

**Sterling Integrator<sup>®</sup>**

---

**Web Extensions**

**Version 5.0**

**Sterling Commerce**  
An IBM Company

© Copyright 2009 Sterling Commerce, Inc. All rights reserved.  
Additional copyright information is located on the Sterling Integrator Documentation Library:  
<http://www.sterlingcommerce.com/Documentation/SI50/homepage.htm>

---

# Contents

Features of Web Extensions . . . . .	5
Obtain a License for Web Extensions . . . . .	5
Human Interaction Services . . . . .	5
Role-Based Security . . . . .	6
Use Tag Libraries with Web Extensions . . . . .	7
JSP Tag Libraries . . . . .	7
Before You Can Use JSP Tag Libraries . . . . .	7
Run Business Process Tag Libraries . . . . .	7
runBP . . . . .	7
runBPnoReturnDoc . . . . .	8
getDocuments . . . . .	8
Business Process Resolver . . . . .	9
Authorization (autho) Tag Library . . . . .	10
Session Validation (validSession) Tag Library . . . . .	11
Authentication (authenticate) Tag Library . . . . .	11
Requirements For Using the JSTL . . . . .	12
JSTL Concepts and Definitions. . . . .	12
Set Up and Use JSTL . . . . .	13
Add the JSTL to the Web Application . . . . .	14
Edit the web.xml File for JSTL . . . . .	14
Add the JSTL to a JSP Web Template . . . . .	15
Delete JSTL Code from a JSP Web Template . . . . .	16
Update, Compile, and Redeploy the WAR file in Gentran Integration Suite . . . . .	16
Deploy an External JSP Web Application on Sterling Integrator. . . . .	17
Troubleshoot JSTL Issues . . . . .	17
JSTL Demonstrations . . . . .	17
Manage Web Resources . . . . .	19
About Resources. . . . .	19
Web Resources . . . . .	19
Resource Tags . . . . .	19
Web Archive (.war) Files . . . . .	19
Check In a Web Resource . . . . .	20

Check Out a Web Resource . . . . .	20
Search for a Web Resource . . . . .	21
Delete a Web Resource . . . . .	22
Maintain a Web Resource Version History . . . . .	23
Manage .war Files . . . . .	23
Generate a .war File . . . . .	23
Deploy a Web Application in a DMZ . . . . .	24
Deploy webx.war on Apache/Tomcat 3.2 . . . . .	24
Deploy a Web Application in a DMZ Using Perimeter Servers . . . . .	25
Edit the HTTP Server Adapter . . . . .	26
Test the Web Application . . . . .	26
<b>Index</b>	<b>27</b>

---

---

## Features of Web Extensions

*Web Extensions* is a suite of utilities and services that enable human interaction within business processes when the business processes run in Sterling Integrator. With Web Extensions, you can exchange business documents with your smaller trading partners. Following are the features of Web Extensions:

- ◆ Services enabling the addition of human interaction to virtually any business process
- ◆ Support for Java Server Pages (JSP) when using perimeter services
- ◆ JSP Tag libraries
- ◆ Importing and Exporting of Web resources
- ◆ JSP-based WAR support
- ◆ Support for localization, international standards, and double-byte character sets (DBCS)
- ◆ E-mail notification of business processes waiting for human interaction
- ◆ Role-based security features to protect against unauthorized access

## Obtain a License for Web Extensions

To use Sterling Integrator Web Extensions, you must obtain a license. Obtain a license from the Sterling Commerce Self-Service Web site at [www.productupdates.stercomm.com](http://www.productupdates.stercomm.com).

## Human Interaction Services

To enable human interaction with a business process, the Web Extensions suite uses Human Interaction services. *Human interaction* is any action taken by a person on a business process when it runs.

An example of human interaction is when a business process contains a purchase order that requires a manager's approval. The business process runs until it comes to the service requiring the manager's approval, where the business process is placed in a waiting status. The manager either receives an e-mail notification of a pending document with a link to the document or simply logs in to a Web site that shows the documents requiring approval. The manager opens the document, views it through the Web browser, approves it by making any comments that are needed, and submits the document using the **Submit** button. After it is submitted, the approved purchase order continues to the next service in the business process.

The types of Human Interaction services are:

- ◆ Human Interaction Event service – Identifies business process data that requires human interaction, flags business processes that require some human action, and creates references to the business process data to await human action.
- ◆ Human Interaction Query service – Searches the storage area containing the business processes flagged by the Human Interaction Event service for human interaction and retrieves the business process data.
- ◆ Human Interaction Document Loader service – Receives a MinedDataId as input and uses this ID to determine which document to load into the business process.

---

## Role-Based Security

*Role-based security* provides access to specific files, business processes, services, and product features, according to the permissions associated with the user account.

A *user account* contains the groups the user belongs to and the permissions associated with each group or user. *Permissions* provide access to the different modules within Sterling Integrator and are the foundation of role-based security.

The user account is linked to your Sterling Integrator user name and password. Each time you log in, Sterling Integrator verifies whether you are a valid user and grants access to only those areas you are allowed to use based on the permissions assigned to you by the Sterling Integrator system administrator.

Role-based security also helps direct messages and documents to the appropriate user within Sterling Integrator Web Extensions. In this circumstance, the Web Extensions Human Interaction Event service pulls the approving authority's name and e-mail address from the Sterling Integrator database and routes the document to that person. This routing feature helps you manage message queues.

---

# Use Tag Libraries with Web Extensions

## JSP Tag Libraries

*JavaServer Pages (JSP)* are Web pages that use templates, custom elements, scripting languages, and server-side Java objects to return dynamic content to a Web browser in the form of HTML or XML. You can use the JSP Web templates in Sterling Integrator Web Extensions. A *JSP Web template* is a Web form that is used to interface with a Web application, such as a smaller business partner completing a purchase order and invoicing through a Web site.

A *Java Server Pages (JSP) tag library* is a collection of predefined JSP components that you can use in your JSP Web templates. Use JSP tag libraries to:

- ◆ Reduce manual coding efforts of your JSP Web templates
- ◆ Reduce the amount of Java code in your JSP Web templates

## Before You Can Use JSP Tag Libraries

Before you can use the JSP tag libraries with Sterling Integrator, you must:

- ◆ Have knowledge of Java and Java Server Pages.
- ◆ Have knowledge of application servers.
- ◆ Have read, write, and execute rights to your application server and the server where Sterling Integrator is installed.
- ◆ Verify that *install\_dir*/jar/woodstock.jar is on the classpath accessible to a Web server.
- ◆ Copy *install\_dir*/installed\_data/pshttp/exploded\_wars/helloworld/WEB-INF/tld/userautho.tld to the WEB-INF/tld directory in your Web application.
- ◆ Specify the location of your JSP tag library in your Web application's WEB-INF/web.xml file. Type the following code between the <web-app> and </web-app> tags:

```
<taglib>
<taglib-uri>http://www.stercomm.com/uix/security</taglib-uri>
<taglib-location>tld/userautho.tld</taglib-location>
</taglib>
```

## Run Business Process Tag Libraries

Use the tag libraries to run a single business process per JSP page.

### runBP

The runBP tag library runs a business process, which is specified in the bpname attribute. The runBP tag library also sets the pageContext variable called bpResults in the org.w3c.dom.Document type. The business process results are displayed using the bpResults variable.

The following table describes the attributes the runBP tag library uses:

<b>Attribute</b>	<b>Description</b>
user	User ID. Required. This attribute may be specified at the business process run time.
nvp	Name value pairs to be placed into process data in the format name=value&name2=value2. Required. This attribute may be specified at the business process run time.
bpname	Business process to run. Required. This attribute may be specified at the business process run time.
pridoc	Primary document that invokes the business process. Required. This attribute may be specified at the business process run time.

### runBPnoReturnDoc

The runBPnoReturnDoc tag library runs a business process with no document being returned from the running business process.

The following table describes the attributes the runBPnoReturnDoc tag library uses:

<b>Attribute</b>	<b>Description</b>
user	User ID. Required. This attribute may be specified at the business process run time.
nvp	Name value pairs to be placed into process data in the format name=value&name2=value2. Required. This attribute may be specified at the business process run time.
bpname	Business process to run. Required. This attribute may be specified at the business process run time.
pridoc	Primary document that invokes the business process. Required. This attribute may be specified at the business process run time.

### getDocuments

The getDocuments tag library runs a business process and extracts a document from the business process.

The following table describes the attributes the getDocuments tag library uses:

<b>Attribute</b>	<b>Description</b>
user	User ID. Required. This attribute may be specified at the business process run time.



Attribute	Description
nvp	Name value pairs to be placed into process data in the format name=value&name2=value2. Required. This attribute may be specified at the business process run time.
bpname	Business process to run. Required. This attribute may be specified at the business process run time.
pridoc	Primary document that invokes the business process. Required. This attribute may be specified at the business process run time.

## Business Process Resolver

Use the Business Process Resolver to run multiple business processes on a single JSP page or if you have complex conditional logic. The JSP compiler cannot compile the jsp when using the runBP Tag Library.

### Required Imports

```
<%@page import="org.w3c.dom.Document"%>
<%@page import="com.sterlingcommerce.woodstock.util.xml.XMLUtil"%>
```

### Method Interfaces

```
public org.w3c.dom.Document resolve(String bpName, String processData,
byte[]
    primaryDocument, String username) throws Exception
public org.w3c.dom.Document resolve(String bpName, String processData,
byte[]
    primaryDocument, String username, boolean hasAttachment) throws Exception
public org.w3c.dom.Document resolve(String bpName, String processData,
    Document uploaded_doc, String username, boolean hasAttachment) throws
Exception
public org.w3c.dom.Document resolve(String bpName, String processData,
String
    primaryDocument, String username) throws Exception
public org.w3c.dom.Document resolve(String bpName, String processData,
String
    primaryDocument, String username, boolean hasAttachment) throws Exception
public byte[] resolve(String bpName, String processData, String username)
throws
    Exception
```

### Example

```
<%
Document doc;
BusinessProcessResolver bpResolver = new BusinessProcessResolver();
doc = (org.w3c.dom.Document)
bpResolver.resolve("myBPname", "test=123&test2=234", "", "GIS_user_to_run_as");
strBPResultsData = XMLUtil.documentToString(doc, false);
%>
```

com.sterlingcommerce.woodstock.util.XMLUtil

This utility class is used in the previous code example. It provides several interfaces for managing xml nodes, which is what most of the BusinessProcessResolver class interfaces return, in the form of a org.w3c.dom.Document object.

#### Method Interfaces

```
public static String documentToHTML(Node node, boolean canonical)
public static String documentToString(Node node, boolean canonical)
public static String documentToString(Node node, boolean canonical, boolean
    useReturns)
public static String documentToString(Node node, boolean canonical, boolean
    useReturns)
public static String documentToString(Node node, boolean canonical,
    StringBuffer sb,
    String indent)
public static String documentToString(Node node, boolean canonical,
    StringBuffer sb,
    String indent, String newline)
public static String documentToString(Node node, boolean canonical,
    StringBuffer sb,
    String indent, String newline, boolean useReturns)
```

## Authorization (autho) Tag Library

The autho tag library verifies you, as a Sterling Integrator user, belong to specified Sterling Integrator groups or have specified Sterling Integrator permissions. If the authorization fails, the JSP body is not displayed or you are forwarded to an error page. If neither permissions or groups attributes are set, you are considered authorized.

Since authorization is based on at least one match against a permission or group, authorization succeeds. This means that if a user does not have association with any group, but has one permission applied, the authorization succeeds.

The following table describes the attributes the autho tag library uses:

Attribute	Description
user	User ID. Optional. This attribute may be specified at the business process run time. The tag checks the session for a user name attribute if the attribute is not declared.
group	Group ID. Optional. This attribute may be specified at the business process run time.
groups	List or ArrayList of group IDs. Both the group and the groups attributes must be declared at the same time. Authorization succeeds if you have association with at least one group.
permission	Permission ID. Optional. This attribute may be specified at the business process run time.

Attribute	Description
permissions	List or ArrayList of permissions. Both the permission and the permissions attributes must be declared at the same time. Authorization succeeds if you have association with at least one permission.
login	URL of the login page that you are sent to if your session expires or is invalid. A session is invalid if a user name attribute is not present. Optional. This attribute may be specified at the business process run time. If this attribute is not set and the session expires, the JSP page does not display. The session is valid if the user name attribute is present.
checkExistence	Checks for a group or permission association prior to authorization. Optional. This attribute may be specified at the business process run time. If a permission or group association does not exist and this value is set to true, all non-existent permissions and groups are skipped in the authorization process.

## Session Validation (validSession) Tag Library

The validSession tag library validates your session with Sterling Integrator has not expired. If the validation fails, an error page or login page displays.

The following table describes the attributes the validSession tag library uses:

Attribute	Description
login	URL of the login page that you are sent to if your session expires or is invalid. A session is invalid if a user name attribute is not present. Required. This attribute may be specified at the business process run time. If this attribute is not set and the session expires, the JSP page does not display. The session is valid if the user name attribute is present.

## Authentication (authenticate) Tag Library

The authenticate tag library Authenticates you against the Sterling Integrator database or your LDAP server to determine that you are authorized access to the system.

The following table describes the attributes the authenticate tag library uses:

Attribute	Description
user	User ID you enter in the login page. Required. This attribute may be specified at the business process run time.
pass	Password you enter in the login page. Required. This attribute may be specified at the business process run time.

Attribute	Description
login	<p>URL of the login page that you are sent to if your session expires or is invalid. A session is invalid if a user name attribute is not present. Required.</p> <p>This attribute may be specified at the business process run time. If this attribute is not set and the session expires, the JSP page does not display. The session is valid if the user name attribute is present.</p>

A *Java Server Page Standard Tag Library (JSTL)* is a consolidated grouping of common JavaServer Pages (JSP) code samples that you can use in developing JSP Web templates. JSTL is open source and provides code for common tasks, such as look ups, parsing, date, and time formats. By using the JSTL Standard Tag Library, you can standardize common functions.

The JSTL Standard Tag Library enables you to:

- ◆ Reduce the complexities of developing JSP Web templates by creating standardized code in Web templates to improve consistency of presentation and handling of data
- ◆ Reduce inefficiencies in your business by standardizing code by reducing errors in Web template development
- ◆ Reduce costs and improve revenue generation by speeding the Web template development process
- ◆ Streamline processing of XML messages and XSLT transformations

**Note:** JSTL is open source. These instructions include how to implement JSTL in a Gentran Integration Suite Web application. Instructions for using JSTL can be found on the open source web site.

## Requirements For Using the JSTL

- ◆ JSP 1.2 container – Sterling Integrator uses Jetty, a Java-based HTTP Server and Servlet container that processes a JSP for rendering.
- ◆ Installed copy of Sterling Integrator with Web Extensions.
- ◆ Web application created using J2EE.

## JSTL Concepts and Definitions

Concept	Definition
Expression Language (EL)	Computer language used in JSTL for expressing simple expressions, which is based on the XPath and JavaScript or JScript languages.
JavaServer Page (JSP)	Web pages that use templates, custom elements, scripting languages, and server-side Java objects to return dynamic content to a Web browser in the form of HTML or XML.
JSP Container	An entity that provides life-cycle management and runtime services to the components of a Web application, just like a servlet, and provides an engine that interprets and processes JSPs.
JSTL Namespace	A unique identifier that defines the URI of the tag library and the prefix used in the tags for that specific functional area library.

Concept	Definition
Prefix	A unique identifier that defines the functional area that a tag belongs to. The prefix is defined in the JSTL Namespace. For example, prefix="c" means that all tags that start with <c: are part of the core functional area tag library. The c in <c: is the prefix of the tag.
Runtime (RT) Processing	Scripting used in JSPs processed at runtime. This method is being replaced by the EL method in JSTLs.
Tag Handler	Explanation of how to process the tag when it is encountered in the processing of the Web template. Typically, this is defined in a <tag-class/> tag.
Tag Library Descriptor (.tld)	XML file that describes a tag library and contains the tags in the library. Each functional area library has a .tld file.
Universal Resource Indicator (URI)	Trailing part of a Web address that indicates the location of the resource being used. For example, in http://host:port/uri/sample.xml, the URI is located on host and sample.xml is located in the uri folder.
web.xml	A standard file created when the Web application is developed in J2EE and contains properties controlling servlets for the specific Web application. The web.xml file also includes the tag library declaration, the URI, and tld declarations.
Web Application	Web site that delivers dynamic content. For example, a Web site that has a purchase order form that once submitted returns an acknowledgment that the purchase order was created and sent to the appropriate trading partner.
Web Archive (WAR)	A compressed file that contains the directory structure and file contents of a Web application. The directory structure includes the WEB-INF directory containing the web.xml file and classes directory.

## Set Up and Use JSTL

You do not need to configure Sterling Integrator to use the JSTL. The JSTL is optional and since this is a library of code, it is never directly tied to any part of Sterling Integrator. The library is only referenced by your JSP Web templates.

The following checklist describes what you must do to implement JSTL:

Step	Description	Your Notes
1	Add the JSTL to the Web Application.	
2	Edit the web.xml file for the Web application to include the tag libraries. See Edit the web.xml File for JSTL.	
3	Add the JSTL Namespace and tags to the JSP Web template.	
4	Update, Compile, and Redeploy the WAR file in Gentran Integration Suite <b>Note:</b> You can use the jar utility that is installed with the JDK to compile the WAR file.	
5	Test the Web template in the Web application.	

## Add the JSTL to the Web Application

To add the JSTL library to your Web application, you must create a .TLD Directory and copy the .tld directory files into it.

1. In your *WebApplicationName*/WEB-INF directory, create a new directory named TLD.
2. Unjar the standard.jar file in the *install\_dir*/installeddata/patch/jars/jakarta\_jstl/1\_1\_2/ directory.
3. From the expanded standard.jar file, copy the .tld files to the *WebApplicationName*/WEB-INF/TLD directory you created.
4. Edit the web.xml file properties for your specific Web application. See *Edit the web.xml File for JSTL* on page 14.

## Edit the web.xml File for JSTL

The web.xml file contains properties controlling servlets for a specific Web application. This is a standard file created when the Web application is created using J2EE. If you are adding JSTL Standard Library functions to multiple Web applications, you must edit the web.xml file associated with each Web application.

To use JSTL, you must declare the <taglib> tags in the web.xml file between the <Web-app> and </Web-app> tags.

To edit the web.xml file:

1. Locate the web.xml file in the Web application directory. The standard directory structure is:  
*WebApplicationName*/WebApp/WEB-INF/web.xml
2. Declare the <taglib> tags for each of the following tag libraries between the <Web-app> and </Web-app> tags:

For This Tag Library	Type The Following Tag
Core (EL)	<pre>&lt;taglib&gt; &lt;taglib-uri&gt;http://java.sun.com/jstl/core&lt;taglib-uri&gt; &lt;taglib-location&gt;tld/c-1_0.tld&lt;/taglib-location&gt; &lt;/taglib&gt;</pre>
Core (RT)	<pre>&lt;taglib&gt; &lt;taglib-uri&gt;http://java.sun.com/jstl/core-rt&lt;taglib-uri&gt; &lt;taglib-location&gt;tld/c-1_0-rt.tld&lt;/taglib-location&gt; &lt;/taglib&gt;</pre>
Format (EL)	<pre>&lt;taglib&gt; &lt;taglib-uri&gt;http://java.sun.com/jstl/fmt&lt;taglib-uri&gt; &lt;taglib-location&gt;tld/fmt-1_0.tld&lt;/taglib-location&gt; &lt;/taglib&gt;</pre>

For This Tag Library	Type The Following Tag
Format (RT)	<pre>&lt;taglib&gt; &lt;taglib-uri&gt;http://java.sun.com/jstl/fmt-rt&lt;taglib-uri&gt; &lt;taglib-location&gt;tld/fmt-1_0-rt.tld&lt;/taglib-location&gt; &lt;/taglib&gt;</pre>
SQL (EL)	<pre>&lt;taglib&gt; &lt;taglib-uri&gt;http://java.sun.com/jstl/sql&lt;taglib-uri&gt; &lt;taglib-location&gt;tld/sql-1_0.tld&lt;/taglib-location&gt; &lt;/taglib&gt;</pre>
SQL (RT)	<pre>&lt;taglib&gt; &lt;taglib-uri&gt;http://java.sun.com/jstl/sql-rt&lt;taglib-uri&gt; &lt;taglib-location&gt;tld/sql-1_0-rt.tld&lt;/taglib-location&gt; &lt;/taglib&gt;</pre>
XML (EL)	<pre>&lt;taglib&gt; &lt;taglib-uri&gt;http://java.sun.com/jstl/xml&lt;taglib-uri&gt; &lt;taglib-location&gt;tld/x-1_0.tld&lt;/taglib-location&gt; &lt;/taglib&gt;</pre>
XML (RT)	<pre>&lt;taglib&gt; &lt;taglib-uri&gt;http://java.sun.com/jstl/xml-rt&lt;taglib-uri&gt; &lt;taglib-location&gt;tld/x-1_0-rt.tld&lt;/taglib-location&gt; &lt;/taglib&gt;</pre>

3. Save the web.xml file to the original location.

**Caution:** Do not change the name or the location of the web.xml when saving the file. Doing so may cause errors.

## Add the JSTL to a JSP Web Template

You must add JSTL code to a new JSP Web template or edit an exiting JSP Web template to use the JSTL features in your Web application.

1. Open the JSP Web template in an editor.

2. In the line preceding the <HTML> tag, for each functional library that are adding to the JSP, type the Namespace for the functional area library you are using in the JSP Web template:

```
<%a taglib uri="http://java.sun.com/jstl/core" prefix="c" %>
<%a taglib uri="http://java.sun.com/jstl/fmt" prefix="fmt" %>
<%a taglib uri="http://java.sun.com/jstl/xml" prefix="x" %>
<%a taglib uri="http://java.sun.com/jstl/sql" prefix="sql" %>
<%a taglib uri="http://java.sun.com/jstl/core-rt" prefix="c_rt" %>
<%a taglib uri="http://java.sun.com/jstl/fmt-rt" prefix="fmt_rt" %>
<%a taglib uri="http://java.sun.com/jstl/xml-rt" prefix="x_rt" %>
<%a taglib uri="http://java.sun.com/jstl/sql-rt" prefix="sql_rt" %>
```

3. In the appropriate location of the JSP Web template, between the <body> and </body> tags, type the tag code. For example:

```
<fmt:formatDate type="both" timestyle="short" value=${now}" />
</fmt:formatDate>
```

4. Save the JSP.

After you create or edit the JSP Web template, you can check your JSP Web template in to the Sterling Integrator database. This is optional, unless you intend to use your JSP Web template in Sterling Integrator Web Extensions. In this case, you must check in your JSP Web template, so that it can be processed.

## Delete JSTL Code from a JSP Web Template

1. Open the JSP Web template in an editor.
2. In the appropriate location of the JSP Web template, between the <body> and </body> tags, delete the complete JSTL tag code.

For example, if the tag you are deleting is <c:set>, ensure that you delete the </c:set> tag and all attributes associated with the tag.

3. Save the JSP Web template.

## Update, Compile, and Redeploy the WAR file in Gentran Integration Suite

To deploy the Web application in Sterling Integrator, you must update and compile the Web application WAR file, which decompresses during deployment.

1. On the Sterling Integrator computer, change directories to the *install\_dir/container/applications* directory.
2. In the Web application root directory, verify that the following directory structure is followed:
  - ◆ web.xml (Required)
  - ◆ TLD – Directory containing the Tag Library Descriptors (Required)
  - ◆ classes – Directory containing the server-side classes, utility classes, and JavaBeans components (Required)



- ◆ lib – Directory containing JAR archive of libraries (tag libraries and utility libraries) (Required)
  - ◆ Any miscellaneous subdirectories (Required)
3. Change directories to the *install\_dir/bin* directory.
  4. Run the *Deployer.sh* script to deploy all WAR files.

## Deploy an External JSP Web Application on Sterling Integrator

After you create the JSP Web application WAR file outside of Sterling Integrator, you can deploy it to your server that your application server is installed on, so you can use the Web templates you deployed. If you are deploying your Web application in Sterling Integrator, complete this procedure:

1. Copy the WAR file to *install\_dir/noapp/deploy* directory.
2. Stop Sterling Integrator.
3. Start Sterling Integrator.

The WAR file is deployed in Sterling Integrator, and you can access the Web template by typing <http://localhost:port/webappname>.

**Note:** If you make changes to the Web application outside of Sterling Integrator, you must repackage the Web application as a WAR file and redeploy the Web application in Sterling Integrator. You can use the *jar* utility provided with your JDK to recompile the WAR file.

## Troubleshoot JSTL Issues

Follow this methodology when troubleshooting JSTL issues:

Issue	Action
General Errors	<ul style="list-style-type: none"> <li>◆ Verify that you defined the JSTL Namespace in the JSP Web template.</li> <li>◆ Verify that you added the complete JSTL tag to the JSP Web template.</li> <li>◆ Troubleshoot your JSP code.</li> </ul>
Class Not Found Errors	<ul style="list-style-type: none"> <li>◆ Verify that you included the JSTL libraries to the Web application.</li> <li>◆ Troubleshoot your JSP code.</li> </ul>
Page Cannot Display Error	<ul style="list-style-type: none"> <li>◆ Verify that the prefix defined in JSTL Namespace and tag match and are the same as the <code>&lt;short-name&gt;</code> in the <code>.tld</code> file for that functional area.</li> </ul>

## JSTL Demonstrations

Sterling Integrator contains three demonstrations of simple JSTL functionality located at <http://host:port/helloworld>. Once you go to the *helloworld* site, you will be prompted for your username and password.

**Note:** Other demonstrations are available from the helloworld site. Only those listed in the table below are applicable to JSTL.

Demonstration	Description	Steps
JSTL Counting Demonstration	Demonstrates the core functional area features that enable counting in a Web template by letting you type a number to count to and returning the counted sequence.	<ol style="list-style-type: none"> <li data-bbox="1027 365 1321 478">1. In the Type a number to count text box, type the number you want the computer to count to.</li> <li data-bbox="1027 489 1417 611">2. Click <b>Go!</b> The list of numbers up to the number you typed in the text box displays in sequence.</li> </ol>
JSTL XML Demonstration	Demonstrates the XML functional area features of parsing a document and reformatting the text from the document in a Web browser.	Click <b>Go!</b> next to the list where you selected JSTL XML Demo. The parsed document displays in the JSP Web template.
JSTL Timezone Demonstration	<p data-bbox="613 774 1000 888">Demonstrates the formatting functional area Format Timezone feature by letting you select a city to display the time and date of.</p> <p data-bbox="613 898 1000 1098"><b>Note:</b> The demonstration does not adjust for Daylight Savings Time. Therefore, the time may be off by one hour if Daylight Savings Time is in effect for the selected city.</p>	<p data-bbox="1027 774 1321 798">Perform the following steps:</p> <ol style="list-style-type: none"> <li data-bbox="1027 808 1370 894">1. From the list, select a city that you want to display the time for.</li> <li data-bbox="1027 905 1417 1026">2. Click <b>Go!</b> The time and date for the city you selected displays in the updated JSP Web template.</li> </ol>

---

## Manage Web Resources

Sterling Integrator provides tools to manage Web resources that can be used in building Web applications. After you have checked in the Web resources, associate each resource with a resource tag, and add them to the Web archive (.war) files to easily deploy the Web application in another environment.

### About Resources

*Resources* are files, templates, and documents that are deployed in Sterling Integrator to perform various actions. Resource types in Sterling Integrator include:

- ◆ Trading partner data
- ◆ Business processes
- ◆ XML schemas
- ◆ XSLT style sheets
- ◆ Service configurations
- ◆ Maps
- ◆ Web resources

### Web Resources

A *Web resource* is a resource that Sterling Integrator uses to display and run Web applications and functions in Web browsers. Web resources can be a Java server page file, an HTML file, an image file, an .xml file, a cascading style sheet, or any other file that is used in displaying or running Web applications and functions.

### Resource Tags

A *resource tag* is a name that associates different resources and Web resources, so they can be assigned to a .war file.

### Web Archive (.war) Files

A *Web archive (.war) file* is a compressed file that contains the directory structure and file contents of a Web application. In this definition, file contents refers to Web resources. A .war file enables easy transport and deployment of the Web resources to another environment.

To create a .war file in Sterling Integrator, complete the following steps:

1. Check in the Web resources you want to include in the .war file.  
For more information, see *Checking In a Web Resource*.
2. Create the resource tag associating the Web resources with the resource tag.
3. Generate the .war file for the resource tag.  
For more information, see *Generating a Web Archive (.war) File*.

## Check In a Web Resource

A Web resource can come from many different sources, but before it can be used by Sterling Integrator, check the Web resource into Sterling Integrator. After you check in a Web resource, search for it, and check it out of Sterling Integrator, and manage the versions of the Web resource.

To check in a Web resource:

1. From the Sterling Integrator **Deployment** menu, select **Web Extensions > Web Resources**. The Web Resources page opens.
2. Next to the Check-in new Web Resource File, click **Go!**
3. In the **Web Resource File Name** field, complete one of the following actions:
  - ◆ Type the file name and extension of the Web resource to check in.
  - ◆ Click **Browse** to navigate to the location where the Web resource resides, and select the file name of the Web resource.
4. In the **Description** field, type any comments you want to include (required) and click **Next**. The Naming page opens.

Sterling Integrator uses the comments you type in this field to establish a version for each resource. Ensure that the comments are descriptive to help you maintain version control. For example, if you are checking in a Web resource for which no previous version exists, you could type Version 1 as the description. When you check in subsequent versions of this Web resource, you can type Version 2 and so on.

5. In the **Resource Name** field, type a name that will identify the Web resource in Sterling Integrator.
 

If a the name of the Web resource you are checking in already exists in the Sterling Integrator database, you are prompted to type a new name for the Web resource.
6. Under **Resource Type**, select the type of Web resource that you are checking in and click **Next**.
 

Sterling Integrator provides the initial type choice based on the file extension of the Web resource you are checking in. You can change the type by selecting a different type from the list. Only one resource type can be associated with a resource name.
7. In the Confirm page, complete the following actions:
  - ◆ Select **Enable** to enable the Web resource for use in Sterling Integrator.
  - ◆ Review the information and click **Finish**.

## Check Out a Web Resource

After a Web resource is checked in, you can check it out of the Sterling Integrator database. Checking out a Web resource does not remove it from the Sterling Integrator database; it saves a copy of the resource from the host to your client computer. You can choose to check out a read-only copy of the Web resource, or lock the Web resource in the Sterling Integrator database for editing.

To check out a Web resource:

1. From the Sterling Integrator **Deployment** menu, select **Web Extensions > Web Resources**.

2. In the Web Resources page, locate the resource you want to check out by using either the Search or List option.

For more information, see *Searching for a Web Resource* on page 92.

3. Click **source manager** next to the Web resource you want to check out.
4. In the Web Resource Source Manager page, click **check-out** next to the Web resource you want to check out.
5. In the message about locking the file, complete one of the following actions:
  - ◆ To lock the Web resource file for editing, click **OK**.
  - ◆ To check out a read-only copy of the Web resource file, click **Cancel**. You cannot edit a read-only copy of the Web resource file.
6. In the **File Download** dialog box, select the **Save** option.
7. In the **Save As** dialog box, select the location where you want to save the Web resources and click **Save**. The copy of the Web resource is saved to the new location.
8. In the Download complete dialog box, complete one of the following actions:
  - a. To open the Web resource, click **Open**.
  - b. To open the folder where you saved the Web resource, click **Open Folder**.

## Search for a Web Resource

After a Web resource is checked in, you can search for that resource.

To search for a Web resource:

1. From the **Deployment** menu, select **Web Extensions > Web Resources**.
2. In the Web Resources page, complete one of the following actions:
  - ◆ Under Search in the **Resource Name** field, type either a portion of the name or the entire name of the Web resource you are searching for, and click **Go!** The Web Resource Management page opens, listing all the Web resources containing the full or partial name you typed.
  - ◆ Under **List in the Alphabetically** field, select **ALL** or the letter that begins with the name of the Web resource you are searching for. Selecting **ALL** lists all of the Web resources that are checked in. Click **Go!** The Web Resource Management page opens, listing all of the Web resources that match your search criteria.
  - ◆ Under List in the **by Resource Type** field, select the type of the Web Resource that you are searching for. You can select only one type from the list at a time. The following table describes the Web resource types:

Web Resource Type	Description
JSP	Java Server Page with the file extension .jsp.
JAVASCRIPT	Java scripts with the file extension .js.
HTML	HTML documents with the file extension .htm or .HTML.

Web Resource Type	Description
XML	XML documents with the file extension .xml.
STYLESHEET	Cascading style sheets with the file extension .css.
PROPERTIES	Property files, such as XML Resource bundles, with the file extension .xrb, or property files with the file extension .xml.
IMAGE	Graphic files used in Web pages with the file extensions .gif, .jpg, or .jpeg.
OTHER	All other files that do not fit any of the types listed in this table. File extensions can include .doc for a document file, .txt for text file, .bp for a business process and so forth. Almost any file extension can be included in this Web resource type.

- Depending on your task, complete one of the following actions:
  - Go to Source Manager to check in or check out the Web resource. For more information, see *Checking In a Web Resource* and *Checking Out a Web Resource*.
  - Go to Version Manager to set a default version of the Web resource and to enable or disable Web resources for business process use. For more information, see *Maintaining a Web Resource Version History*.

## Delete a Web Resource

After a Web resource is checked in, you can delete it from the Sterling Integrator database. Deleting a Web resource removes it completely from the Sterling Integrator database. Before you delete a Web resource, check it out of Sterling Integrator and save a copy in another location for future reference.

To delete a Web resource:

- From the Sterling Integrator **Deployment** menu, select **Web Extensions > Web Resources**.
- In the Web Resources page, locate the resource you want to delete by using either the Search or the List option.  
For more information, see *Searching for a Web Resource*.
- Click **source manager** next to the Web resource you want to delete.
- In the Web Resource Source Manager page, click the check box in the Delete column next to the Web resource you are deleting from Sterling Integrator.
- Next to Delete Selected Versions, click **Go!**
- You are prompted to confirm the deletion. Click **OK**.
- In the Delete Resources page, review the information about the Web resource and click **Next**.
- In the Confirm page, a message displays stating *Deleting this resource may cause processes to fail. This action cannot be reversed*. Click **Delete** to delete the Web resource.

The Web resource is deleted from the database.

## Maintain a Web Resource Version History

Sterling Integrator enables you to maintain versions of your Web resources. A *version* is an edited copy of a Web resource. Each time a Web resource is checked out and then checked in, a new version is created, but the original Web resource remains intact for future use.

A Web resource version is designated by the description in the Comments field when you check in a Web resource. Because this description provides the basis for maintaining the version for that Web resource, make the description unique and meaningful to the use of the resource. Doing this makes it easier to quickly choose the resource you need. In addition, Sterling Integrator adds a date and time stamp to each version that can be used for version identification.

To maintain Web resource version:

1. From the Sterling Integrator **Deployment** menu, select **Web Extensions > Web Resources**.
2. In the Web Resources page, locate the resource you want to review by using either the Search or the List option.  
For more information, see *Searching for a Web Resource* on page 92.
3. Next to the Web resource you want to review, click **version manager**.
4. In the Web Resource Version Manager page, click **Default** next to the Web resource version you want to set as the default resource. This is the version of the resource to be used for business processes using this resource.
5. Do you want to enable the Web resource to make it available for use by business processes?
  - ◆ If Yes, ensure that the check box next to the Web resource you want to enable is selected.
  - ◆ If No, ensure that the check box next to the Web resource you want to disable is cleared.
6. Click **Save**. Your changes are applied to the Web resources in the Sterling Integrator database.

---

## Manage .war Files

After you have checked in the Web resources and associated each resource with a resource tag, you can generate a Web archive (.war) file to deploy the resources in another environment. Generating a .war file enables you to:

- ◆ Deploy Web applications easily in a test environment or a production environment
- ◆ Deploy Web applications easily in a DMZ

## Generate a .war File

To generate a .war file:

1. From the **Deployment** menu, select **Web Extensions > Utilities**.
2. In the Web Utilities page, under Export Web Resources, click **Go!** next to Generate a (.war) file using system resources.

3. In the **WAR File Name** field, type a file name for the .war file, including the .war file extension.
4. In the **Resource Tag** field, select the resource tag from the list that contains the resources you want to include in the .war file and click **Next**.
5. Do you want to save the .war file on the server?
  - ◆ If Yes, ensure that the check box next to Save generated (.war) file on server is selected.
  - ◆ If No, ensure that the check box next to Save generated (.war) file on server is cleared.
6. In the Confirm page, verify the .war file information and click **Finish**.
7. In the page showing that the system update completed successfully, complete the following actions:
  - a. Click the document icon next to View Export Report to verify whether the .war file generated without errors.
  - b. Click **Download** next to Download (.war) file to download to your local computer.
8. In the **File Download** dialog box, complete one of the following actions:
  - ◆ To open and view the .war file, click **Open**.
  - ◆ To save the .war file to your local computer, click **Save**.
  - ◆ To cancel the download, click **Cancel**.
  - ◆ For more information about downloading files, click **More Info**.
9. If you selected Save, make changes to the file name, if any, and browse to the location you want to save the .war file to and click **Save**.
10. In the Download Complete dialog box, complete one of the following actions:
  - ◆ To open the file you downloaded, click **Open**.
  - ◆ To open the folder that you downloaded the file in, click **Open Folder**.
  - ◆ To close the Download Complete dialog box, click **Close**.

You are now ready to deploy the downloaded .war file in the new environment.

---

## Deploy a Web Application in a DMZ

You can deploy Web applications in a DMZ to enable someone outside your company fire wall to access your Web applications.

### Deploy webx.war on Apache/Tomcat 3.2

If you are deploying a .war file in a DMZ that will invoke a business process in Sterling Integrator, you must also deploy the HTTP servlet (webx.war) in the same DMZ environment.

When deploying the webx.war file on a computer other than one that has Sterling Integrator installed on it, complete the following steps:



1. Verify that you have an installed copy of Apache/Tomcat on the computer that you want to deploy the webx.war on.
2. Verify that the JDK version installed on the computer is the same version required by Sterling Integrator.
3. Deploy the webx.war file on the new applications server.
4. In Sterling Integrator, open the HTTP Server adapter configuration and select the Web Extensions HTTP Server Adapter configuration.
5. Type the IP address of the computer that is in the DMZ.
6. In the applications server where you deployed the webx.war file, locate and open the webx web.xml file.
7. In the webx web.xml file, verify that the port number matches the HTTP Server adapter configuration on Sterling Integrator.

The web.xml file is located in the /tomcat\_install\_dir/webapps/webx/web-inf/ directory.

The web.xml file is located in the /tomcat\_install\_dir/webapps/webx/web-inf/ directory.

The HTTP Server adapter and the b2bhttp servlet are configured to send data to and receive data from each other.

**Note:** For instructions on deploying .war files on other applications servers, see the documentation for your specific applications server.

To deploy the webx.war file on Apache/Tomcat 3.2:

1. Stop Tomcat.
2. From the Sterling Integrator **Deployment** menu, select **Web Extensions > Utilities**.
3. In the Download area, select from the list the **webx.war** file and click **Go!**
4. In the **File Download** dialog box, click **Save**.
5. In the **Save As** dialog box, select the location that you want to save the b2bhttp.war file to. Do not change the name of the file.
6. Copy **webx.war** from the location that you saved the file to the Web application folder in the DMZ:  
For example, copy the webx.war to the /usr/local/tomcat3.2.2/webapp directory.
7. Restart Tomcat.

## Deploy a Web Application in a DMZ Using Perimeter Servers

You can deploy your Web applications in a DMZ using perimeter servers. Using perimeter servers provides security features and performance enhancements. You must use the perimeter servers configuration if your Web templates are JSPs.

To deploy a Web application in a DMZ using perimeter services:

1. Configure the perimeter server in Sterling Integrator.
2. Deploy the perimeter server to the DMZ.
3. Edit the HTTP Server adapter in Sterling Integrator.

4. Test the Web application.

## Edit the HTTP Server Adapter

To edit the HTTP Server adapter:

1. From the **Deployment** menu, select **Services > Configuration**.
2. Using either the **List** or **Search** feature, locate the HTTP Server adapter.
3. In the Service Configuration List, next to the HTTP Server adapter, click **copy**.
4. In the **Name** field, type a new descriptive name.
5. In the **Description** field, type a new description of the adapter.
6. Choose the appropriate group that you want to associate with this adapter and click **Next**.
7. In the **Host Listen Port** field, type the port number from the DMZ computer on which the adapter listens.
8. In the **Perimeter Server Name** field, select the appropriate name from the list.
9. Choose the appropriate document storage, user authentication, SSL, certificate, and cipher strength options and click **Next**.
10. In the URI page, next to **New URI**, click **add**.
11. In the URI Config page, in the **URI** field, type the URI you are adding to the adapter.
12. Under Launch BP Or War, select **WAR File**, and click **Next**.
13. In the WAR Config page, in the **Enter WAR File Path** field, type the complete path to the .war file and click **Next**.
14. In the URI page, click **Next**.
15. In the Confirm page, verify the settings, select **Enable Service for Business Processes**, and click **Finish**.

## Test the Web Application

After you deploy a Web application in a DMZ using the perimeter servers, you must test the application to ensure that it works properly.

To test a Web application deployed in a DMZ:

1. Point your Web browser to `http://host:port/filename`, where the host is the DMZ IP address, the port is the HTTP Server adapter port, and the filename is the name of the .war file deployed.
2. Log in using your user name and password.

## A

Authentication (authenticate) tag library 11  
Authorization (autho) tag library 10

## C

checking in  
    Web resource 20  
checking out  
    Web resource 20

## D

deleting  
    Web Resource 22  
deploying  
    webx.war 24  
documents  
    routing 6

## F

features, Web Extensions 5

## H

Human Interaction Document Loader service function 5  
Human Interaction Event service function 5  
Human Interaction Query service function 5  
Human Interaction service  
    overview 5  
    types 5

## J

Java Server Pages (JSP) 5, 7  
JSP tag libraries  
    prerequisites 7

## L

license file, obtaining 5

## M

managing  
    .war file 23

## P

perimeter servers 25  
perimeter services 5  
permissions  
    role-based security 6

## R

resource tag 19  
resource, Web  
    checking in 20  
    checking out 20  
    locating 21  
    version history 23  
role-based security  
    document routing 6  
    permissions 6  
    user account 6

## S

searching  
    Web resource 21  
service, Human Interaction 5  
Session Validation (validSession) tag library 11

## U

user account, role-based security 6

**V**

version history  
  Web resource 23

**W**

Web archive (.war) file 19

Web Extensions

  example 5  
  features 5

Web resource

  checking in 20  
  checking out 20  
  general 19  
  locating 21  
  managing 19  
  searching 21  
  version history 23

Web.xml 14

webx.war 24, 25

work page

  editing 19