Sterling Integrator®

Web Extensions

Version 5.1

Contents

Overview **3**
Human Interaction Services 3

Role-Based Security **5**

Using JavaServer Pages (JSP) Tag Libraries with Web Extensions **6**
Before You Begin Using JSP Tag Libraries 6
Run Business Process Tag Libraries 7
Business Process Resolver 8
Authorization (autho) Tag Library 10
Session Validation (validSession) Tag Library 11
Authentication (authenticate) Tag Library 11
JavaServer Pages Standard Tag Library (JSTL) - Overview 11
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 16
Delete JSTL Code from a JSP Web Template 16
Update, Compile, and Redeploy the WAR file in Sterling Integrator 16
Deploy an External JSP Web Application on Sterling Integrator 17
Troubleshooting JSTL Issues 17
JSTL Demonstrations 18

Manage Web Resources **19**
Check In a Web Resource 20
Check Out a Web Resource 20
Search and Delete a Web Resource 21
Maintain a Web Resource Version History 22

Manage Web ARchive (WAR) Files **24**

Deploy a Web Application in a Demilitarized Zone (DMZ) **25**
Deploy webx.war on Apache/Tomcat 3.2 25
Deploy a Web Application in a DMZ Using Perimeter Servers 26
Edit the HTTP Server Adapter 26

Copyright **28**

Overview

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:

- Enable addition of human interaction to virtually any business process
- Support for Java Server Pages (JSP) when using perimeter services
- Tag JSP libraries
- Import and export of web resources
- Support JSP-based WebARchive (WAR) files
- Support localization, international standards, and double-byte character sets (DBCS)
- Notify users through E-mail on business processes waiting for human interaction
- Protect against unauthorized access through role-based security features

Obtain a License for Web Extensions

To use Sterling Integrator Web Extensions, you must obtain a license. You must obtain a license from the Sterling Commerce Self-Service Web site at *www.productupdates.stercomm.com*.

Human Interaction Services

Human interaction is any action taken by a person on a business process when it runs. Web Extensions uses Human Interaction services to enable human interaction.

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 displays 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 following table lists the different types of Human Interaction services:

Service	Description
Human Interaction Event service	• Identify business process data requiring human interaction • Flag business processes requiring human interaction • Create references to the business process data awaiting human action.
Human Interaction Query service	• Search the storage area containing the business processes flagged by the Human Interaction Event service for human interaction • Retrieve the business process data
Human Interaction Document Loader service	• Receive a MinedDataId as input and use it 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 login, 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.

Using JavaServer Pages (JSP) Tag Libraries with Web Extensions

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 JSP tag library is a collection of predefined JSP components that you can use in your JSP Web templates. You can 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 Begin Using JSP Tag Libraries

Before you begin using 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*/installed_data/B2BF/components/b2b_base/jars/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<tagelib-location>
</taglib>
```

Run Business Process Tag Libraries

You can use the tag libraries to run a single business process per JSP page. The following section describes the business process tag libraries.

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 used by the runBP tag library:

Attribute	Description
user	(Required) User ID This attribute may be specified at the business process run time.
nvp	(Required) Name of the value pairs to be placed into process data in the format name=value&name2=value2. This attribute may be specified at the business process run time.
bpname	(Required) Name of the business process to run. This attribute may be specified at the business process run time.
pridoc	(Required) Primary document that invokes the business process. 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 used by the runBPnoReturnDoc tag library:

Attribute	Description
user	(Required) User ID This attribute may be specified at the business process run time.
nvp	(Required) Name of the value pairs to be placed into process data in the format name=value&name2=value2.

Attribute	Description
	This attribute may be specified at the business process run time.
bpname	(Required) Name of the business process to run. This attribute may be specified at the business process run time.
pridoc	(Required) Primary document that invokes the business process. 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 used by the getDocuments tag library:

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

Business Process Resolver

You can 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 an 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, to verify if you belong to a Sterling Integrator group or to verify your specified Sterling Integrator permission. If the authorization fails, the JSP body is not displayed or you are redirected to an error page. If neither permissions nor groups attributes are set, you are authorized.

Since authorization is based on at least one match against permission or a 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 used by the autho tag library:

Attribute	Description
user	(Optional) User ID 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	(Optional) Group ID 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	(Optional) Permission ID This attribute may be specified at the business process run time.
permissions	List or ArrayList of permission 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	(Optional) URL of the login page that you are redirected to if the session expires or is invalid. A session is invalid if a user name attribute is not present.. 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	(Optional) Checks for a group or permission association prior to authorization. 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 if your session is active or expired with Sterling Integrator. If the validation fails, an error page or login page displays.

The following table describes the attributes used by the validSession tag library:

Attribute	Description
login	(Required) URL of the login page that you are redirected to if your session expires or is invalid. A session is invalid if a user name attribute is not present.. 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 to access the system.

The following table describes the attributes used by the authenticate tag library:

Attribute	Description
user	(Required) User ID you enter in the login page. This attribute may be specified at the business process run time.
pass	(Required) Password you enter in the login page. This attribute may be specified at the business process run time.
login	(Required) URL of the login page that you are redirected to if your session expires or is invalid. A session is invalid if a user name attribute is not present. 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.

JavaServer Pages Standard Tag Library (JSTL) - Overview

A JavaServer Pages Standard Tag Library (JSTL) is a consolidated grouping of common 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 to improve consistency of presentation and handling of data
- Increase efficiency in your business by standardizing code to reduce errors in Web template development
- Reduce costs and improve revenue generation by increasing speed of 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 Sterling Integrator 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.
- Sterling Integrator with Web Extensions.
- Web application created using J2EE.

JSTL Concepts and Definitions

The following table lists the JSTL concepts and its 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 Universal Resource Indicator (URI) of the tag library and the prefix used in the tags for that specific functional area library.
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.

Concept	Definition
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. For more information on adding JSTL to the Web application, refer to the topic Add the JSTL to the Web Application.	
2	Edit the web.xml file for the Web application to include the tag libraries. For more information on editing web.xml file, refer to the topic Edit the web.xml File for JSTL.	
3	Add the JSTL Namespace and tags to the JSP Web template. Add the JSTL Namespace and tags to the JSP Web template.	

Step	Description	Your Notes
4	You can use the jar utility that is installed with the JDK to compile the WAR file. Update, Compile, and Redeploy the WAR file in Sterling Integrator. For more information on updating, compiling, and redeploying the WAR file, refer to the topic Update, Compile, and Redeploy the WAR file in Sterling Integrator.	
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.

To add the JSTL to your Web application:

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/jarkarta_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. For more information about editing the web.xml files, refer to the topic *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)	`<taglib>`
	`<taglib-uri>http://java.sun.com/jstl/core<taglib-uri>`
	`<taglib-location>tld/c-1_0.tld</taglib-location>`
	`</taglib>`

Core (RT)	`<taglib>`
	`<taglib-uri>http://java.sun.com/jstl/core-rt<taglib-uri>`
	`<taglib-location>tld/c-1_0-rt.tld</taglib-location>`
	`</taglib>`

Format (EL)	`<taglib>`
	`<taglib-uri>http://java.sun.com/jstl/fmt<taglib-uri>`
	`<taglib-location>tld/fmt-1_0.tld</taglib-location>`
	`</taglib>`

Format (RT)	`<taglib>`
	`<taglib-uri>http://java.sun.com/jstl/fmt-rt<taglib-uri>`
	`<taglib-location>tld/fmt-1_0-rt.tld</taglib-location>`
	`</taglib>`

SQL (EL)	`<taglib>`
	`<taglib-uri>http://java.sun.com/jstl/sql<taglib-uri>`
	`<taglib-location>tld/sql-1_0.tld</taglib-location>`
	`</taglib>`

SQL (RT)	`<taglib>`
	`<taglib-uri>http://java.sun.com/jstl/sql-rt<taglib-uri>`
	`<taglib-location>tld/sql-1_0-rt.tld</taglib-location>`
	`</taglib>`

XML (EL)	`<taglib>`
	`<taglib-uri>http://java.sun.com/jstl/xml<taglib-uri>`
	`<taglib-location>tld/x-1_0.tld</taglib-location>`
	`</taglib>`

XML (RT)	`<taglib>`
	`<taglib-uri>http://java.sun.com/jstl/xml-rt<taglib-uri>`
	`<taglib-location>tld/x-1_0-rt.tld</taglib-location>`
	`</taglib>`

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.

To add JSTL code to a JSP Web template:

1. Open the JSP Web template in a text editor.
2. On The line preceding the <HTML> tag, for each functional library that you 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

To delete JSTL code from a JSP Web template:

1. Open the JSP Web template in a text 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 Sterling Integrator

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

To deploy the WAR file in Sterling Integrator:

1. Access the computer on which Sterling Integrator is installed.
2. Change your working directory to *install_dir*/container/applications directory.
3. 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)
4. Change your working directory to *install_dir*/bin directory.
5. 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.

To deploy an external JSP Web application on Sterling Integrator:

1. Change your working directory to *install_dir*/noapp/deploy directory and copy the WAR file.
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.

Troubleshooting JSTL Issues

The following table lists the methodology to troubleshoot JSTL issues:

Situation	Resolution
General Errors	• Verify that you defined the JSTL Namespace in the JSP Web template. • Verify that you added the complete JSTL tag to the JSP Web template. • Troubleshoot your JSP code.
Class Not Found Errors	• Verify that you included the JSTL libraries to the Web application. • Troubleshoot your JSP code.

Situation	Resolution
Page Cannot Display Error	• Verify that the prefix defined in JSTL Namespace and tag match the same `<short-name>` in the .tld file for that functional area.

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.	1. In the **Type a number to count** text box, enter the number you want the computer to count to. 2. Click **Go!** The list of numbers up to the number you entered in the text box displays in sequence.
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 **Go!** next to the list where you selected JSTL XML Demo. The parsed document displays in the JSP Web template.
JSTL Timezone Demonstration	Demonstrates the formatting functional area Format Timezone feature by letting you select a city to display the time and date of. **Note:** The demonstration does not adjust for Daylight Savings Time.	Perform the following steps: 1. From the list, select a city that you want to display the time for. 2. Click **Go!** The time and date for the city you selected displays in the updated JSP Web template.

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.

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 that 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:

1. Check in the Web resources you want to include in the WAR file.

   For more information about checking in a web resource, refer to the topic *Check In a Web Resource* on page 20.
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 about generating a WAR file, refer to the topic *Manage Web ARchive (WAR) Files* on page 24.

Check In a Web Resource

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

To check in a Web resource:

1. From the **Deployment** menu, select **Web Extensions** > **Web Resources**. The Web Resources page opens.
2. On the Web Resources page, under **Check In Web Resource**, next to **Check In new Web Resource File**, click **Go!**.
3. On The Select Resource page, select the web resource file name in the **Web Resource File Name** field.
4. Enter a description in the **Description** field to help you identify this version of the Web resource.

   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. Click **Next**.
6. On The Naming page, next to **Resource Name**, enter a name that will identify the Web resource in Sterling Integrator.

   If 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.
7. Select the appropriate **Resource Type** 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.
8. On The Confirm page, review the changes and perform one of the following tasks:
   - To modify the Web resource check in options, click **Back**.
   - To abandon the Web resource check in options, click **Cancel**.
   - To save the Web resource check in options, 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 **Deployment** menu, select **Web Extensions** > **Web Resources**.
2. On The Web Resources page, under **Search**, next to **Resource Name**, enter the Web resource name to check out and click **Go!**

   You can also locate the Web resource by using the List option.

   For more information about searching Web resources, refer to the topic *Search and Delete a Web Resource* on page 21.
3. On The Web Resource Management page, click **source manager** next to the Web resource you want to check out.
4. On The Web Resource Source Manager page, click **check out** next to the Web resource you want to check out.
5. When prompted with the following message, click **OK** to edit the file or click **Cancel** to check out a read-only copy of the file.

   ```
   Click OK to lock the file for editing. During the check in process, you can
   release the lock. Click Cancel to check out a read-only copy of the file.
   ```
6. Save the Web resource file in an appropriate location.

Search and Delete a Web Resource

After a Web resource is checked in, you can search and delete the Web resource 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 search for a Web resource:

1. From the **Deployment** menu, select **Web Extensions** > **Web Resources**.
2. On The Web Resources page, complete one of the following actions:
   - Under **Search**, next to **Resource Name**, enter 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 entered.
   - Under **List**, next to **Alphabetically** drop-down list, 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**, next to **by Resource Type** drop-down list, 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.

3. On the Web Resource Management page, click **source manager** next to the Web resource you want to delete.
4. On the Web Resource Source Manager page, select **delete** check box next to the Web resource you want to delete from the Sterling Integrator database.
5. Next to **Delete Selected Versions**, click **Go!**.

   Note: You can also delete all versions of the Web resource by clicking **Go!** next to **Delete All Versions**.

6. When prompted with the following message, click **OK**.

   ```
   Are you sure you want to delete the selected versions?
   ```

7. On the Delete Resources page, review the information about the Web resource and click **Next**.
8. On the Confirm page, the following message is displayed:

   ```
   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. You can provide a unique and meaningful description as it is the basis for maintaining the version for that Web 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 **Deployment** menu, select **Web Extensions** > **Web Resources**.
2. On The Web Resources page, under **Search**, next to **Resource Name**, enter the Web resource name to check out and click **Go!**

   You can also locate the Web resource by using the List option.

   For more information about searching Web resources, refer to the topic *Search and Delete a Web Resource* on page 21.
3. On The Web Resource Management page, click **version manager** next to the Web resource you want to review.
4. On The Web Resource Version Manager page, select the **Default** radio button for the version you want to set as the default resource. The business processes will use the default version of the Web resource.
5. Click **Save**. Your changes are applied to the Web resources in the Sterling Integrator database and you will get the following message: `Resource status has been successfully updated`

Manage Web ARchive (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 in a production environment
- Deploy Web applications easily in a demilitarized zone (DMZ)

To generate a WAR file:

1. From the **Deployment** menu, select **Web Extensions** > **Utilities**.
2. On The Web Utilities page, under **Export Web Resources**, next to **Generate a WAR File using system resources**, click **Go!**
3. On The **Export** page, next to WAR File Name: field, enter a unique name for the WAR file, including the .war file extension.
4. Select the appropriate resource tag you want to include in the WAR file from the **Resource Tag** drop-down list.
5. Select the check box next to **Save generated WAR file on Server** if you want to save the WAR file on the server.
6. Click **Next**.
7. On The Confirm page, verify the WAR file information and click **Finish**.
8. On 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.

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

Deploy a Web Application in a Demilitarized Zone (DMZ)

You can deploy Web applications in a demilitarized zone (DMZ) to enable external authorized users outside your company firewall 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.

1. Complete the following steps if you are deploying webx.war on a computer other than one that has Sterling Integrator installed on it:
   a) Verify that you have Apache/Tomcat installed on the computer that you want to deploy the webx.war on.
   b) Verify that the JDK version installed on the computer is the same version required by Sterling Integrator.
   c) Deploy the webx.war file on the new applications server.
   d) In Sterling Integrator, open the HTTP Server adapter configuration and select the Web Extensions HTTP Server Adapter configuration.
   e) Type the IP address of the computer that is in the DMZ.
   f) In the application server where you deployed the webx.war file, locate and open the webx web.xml file.

      The web.xml file is located in the /tomcat_install_dir/webapps/webx/web-inf/ directory.

   g) 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 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, refer to your application server documentation.

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

a) Stop Tomcat.
b) From the **Deployment** menu, select **Web Extensions** > **Utilities**.
c) On The Web Utilities page, under Download, select **webx.war** from the **Download (.war) file** drop-down list and click **Go!**
d) Select the location to save the WAR file.
e) 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.

f) Restart Tomcat.

Deploy a Web Application in a DMZ Using Perimeter Servers

You can deploy your Web applications in a DMZ using perimeter servers, which provides security features and performance enhancements. You must use the perimeter server 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. For more information about editing HTTP server adapter, refer to the topic *Edit the HTTP Server Adapter* on page 26.
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. On The Services Configuration page, next to the **HTTP Server Adapter**, click **copy**.
4. In the **Name** field, enter a new descriptive name.
5. In the **Description** field, enter a new description for 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, enter 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. On The URI page, next to **New URI**, click **add**.
11. On The URI Config page, next to **URI** field, enter the URI you are adding to the adapter.
12. Under Launch BP Or War, select **WAR File**, and click **Next**.
13. On The WAR Config page, in the **Enter WAR File Path** field, enter the complete path to the WAR file and click **Next**.
14. On The URI page, click **Next**.

15. On 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, where 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.

Copyright

Index

A

Authentication (authenticate) tag library 11
Authorization (autho) tag library 10
Authorization(autho) tag library 9

C

checking out
    Web resource 20

D

deploying
    webx.war 25
documents
    routing 5

F

features, Web Extensions 3

H

Human Interaction Document Loader service function 4
Human Interaction Event service function 4
Human Interaction Query service function 4
Human Interaction service
    overview 3
    types 3

J

Java Server Pages (JSP) 3, 6
JSP tag libraries
    prerequisites 6

L

license file, obtaining 3

M

managing
    .war file 24

P

perimeter servers 26
perimeter services 3
permissions
    role-based security 5

R

resource tag 19
resource, Web
    checking out 20
    locating 21
    version history 22
role-based security
    document routing 5
    permissions 5
    user account 5

S

searching
    Web resource 21
service, Human Interaction 3
Session Validation (validSession) tag library 11

U

user account, role-based security 5

V

version history
    Web resource 22

W

Web archive (.war) file 19
Web Extensions
    features 3
Web resource
    checking out 20
    deleting 21
    general 19
    locating 21
    managing 19
    searching 21
    version history 22
Web.xml 14
webx.war 25
work page
    editing 19