see Customizing the Step Definition Template on page 33.

Back to Library

Security

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

If you intend to use the iProcess Client (ASP) to transmit sensitive data - either passwords or form data, you are advised to use secure protocols, such as Secure Socket Layer (SSL), to establish a secure connection between the browser and IIS.

For more information about the use of secured connections, please consult your IIS documentation.

Back to Library

Chapter 2 Configuring the iProcess Client (ASP)

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

Topics

Back to Library

Using the URL Login

In addition to logging in using the standard login dialog, you can login to the iProcess Client (ASP) using a URL. The username, password and nodetag parameters are passed to the web server in the URL rather than through the standard login dialog. 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*=*iprocess server*|*nodename*|*spo server IP address*|*port number*

where:

- *web server* is the name of the computer where IIS 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.

Below is an example URL:

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

Back to Library

where the parameters are:

Parameter	Description
Web Server	uk_joannablap
iProcess Username	joannab
iProcess Password	mypassword
Nodetag	uk_joannablap
Nodename	sw9010
TIBCO iProcess Objects IP Address	127.0.0.1
Port Number	55555

Back to Library

Modifying the iProcess Client (ASP) Configuration File

The iProcess Client (ASP) configuration file, *SWWebClientDir***config.asp**, enables you to configure a number of different areas of the iProcess Client (ASP).

The following sections describe the different areas that you can configure. The areas are described in the order that they appear in the file.

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

Back to Library

Text Strings

Most text strings used in the iProcess Client (ASP) are defined as variables, to make it easy to either translate the user interface or tailor particular areas. For example:

```
dim text(121)
SWID_USERNAME= 0
SWID_SERVER= 1
.
.
text(SWID_USERNAME)= "Username"
text(SWID_SERVER)= "Server"
.
.
```

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

This section determines the number of items returned per block from the TIBCO iProcess Objects server. There are three parameters:

- **xWQBlockSize** configures the number of work items returned.
- **xACBlockSize** configures the number of audit trails returned.
- **xGenBlockSize** configures the number of other items returned that use XLISTS. For example, iProcess lists.

All three parameters are set to **20** by default. You may want to change these parameters, depending on your requirements. For example, work items and audit trails can vary in size, so if your iProcess server has a large number of audit trails but only a few work items, you may want to increase the **xACBlockSize** parameter setting and decrease the **xWQBlockSize** setting.

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

The **xMaxAuditList** 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 **xMaxAuditList** 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.

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

The lowest permissable value for the **xMaxAuditList** parameter is **1**.

Back to Library

Setting Logging Options

The **logfilename** variable defines the full pathname of the log file created when logging is enabled (by default **c:\webclient.log**)

The log file displays the following information:

- date/time of the entry
- username and IP address of the client machine, and session ID (for tracing the user's activities)
- entry type. This is a text string as defined in the **logtypes** array - for example:

```
logtypes(LOG_WARN,1) = "WARNING"
```

- problem description (as returned by the code).

The **logtypes** array defines the level of logging activity to be performed by the iProcess Client (ASP). Logging of the following areas can be enabled (= true) or disabled (= false).

Variable	Description
LOG_ERR	Logs errors. An error is a problem which the iProcess Client (ASP) cannot handle, and which forces it to abort the user's session.

Variable	Description
LOG_WARN	Logs warnings. A warning is a problem which the iProcess Client (ASP) cannot handle, but which does not prevent the user's session from continuing.
LOG_CONNECT	Logs each TIBCO iProcess Objects login attempt, which occurs every time an ASP page is loaded (because the iProcess Client (ASP) does not persist its connection to the TIBCO iProcess Objects server).
LOG_AUTH	Logs authentication attempts (when a user logs on at the start of a session and logs off at the end) and their success or failure.
LOG_TIMING	Logs the length of time the ASP Web Pages take to execute.

It is recommended that **LOG_ERR** and **LOG_WARN** are always switched on.

Back to Library

Defining if Locked Work Items can be Forced Open

Work items are marked as locked (by the TIBCO iProcess Objects COM 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** variable 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 iProcess Client (ASP) 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 TIBCO iProcess Workspace.

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

Adding iProcess Servers to the Static Login List

This section defines iProcess servers that should always be displayed in the list of available nodes on the login screen, irrespective of whether they can be contacted by the TIBCO iProcess Objects COM Client.

The drop down list of available nodes which is displayed on the login screen, is generated dynamically. The TIBCO iProcess Objects COM Client sends a UDP broadcast message to detectTIBCO 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.

Back to Library

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

To add an iProcess server to the static list:

1. Increment the **numstaticservers** variable by 1.
2. Increment the **dim** statements for **staticserver**, **port** and **ipaddr** by 1 (if necessary).
3. Add an entry to the array which defines the server details, using the following variables.

Variable	Description
staticserver(*n*)	The name of the iProcess server, in the form: *nodename@machinename*
port(*n*)	The port number of the machine hosting the iProcess server. **Note: If the TCPServiceName parameter in the TIBCO iProcess Objects server configuration file is set to DEFAULT, set** port(n)=0

Variable	Description
ipaddr(*n*)	The IP address of the machine hosting the iProcess server. **Note: If the TCPServiceName parameter in the TIBCO iProcess Objects server configuration file is set to DEFAULT, set** ipaddr(n)=""

Specifying a port and IP address causes the iProcess Client (ASP) to use direct TCP communications with the specified server.

For example, the following code fragment adds the iProcess server **Enterprise1** on machine **PURCH1** to the static login list. A port and IP address are specified, so the iProcess Client (ASP) will communicate directly with this server.

```
numstaticservers = 1
staticserver(0) = "Enterprise1@PURCH1"
port(0)= 2556
ipaddr(0)= “10.10.2.21”
```

Back to Library

Enabling Automatic Login to iProcess Servers

This section defines iProcess servers that will be automatically logged into when the iProcess Client (ASP) is accessed.

To enable automatic login to a server:

1. Increment the **numautologins** variable by 1.
2. Increment the **dim** statements for **autoservername**, **autousername** and **autopassword** by 1 (if necessary).
3. Add an entry to the array which defines the login information for the server, using the following variables.

Variable	Description
autoservername(*n*)	The name of the iProcess server, in the form: *nodename@machinename*

Variable	Description
autousername(*n*)	The iProcess user name to be used to log in. You can either specify this explicitly, or use the LOGON_USER server variable to log on using the current Windows username. **Note: If you are using LOGON_USER, login password checking on the iProcess server should be disabled.**
autopassword(*n*)	The iProcess password for this login. **Note: You can leave this blank if login password checking is disabled on the iProcess server.**

Back to Library

For example, the following code fragment enables automatic login to the iProcess server **Enterprise1** on machine **PURCH1**. The current O/S username will be used to login, with no password supplied. (Password checking must therefore be disabled on the iProcess server for the login to succeed.)

```
numautologins = 1
autoservername(0) = "Enterprise1@PURCH1"
autousername(0) = request.servervariable("LOGON_USER")
autopassword(0) = ""
```

Login Dialog Options

This section provides two variables which can be used to control the login options available to the user.

Variable	Description
SW_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.

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

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.

Variable	Description	SPO ASCII 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 **asc(“0”)** (**swPrecedenceR**).

You must set the TIBCO iProcess Objects integer in the code because this is how case precedence is defined in TIBCO iProcess Objects. For example, the following code sets a precedence order of Unreleased > Model > Released.

```
xSubProcPrecedenceType = asc("3")
```

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

Enabling Forms Generation

This section defines whether or not the ASP versions of iProcess forms in the forms cache should be (re)generated from their iProcess originals when they are accessed.

Forms generation is enabled by default. You may want to disable it if you do not want your current ASP forms to be overwritten - for example, if you use ASP versions of your iProcess forms which have been either modified since generation, or created by other means.

For more information about forms generation please see *The Step Definition Converter* on page 4.

To enable or disable forms generation set the **genforms** variable to **true** or **false**.

Back to Library

Defining TIBCO iProcess Client (ASP) ASP Pages

This section includes 4 variables which define the ASP files used to generate the work queues, audit trails and forms displayed by the iProcess Client (ASP). You can change the default values assigned to these variables if you want to replace these pages.

Variable	Defines the ASP page used to...
workqueuelistpage	Display the contents of a work queue for a procedure (when called from **list.asp**). Default value is **Queue.asp**.
formpage	Display the form for a work item (when called from **Queue.asp**). Default value is **Form.asp**.
auditcaselistpage	Display the audit trail case list for a procedure (when called from **list.asp**). Default value is **AuditCaseList.asp**.
casestartpage	Display the form for a case start (when called from **list.asp**). Default value is **CaseStart.asp**.

For more information about customizing the iProcess Client (ASP) ASP pages, please see *Customizing TIBCO iProcess Client (ASP) ASP Pages* on page 33.

Setting the Number of Items Per Page

This section determines how many items are sent to the browser to be displayed on one page in the Web Client window. For example, the default is set to 20 items so a work queue list will only display 20 work items on one page. If there are more work items, they will be displayed on further pages.

```
wqpage = 20
```

You can turn this feature off by setting this value to **-1**. This means that all the items will be sent to be displayed on one page. If you have a large number of items being sent from the iProcess server, this can result in pages taking longer to display.

Setting the Work Item List Refresh Interval

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

```
ListRefreshInterval = 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**.

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 (ASP) 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.

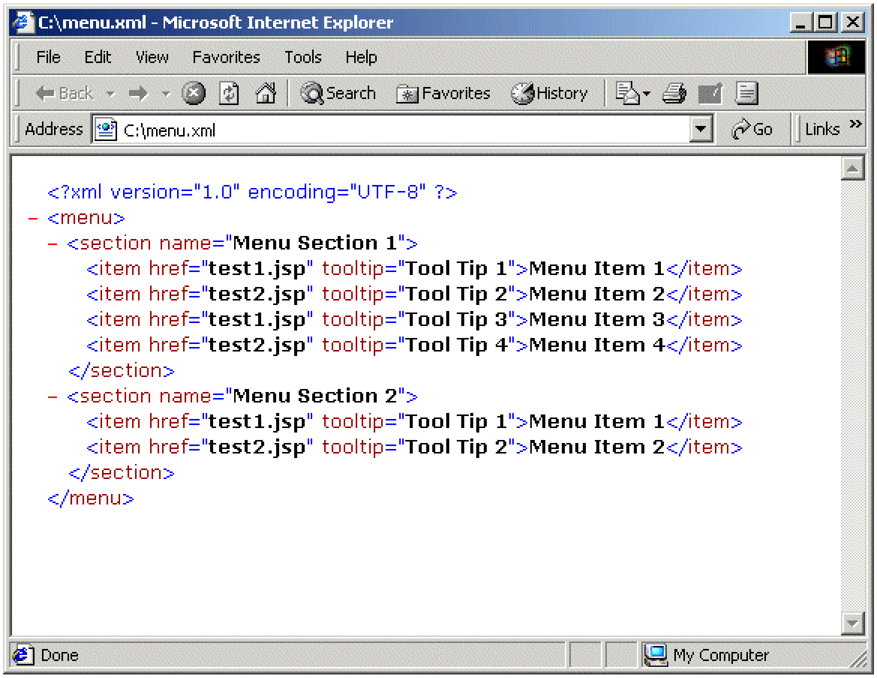
You must create a file(s) that contains the code for the new menu item. See *Configuring the TIBCO iProcess Client (ASP)* on page 7 for more information.

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 cannot rename the **menu.xml** file to another name.

Once the **menu.xml** file is amended, you need to enable the **menu.xml** file by setting the **menusection** parameter in the **config.asp** file.

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>
```

Back to Library

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, **.asp** 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.

Enabling the menu.xml File

To enable the **menu.xml** file, amend the **menusection** entry in the configuration file from **false** to **true**. For example,

```
menusection=true
```

Changing the Default Folder Display

This section configures the default folders that are displayed for individual users.

To change the default folder display, do the following:

1. Set the **numberofmenuoptions** parameter to be the same as the number of **menuoption** entries in the **config.asp**.
2. 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 **numberofmenuoptions** is set to 0, the default folder display is used. The following folders are displayed by default:

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

This section is in the following format:

```
numberofmenuoptions=0
```

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

where:

- **numberofmenuoptions** is the number of menu options you want to define. For example, if you want to define 5 menuoptions, amend the **numberofmenuoptions** parameter as follows:

```
numberofmenuoptions=5
```

- **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.

Back to Library

- 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.

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:

```
numberofmenuoptions=2

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.

```
numberofmenuoptions=1

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

Changing the Default View

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

To change the default view, amend the **firstpage** entry in the **config.asp** file in the following format:

```
firstpage="filename"
```

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

```
firstpage="firstpage.asp"
```

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

Back to Library

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 (ASP): User’s Guide*.

Specifying the Type of Cases to Search For

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

- **0** to only search for open cases, or
- **1** to search for both open and closed cases.

The default is **0**. For example:

```
xInactiveCaseSearch = 0
```

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

The **xMaxWorkItemSearch** variable defines the maximum number of work items that are returned by a work queue search. The default is set to **100**. You can change this depending on your requirements.

If the **xMaxWorkItemSearch** parameter is set to 100, a work queue search that returns 100 or less work items will be returned. If the work queue search will return 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 will be returned.

Configuring Dynamic Lists

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

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

Back to Library

Pre-loading lists can affect performance when you have a large number of users logging in to the TIBCO iProcess Client (ASP). 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
```

Back to Library

Changing the Default Work Queue Display

The *SWWebClientDir***swaccess.mdb** defines the iProcess fields that are displayed in the work queue.

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

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

To change the default work queue display, do the following:

1. Open the *SWWebClientDir***swaccess.mdb** file in MS Access. The **swaccess** database window is displayed.
2. From the **swaccess** database window, click **swfields**. Click **Open** to open the **swfields** table in datasheet view.
3. The **displaydefaultorder** column defines the work queue display. The numeric values in these columns correspond to the order in which the columns are displayed in the work queue. For example, SW_STATUS has a value of **1**, and SW_CASEREF has a value of **2** and so on. If a field has a numeric value of **0**, it is not displayed in the work queue at all.

   Amend the numeric values in the **displaydefaultorder** column depending on your requirements.
4. Click **Save** and exit from MS Access.

Back to Library

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

iProcess Field	Description
SW_ARRIVAL	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 **displaydefaultorder** entries in the **swaccess.mdb** file.

Back to Library

Setting the Session Inactivity Timeout Value

The *SWWebClientDir***global.asa** file defines the inactivity timeout value for iProcess Client (ASP) users. By default it is:

```
Session.Timeout=60
```

If any work items are open when the session times out they will be left locked. For more information about how locked work items are handled, see *Defining if Locked Work Items can be Forced Open* on page 12.

Back to Library

Managing TIBCO iProcess Client (ASP) Availability

It is obviously important that the iProcess Client (ASP) is, whenever possible, always available to users.

To ensure this, the Web Server you are using should be configured so that each instance of the iProcess Client (ASP) runs in its own memory space. If an iProcess Client (ASP) instance crashes, other instances (or applications, or the Web server itself) are not affected.

Back to Library

Managing Caching

Web pages are frequently cached (either by the browser, proxy server or web server) as a means of increasing web site response or throughput.

However, when using the iProcess Client (ASP), cached pages are undesirable because they may contain out-of-date or misleading information.

⚠ It is also worth notifying users that if they see unexpected results when using the iProcess Client (ASP), they can refresh an individual panel by clicking **Refresh** Queue (**not** the browser's **Refresh**). This will nearly always resolve the problem.

To minimize the chance of a page being cached all browsers that access the iProcess Client (ASP) should have their cache set to check for newer versions of stored pages every time the page is visited.

Back to Library

Back to Library

Chapter 3 Customizing the iProcess Client (ASP)

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

Topics

Back to Library

Changing Styles and Images

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

- The *SWWebClientDir***global.css** file defines the basic styles used throughout the iProcess Client (ASP), 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 iProcess Client (ASP) ASP files that require it.

  — to the ASP versions of the iProcess step definitions, when they are generated, from the **\Winnt\System32\Template.fc** file. See the following section for more information.

- The **IEForm.css** is used to customize Internet Explorer forms. It sets up the default font family and background so that the iProcess step definitions display correctly in the browser.
- All image files used by the iProcess Client (ASP) are contained in the *SWWebClientDir***images** directory, and referenced from the appropriate ASP pages.

Back to Library

Customizing the Step Definition Template

The file **Winnt\System32\Template.fc** is a step definition template which is used by the Step Definition Converter when it generates a new ASP version of an iProcess step definition. (See The Step Definition Converter on page 5.)

The step definition template is an HTML 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 ASP 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 definition 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.
<$SWSTEPKEY$>	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 COM Client **Help**.)

Back to Library

Tag	Description
<$SWPROCNAME$>	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 COM 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 COM 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 (ASP) ASP Pages

The iProcess Client (ASP) application consists of a number of Active Server Page files (in *SWWebClientDir****.asp**). By modifying, replacing or extending these ASP pages, you can customize the iProcess Client (ASP) to your own requirements.

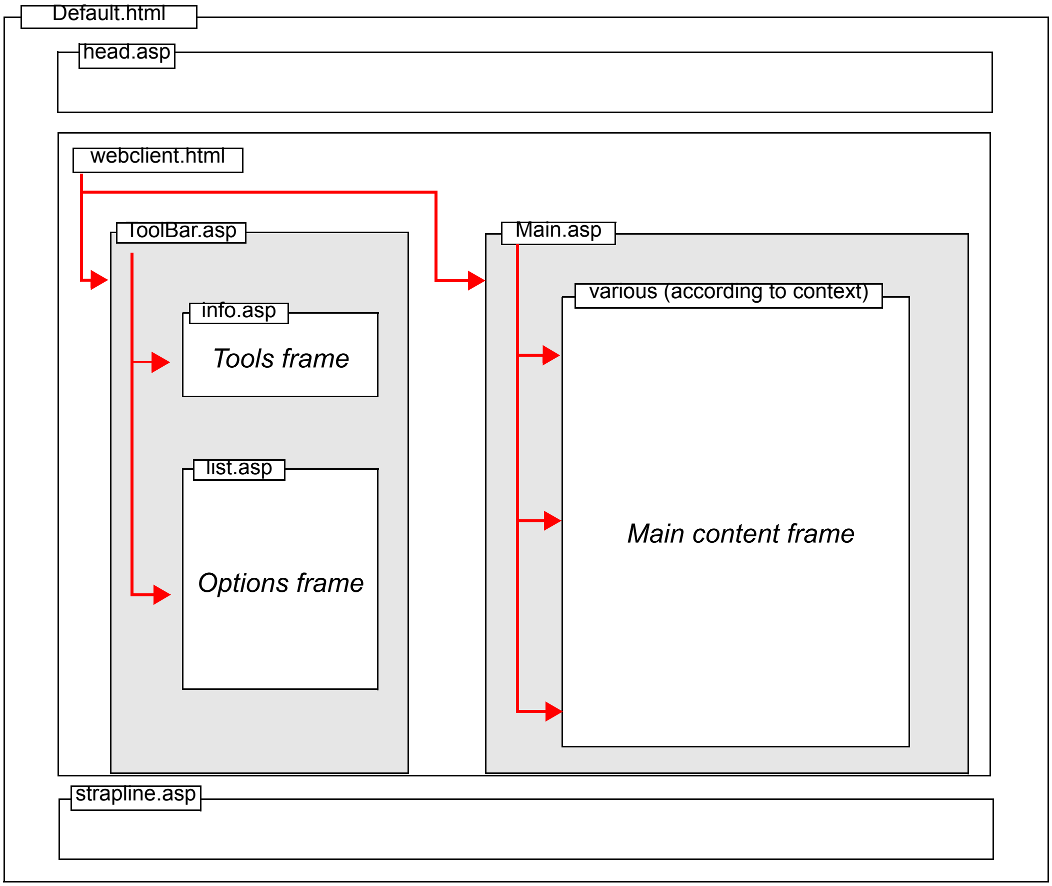
TIBCO iProcess Client (ASP) ASP Page Structure

The *SWWebClientDir****.asp** files contain the necessary server-side (VBScript) and client-side (JavaScript) scripting to provide the iProcess Client (ASP) functionality.

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

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

Back to Library



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

.asp File	Includes...	Description
AuditCaseList.asp	standard.asp	Displays the Audit Trail case list.
auditdisplay.asp	standard.asp	Displays the Procedure Case List Display Criteria.
auditframe.asp	None	Displays the audit trails as per the **auditpage** parameter in the **config.asp**.
auditpagecnt.asp	standard.asp	Displays the audit trails as per the **audipage** parameter in the **config.asp**.

Back to Library

.asp File	Includes...	Description
AuditTrailList.asp	standard.asp	Displays the audit trail for a particular case.
casefilter.asp	standard.asp	Displays the Audit Case Filter criteria window.
CasePredictList.asp	standard.asp	Displays the Case Prediction window.
CaseStart.asp	standard.asp	Handles case starts (displays a step definition if appropriate, or calls **doCaseStart.asp**).
config.asp	None	Main configuration file. For more information, see page 10.
DateHandler.asp	config.asp	Provides date handling.
default.asp	config.asp	Performs the iProcess Client (ASP) connection.
display.asp	standard.asp	Provides display option for access criteria.
doauditdisplay.asp	standard.asp	Displays the customized audit case list view.
docasefilter.asp	standard.asp	Builds the audit case list filter criteria.
doCaseStart.asp	standard.asp	Performs case starts. This file is called from **CaseStart.asp**.
doforward.asp	standard.asp	Performs forwarding of work items. This file is called from **Forward.asp**.
dologinout.asp	standard.asp	Performs login/logouts. This file is called from **login.asp** and **Logout.asp**.
Error.asp	standard.asp	Error handler.
filter.asp	standard.asp	Provides filter option for access criteria.
Footer.asp	standard.asp	This file is no longer used.
Form.asp	standard.asp	Handles attempts to open a work item.
FormHandler.asp	config.asp	Provides client-side form handling functions (field validation and so on).
forward.asp	standard.asp	Provides the forwarding page which includes a list of Work Queues that the selected items can be forwarded to.

Back to Library

.asp File	Includes...	Description
head.asp	None	Displays the graphics for the default.asp.
Header.asp	standard.asp	This file is no longer used.
list.asp	standard.asp	Displays the options available to the user (Work Queues, Audit Trail, Case Start, Tools and Nodes).
list_at.asp	standard.asp	Displays the Audit Trail list.
list_cs.asp	standard.asp	Displays the Case Start list.
list_functions.asp	standard.asp	Contains a set of helper functions used by the other lists.
list_wq.asp	standard.asp	Displays the Work Queue list.
login.asp	standard.asp	Performs a login.
Logout.asp	standard.asp	Performs a logout.
Main.asp	standard.asp	Displays the right-hand pane of the window.
memo.asp	standard.asp	Displays memos.
participation.asp	standard.asp	Provides participation set up.
QFooter.asp	standard.asp, Footer.asp	Displays the footer area in the right-hand pane when a work queue is displayed.
Queue.asp	standard.asp	Displays the work items list.
QueueAdmin.asp	standard.asp	Displays the list of participants of a work queue.
QueueAdminList.asp	standard.asp	Displays the list of users being supervised.
queueframe.asp	standard.asp	Contains queue.asp and qfooter.asp.
QueueSummary.asp	standard.asp	Displays the work queue summary.
redirection.asp	standard.asp	Provides redirection set up.
sort.asp	standard.asp	Provides sort option for access criteria.
standard.asp	config.asp	Provides a core set of server-side functions. Included in most other **.asp** files.

Back to Library

.asp File	Includes...	Description
Status.asp	standard.asp	Performs a search of work items (search by case description). Displays the priority and the audit trail for the selected work item.
strapline.asp	standard.asp	Displays the bottom of the page, queue summary and access criteria.
ToolBar.asp	none	Displays the left-hand pane of the window.
tools.asp	standard.asp	Provides the tools available to the user (Login, Logout, Help and Refresh).
webclient.asp	config.asp	Provides a core set of client-side functions.
WorkItem.asp	standard.asp	This file is no longer used.

Back to Library

The following table lists the files which are generated by the Step Definition Converter and stored in the step definition cache. See The Step Definition Converter on page 5 for more information.

.asp File	Includes...	Description
SWWebClientDir\ *nodeName*\ *procName*\ *stepName*.asp	DateHandler.asp, FormHandler.asp, webclient.asp	The step definition file for a particular step, as generated by the Step Definition Converter.
SWWebClientDir\ *nodeName*\ *procName*\ *stepName*Close.asp	standard.asp, webclient.asp	The persist file (which handles the keep and release functions) for a particular step, as generated by the Step Definition Converter.

Core Scripting Functions

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

- Server-side scripting functions are provided in the file *SWWebClientDir***standard.asp**.
- Client-side functions are provided in the file *SWWebClientDir***webclient.asp**.

Integrating the TIBCO iProcess Client (ASP) with Other Applications

The iProcess Client (ASP) can be integrated with other technologies, either by using its server-based ASP pages to access other COM objects, 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 Step Definition Template (**\Winnt\System32\Template.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 definition is being accessed.
- The iProcess Client (ASP) stores all fields used within a step definition 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 iProcess Client (ASP) 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.asp** file.

Controlling Step Flow

If you are modifying code within the Step Definition Template (**\Winnt\System32\Template.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.asp** 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 TIBCO iProcess Modeler). Any HTML tagged text will be preserved intact when the ASP form is generated by the Step Definition Converter.

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

Back to Library

Back to Library

Appendix A Form Converter Mappings

The following tables show how iProcess form elements are mapped to HTML elements and attributes when the form is converted to an ASP page.

Back to Library

General Mappings

iProcess form 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 iProcess 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)	Hidden fields are generated like optional fields but are not displayed.
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 attachment are not supported by TIBCO iProcess Objects. However, the same HTML code is generated as for normal text fields.

Back to Library

iProcess form element	HTML elements and attributes
Field markings (validations)	Validations can have the following class attributes: • ‘class=REQ’ • ‘class=OPT’ • ‘class=DIS’
Field markings (drop down lists)	Drop down lists can have the following class attributes: • ‘class=REQ’ • ‘class=OPT’ • ‘class=DIS’
Field markings (help)	A ‘title=…’ attribute (Internet Explorer specific) is used to specify pop-up help for the field.
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 colours	Font colours are supported using the ‘color=…’ attribute. No named colours 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