
# SDWIS/LabToState 2.0 Installation Guide (Draft)

United States Environmental Protection Agency

Office of Ground Water and Drinking Water

Contract No.
EPA -1001002
SDWIS Project
Product Control No.
Guident-SAIC-SDWIS-8.d1c
Deliverable No. 8.d1c


December 19, 2007

# SDWIS/LABTOSTATE 2.0 INSTALLATION GUIDE
(Draft)

**CONTRACT NO. EPA -1001002**

**SDWIS PROJECT**

**Prepared for:**

**United States Environmental Protection Agency**
**Office of Ground Water and Drinking Water**
**Drinking Water Protection Division**
**1200 Pennsylvania Ave., NW**
**Washington, DC 20460**

**Contracting Officer Representative**

**Clint Lemmons**

**Prepared by:**

**SAIC Solutions Delivery Center**
**Science Applications International Corporation**
**6565 Arlington Boulevard**
**Falls Church, VA 22042**

## CONTENTS



## EXHIBITS

Draft

# 1.0 INTRODUCTION

To electronically submit drinking water sample results to SDWIS/XML Sampling for processing and permanent storage in the SDWIS database, the sample results must be formatted in XML documents. SDWIS/LabToState assists laboratories and primacy agencies with the formatting and validation of XML documents. The *SDWIS/LabToState 2.0 Installation Guide* contains instructions for installing SDWIS/LabToState 2.0.

**Note:** This document is intended for use by primacy agencies, which will in turn provide operation-specific instructions for the laboratories.

## 1.1 Document Overview

This document contains instructions for installing and administering SDWIS/LabToState 2.0. It comprises the following sections:

- Section 1.0, Introduction, describes the content of the document, the software and hardware test environment, and user support procedures.
- Section 2.0, Installation Procedures, describes the steps required to install SDWIS/LabToState 2.0 as a stand alone client application and as a web-based application using a J2EE application server.
- Section 3.0, Site Customization Procedures, describes the steps required to customize SDWIS/LabToState 2.0 at your site.
- Section 4.0, CROMERR Implementation, describes the SDWIS/LabToState's implementation of the CROMERR requirements.

## 1.2 Software and Hardware Test Environment

SDWIS/LabToState 2.0 was tested using the hardware and supporting software versions listed in Exhibit 1. More recent versions of the supporting software may be available; the application, however, has not been tested with these newer versions.

As a web based application, the SDWIS/LabToState software may reside on an application server or on a network-connected workstation configured as a stand alone server. As a stand alone client application, the SDWIS/LabToState software may be installed on any windows based PC.

SDWIS/LabToState may interface with the site's local e-mail server. Additionally, for sites using the SQL PlugIn to retrieve sample data, SDWIS/LabToState must have access to the database server from which it is to retrieve data. The SDWIS/LabToState SQL PlugIn was developed and tested using databases deployed on Oracle and Microsoft Access.

| Platform | Operating System | Software | Hardware |
|---|---|---|---|
| Application Server / Web Server | WINDOWS XP | WINDOWS XP<br><br>Internet Explorer v6.0 with Service Pack 1<br><br>Java 2 SDK, Standard Edition (J2SE), v 1.5 and v 1.6.03<br>(To use J2SK v1.5, see Release Notes, Section 3.0, Notes and Advisories)<br><br>Tomcat 6.0.14<br><br>SDWIS/XML Sampling v2.0 | Dell OptiPlex GX400<br>(Pentium 4 – 1.4 GHz Processor)<br><br>1 GB RAM<br><br>Screen Resolution: 1024 x 768 and Small Fonts |
| Database Server | Windows 2000 with Server Service Pack 4 | Software with Service Pack 6<br>Oracle 10g | Compaq Proliant DL380, Pentium III, 933 MHz CPU, 4 GB Hard Disk |
| E-Mail Server | Windows 2003 Server | MS Exchange 2000 | Compaq Proliant ML370 G2, two 1.3 MHz CPU, 2 MB of memory |
| Client Workstation | Windows XP Professional Version 2002 with Service Pack 2 | Internet Explorer v6.0 with Service Pack 1 | Dell OptiPlex GX400<br>(Pentium 4 – 1.4 GHz Processor)<br><br>1 GB RAM<br><br>Screen Resolution: 1024 x 768 and Small Fonts |

**Exhibit 1.** Hardware and Software Test Environment

### 1.3 User Support

As a representative of a primacy agency, you may call the SDWIS User Support Hotline at (703) 292-6298 or e-mail SDWIS User Support at sdwis@saic.com. If you represent a laboratory using the SDWIS/LabToState product, you are encouraged to contact the primacy agency to which you are reporting sample data. The primacy agency will in turn contact the SDWIS User Support Hotline on your behalf. The hotline/e-mail account is intended as a technical support tool for the operation and functionality of SDWIS products such as SDWIS/LabToState. A SDWIS team member answers calls to the hotline between 8 a.m. and 5 p.m., Eastern Standard Time (EST), on weekdays (except federal holidays) and offers immediate user support when possible. During evenings, weekends, or those times when the support

personnel are temporarily not available, callers may leave a detailed message. Questions requiring the expertise of other team members, such as developers or subject matter experts, are documented in the Events Tracking System, forwarded to those individuals, and answered as soon as possible.

## 2.0 INSTALLATION PROCEDURES

SDWIS/LabToState may be deployed as a web based, platform-independent application or as a stand alone client application. It allows laboratories and other entities to format sample data into XML documents, which are transferred to SDWIS/XML Sampling for additional processing. The software is designed for use by and is available to all laboratories, primacy agencies, and other entities to format sample data for submission to primacy agencies using SDWIS/XML Sampling.

If SDWIS/LabToState is deployed at the primacy agency, the primacy agency configures the format and the content of the files submitted by the laboratories and describes that format to the laboratory. The laboratory generates files in the prescribed format and uploads those files using SDWIS/LabToState as it is configured and deployed on the primacy agency's web-site.

A laboratory may elect to install SDWIS/LabToState at its site and locally generate the XML documents containing the sample data. Then, the laboratory uploads the XML documents using procedures described by the primacy agency. A locally deployed version of SDWIS/LabToState may be suitable for laboratories with a well-defined LIMS system; laboratories that submit larger volumes of data; or laboratories that submit sample data to multiple state agencies.

If you are installing SDWIS/LabToState as a stand alone client application, proceed to Section 2.1, Stand Alone Client Application Installation. To install it as a web-based application, proceed to Section 2.2, Web Based Application Installation.

### 2.1 Stand Alone Client Application Installation

This section describes procedures for installing SDWIS/LabToState as a stand alone client application. The SDWIS/LabToState software interfaces with the site's e-mail server. Therefore, a System Administrator familiar with the e-mail server must be available to you (as the person installing the SDWIS/LabToState application) throughout the installation process. For primacy agencies using the SQL PlugIn, SDWIS/LabToState interfaces with your database and the Database Administrator (DBA) must also be available.

If you are installing SDWIS/LabToState and need to upgrade your current installation, you may proceed to Section 2.1.1, Upgrade Installation-Stand Alone Client Application. If you have never installed it at your site, you should proceed to Section 2.1.2, First Time Installation-Stand Alone Client Application.

Draft

### 2.1.1 Upgrade Installation-Stand Alone Client Application

Exhibit 2 is a checklist of the steps required to upgrade to a new release of SDWIS/LabToState. The remainder of this section further describes each step in the upgrade installation procedures.

| SDWIS/LabToState Upgrade Installation Checklist | | |
|---|---|---|
| **Installation Step** | **Installation Activity Description** | **Installation Guide Reference** |
| 1 | Archive SDWIS/LabToState Files | Section 2.1.1.1, Archive SDWIS/LabToState Files |
| 2 | Remove the SDWIS/LabToState Software | Section 2.1.1.2, Remove the SDWIS/LabToState Software |
| 3 | Install SDWIS/LabToState Software | Section 2.1.1.3, Install SDWIS/LabToState Software |

**Exhibit 2.** SDWIS/LabToState Upgrade Installation Checklist

#### 2.1.1.1 Archive SDWIS/LabToState Files

The first step to upgrading SDWIS/LabToState is to move the files and directories containing SDWIS/LabToState processing information. Otherwise, those files and directories will be deleted during the installation of the new release. The SDWIS/LabToState processing files and directories you need to move are listed in Exhibit 3.

| File or Directory | Name | Description |
|---|---|---|
| Directory | [InstalledDirectory]\docBase\job Folders | Contains the uploaded sample data, output reports, and generated XML documents for each job processed by SDWIS/LabToState. |
| Directory | [InstalledDirectory]\docBase\WEB-INF\classes\properties | Contains site-specific values used to configure the SDWIS/LabToState software. |

**Exhibit 3.** SDWIS/LabToState Files To Archive

The Job Folders directory is archived to maintain a history of the sample data processed by SDWIS/LabToState at your site. The properties directory is archived so that it may be used as a reference to set the values of the site configuration files during the SDWIS/LabToState installation.

**Note:** Since the Site Configuration files may have changed since the last installation of SDWIS/LabToState, you should use the current values as a reference to update values when you install the application again.

#### 2.1.1.2 Remove the SDWIS/LabToState Software

In Step 2, you remove the SDWIS/LabToState software by deleting the folder in which you originally installed the software.

#### 2.1.1.3 Install SDWIS/LabToState Software

After the existing release of SDWIS/LabToState is removed, you should install the SDWIS/LabToState software using the installation procedures described in Section 2.1.2, First Time Installation-Stand Alone Client Application.

### 2.1.2 First Time Installation-Stand Alone Client Application

This section describes procedures to install SDWIS/LabToState as a stand alone-client application for the first time. Exhibit 4 is a checklist of the steps required to install SDWIS/LabToState. The remainder of this section further describes each step in the installation procedures.

| SDWIS/LabToState Installation Check List | | |
|---|---|---|
| **Installation Step** | **Installation Activity Description** | **Installation Guide Reference** |
| 1 | Download SDWIS/LabToState Installation Package and documentation | Section 2.1.2.1, Download SDWIS/LabToState Installation Package |
| 2 | Install the SDWIS/LabToState software | Section 2.1.2.2, Install the SDWIS/LabToState Software |
| 3 | Start SDWIS/LabToState | Section 2.1.2.3, SDWIS/LabToState Startup |
| 4 | Shut down SDWIS/LabToState | Section 2.1.2.4, SDWIS/LabToState Shut Down |
| 5 | Configure SDWIS/LabToState | Section 2.1.2.5, Configure SDWIS/LabToState |

**Exhibit 4.** Install SDWIS/LabToState-Stand Alone Client Application

Draft

### 2.1.2.1 Download SDWIS/LabToState Installation Package

The first step is to download the SDWIS/LabToState 2.0 Installation Package from the SDWIS web site (http://www.epa.gov/safewater/sdwisfed/sdwismod.htm). You should also download the *SDWIS/LabToState 2.0 User Guide* (Guident-SAIC-SDWIS-8-d1d, December 19, 2007) and the *SDWIS/LabToState 2.0 Release Notes* (Guident-SAIC-SDWIS-8-d1a, December 19, 2007).

### 2.1.2.2 Install the SDWIS/LabToState Software

In Step 2, you unzip the installation package to a directory of your choice. To simplify the installation of the stand alone client application, the installation package bundles the required supporting software (e.g., Java SDK and Tomcat server). Exhibit 5 provides an annotated list of the directory structure after unzipping the installation package.

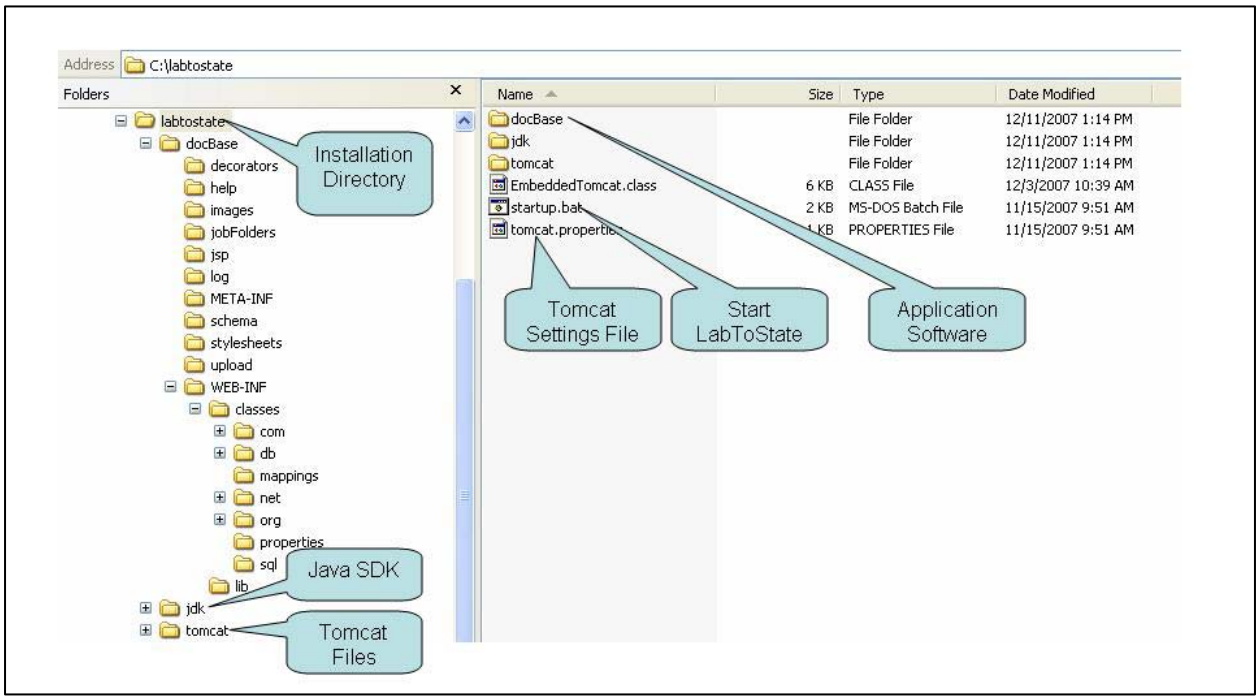



**Exhibit 5.** Install SDWIS/LabToState-Web Based Application

### 2.1.2.3 SDWIS/LabToState Startup

In Step 3, you start the application by double clicking on the startup.bat located in the folder in which you installed the application. As depicted in Exhibit 6, this procedure opens a DOS window for the application process. It also opens the default browser pointing to the URL of the application.

**Note:** You may see warning as depicted in Exhibit 6 – please ignore.

The default hostname is set to “localhost” and the default port is set to “8090”. The URL to the application with these settings is http://localhost:8090/labtostate/jsp/index.jsp. You may change the host name and port number of the embedded Tomcat by changing the tomcat.properties file found in the top folder of the application installation directory.

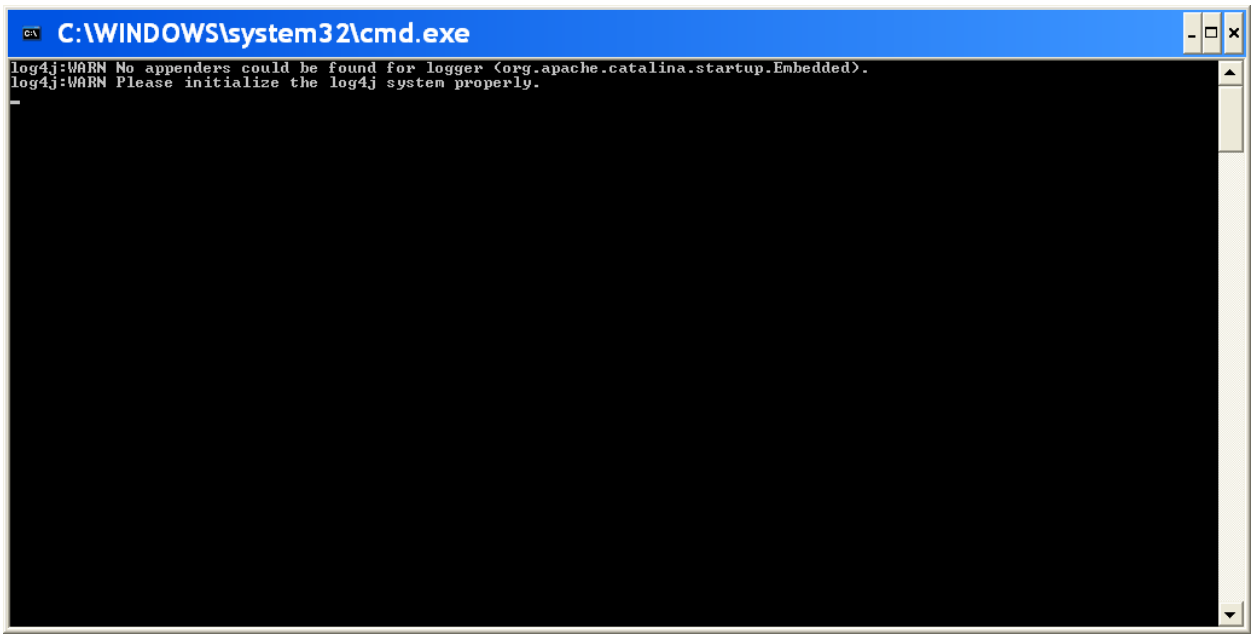



**Exhibit 6.** Stand alone Client Application DOS Window

#### 2.1.2.4 SDWIS/LabToState Shut Down

In Step 4, you shut down the application by closing the DOS window. To close the DOS window click the “X” in the upper right corner of the DOS window.

#### 2.1.2.5 Configure SDWIS/LabToState

After completing the above steps, the installation of SDWIS/LabToState software is complete. You should continue to Section 3.0, Site Customization Procedures of this document to configure the SDWIS/LabToState software for your site.

### 2.2 Web Based Application Installation

This section describes procedures for installing SDWIS/LabToState as a web based application. The SDWIS/LabToState software is installed on a web server and interfaces with the e-mail server. Therefore, a System Administrator familiar with the web server, application server, and e-mail server must be available to you (as the person installing the SDWIS/LabToState application) throughout the installation process. For primacy agencies using SQL PlugIn, SDWIS/LabToState interfaces with your database and the Database Administrator (DBA) must also be available.

The installation procedures described in this section assume that the support software (e.g., web server and Java SDK 1.6) required by SDWIS/LabToState has been previously installed. If you are installing SDWIS/LabToState as a web based application and need to upgrade your current installation, you may proceed to Section 2.2.1, Upgrade Installation-Web Based Application. If you have never installed it at your site, you should proceed to Section 2.2.2, First Time Installation Web-Based Application.

### 2.2.1 Upgrade Installation-Web Based Application

Exhibit 7 is a checklist of the steps required to upgrade to a new release of SDWIS/LabToState. The remainder of this section further describes each step in the upgrade installation procedures.

| SDWIS/LabToState Upgrade Installation Checklist | | |
|---|---|---|
| **Installation Step** | **Installation Activity Description** | **Installation Guide Reference** |
| 1 | Archive SDWIS/LabToState Files | Section 2.2.1.1, Archive SDWIS/LabToState Files |
| 2 | Remove the SDWIS/LabToState Software | Section 2.2.1.2, Remove the SDWIS/LabToState Software |
| 3 | Install SDWIS/LabToState Software | Section 2.2.1.3, Install SDWIS/LabToState Software |

**Exhibit 7.** SDWIS/LabToState Upgrade Installation Checklist

#### 2.2.1.1 Archive SDWIS/LabToState Files

The first step to upgrading SDWIS/LabToState is to move the files and directories containing SDWIS/LabToState processing information. Otherwise, those files and directories will be deleted during the installation of the new release. The SDWIS/LabToState processing files and directories you need to move are listed in Exhibit 8.

| File or Directory | Name | Description |
|---|---|---|
| Directory | *[Tomcat_Home]*\webapps\labtoState\jobFolders | Contains the uploaded sample data, output reports, and generated XML documents for each job processed by SDWIS/LabToState. |
| Directory | *[Tomcat_Home]*\labtoState\webapps\WEB-INF\classes\properties | Contains site-specific values used to configure the SDWIS/LabToState software. |

**Exhibit 8.** SDWIS/LabToState Files To Archive

The Job Folders directory is archived to maintain a history of the XML documents validated and prepared by SDWIS/LabToState at your site. The properties directory is archived so that it may be used as a reference to set the values of the site configuration files during the SDWIS/LabToState installation.

**Note:** Since the Site Configuration files may have changed since the last release, use the current values as a reference to update values when the new release of the application is installed.

#### 2.2.1.2 Remove the SDWIS/LabToState Software

In Step 2, you remove the SDWIS/LabToState software from the application server by deleting the LabToState directory (i.e., *[Tomcat_Home]*\webapps\labtostate).

#### 2.2.1.3 Install SDWIS/LabToState Software

After the existing release of SDWIS/LabToState is removed from the application server, you should install the SDWIS/LabToState software using the installation procedures described in Section 2.2.2, First Time Installation Web-Based Application.

### 2.2.2 First Time Installation Web-Based Application

Exhibit 9 is a checklist of the steps required to install SDWIS/LabToState for the first time. The remainder of this section further describes each step in the first time installation procedures.

| SDWIS/LabToState First Time Installation Check List | | |
|---|---|---|
| **Installation Step** | **Installation Activity Description** | **Installation Guide Reference** |
| 1 | Download SDWIS/LabToState Installation Package and documentation | 2.2.2.1 Download SDWIS/LabToState Installation Package |
| 2 | Install support software (i.e., Java SDK and Tomcat application server) used by SDWIS/LabToState | 2.2.2.2 Install Support Software Used by SDWIS/LabToState |
| 3 | Install the SDWIS/LabToState software | 2.2.2.3 Install the SDWIS/LabToState Software |
| 4 | Set Java environment variable | 2.2.2.4 Set Environment Variables |
| 5 | Configure SDWIS/LabToState for deployment at your site | 2.2.2.6 Configure and Start Up SDWIS/LabToState |

**Exhibit 9.** SDWIS/LabToState First Time Installation Checklist

#### 2.2.2.1 Download SDWIS/LabToState Installation Package

The first step is to download the SDWIS/LabToState 2.0 Installation Package from the SDWIS web site (http://www.epa.gov/safewater/sdwisfed/sdwismod.htm). You should also download the *SDWIS/LabToState 2.0 User Guide* and the *SDWIS/LabToState 2.0 Release Notes*.

#### 2.2.2.2 Install Support Software Used by SDWIS/LabToState

In Step 2, you load the Java 2 Platform Standard Edition, Software Development Kit (SDK), and Tomcat on the application server. If both the Java SDK and Tomcat are already installed on the application server, this step may be skipped. To install or upgrade those components, you should refer to the documentation provided with those products.

**Note:** SDWIS/LabToState 2.0 may be deployed using the JDK v 1.5 or v 1.6. However, to use JDK 1.5, you must copy the jaxb-api.jar from the *[Tomcat_Home]*\webapps\labtoState\WEB-INF\lib to *[Tomcat_Home]*\common\endorsed.

#### 2.2.2.3 Install the SDWIS/LabToState Software

In Step 3, you load the SDWIS/LabToState software on the Tomcat application server. The installation package delivers the SDWIS/LabToState software as a single packed file format known as a Web Archive (WAR) file. This format is generally used to distribute applications before installation. By completing the following tasks, the WAR file is unpacked into the appropriate file structure.

1. Start the Tomcat server.

2. Open a web browser and type **http://localhost:8080** in the address bar. The Tomcat home page will be displayed.

3. Click on the Tomcat Manager link, located on the left side of the page in the box titled Administration. Enter the User Name and Password created for the admin and manager roles in Section 2.1.3, Assign User Identification and Passwords. The Tomcat Web Application Manager page is displayed as depicted in Exhibit 10.

4. Enter the SDWIS/LabToState WAR file in the text box labeled “Select WAR file to upload” located in the Deploy section of the page. Use the **Browse** button to assist with locating the file. After the SDWIS/LabToState War file is displayed in the text box, click on the **Deploy** button. After the application is deployed, the SDWIS/LabToState application is listed in the Applications section of the page.

5. Shut down the Tomcat server by double-clicking the shutdown.bat file.

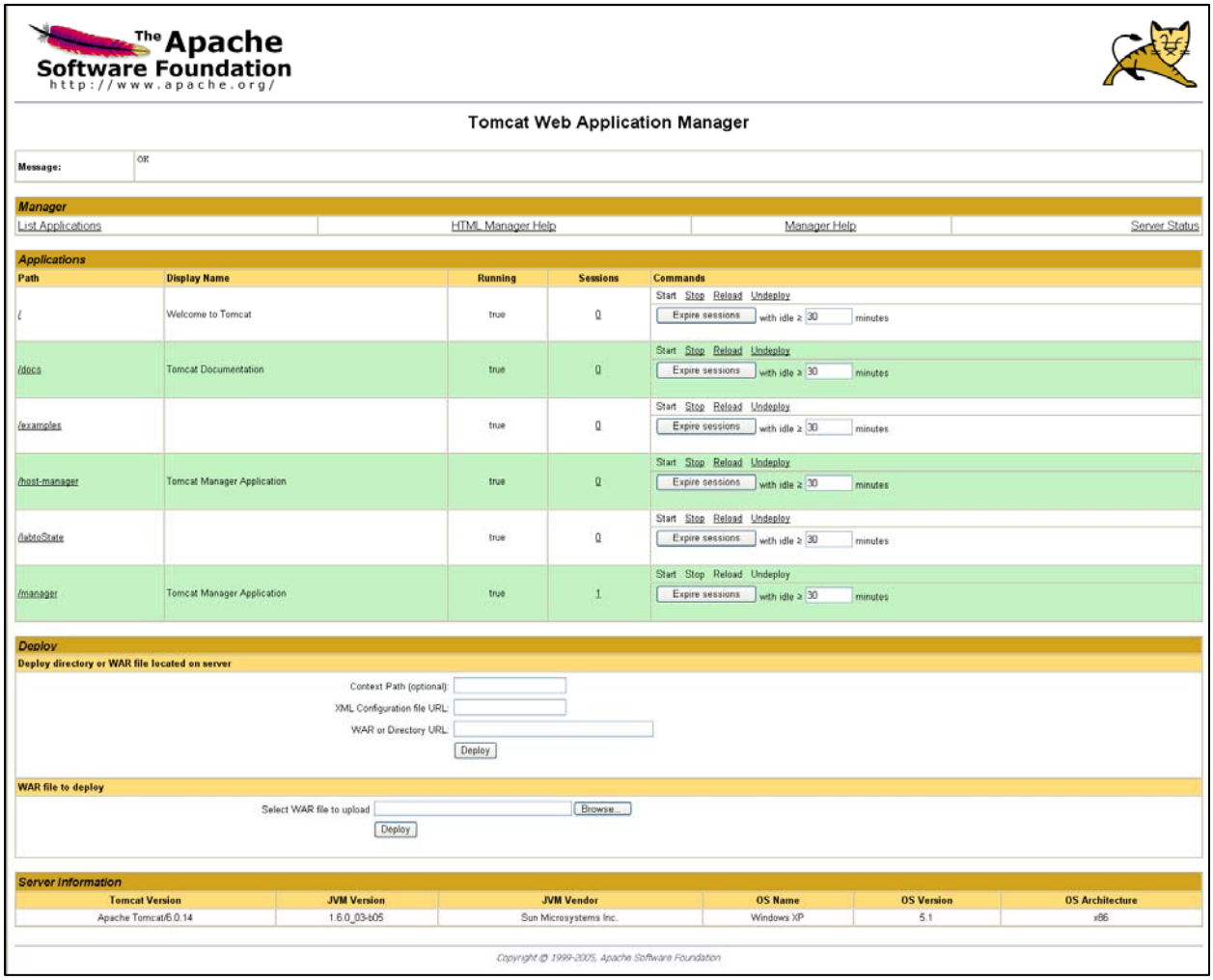



**Exhibit 10.** Tomcat Web Application Manager — Before Deployment

### 2.2.2.4 Set Environment Variables

In Step 5, you set the environment variable used by the Apache Tomcat. The "JAVA_HOME" variable is set to the location of the java compiler software and is used by the Apache Tomcat server. By completing the following tasks, you set these environment variables:

1. From the Window's Control Panel, click on the System icon.

2. Select the Advanced tab.

3. Click on Environment Variables button.

4. In the Environment Variables window (Exhibit 11), add a system variable named JAVA_HOME with the value of the variable set to the location of the java compiler software (e.g., c:\Program Files\Java\jdk1.6.0_03).

Draft

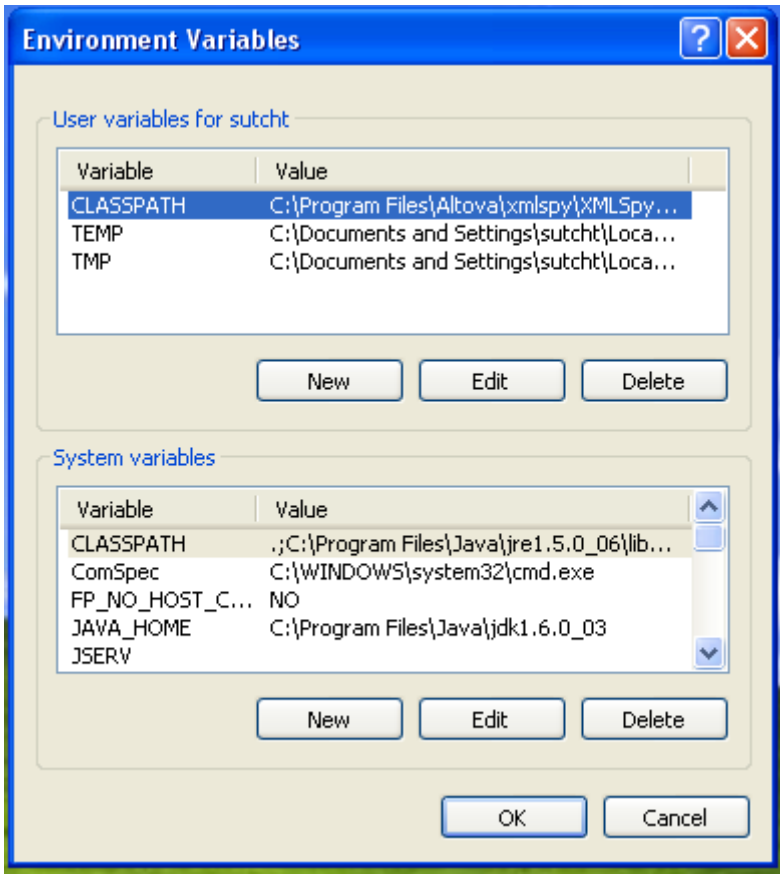



**Exhibit 11.** SDWIS/LabToState System Environment Variables

### 2.2.2.5 Set Memory Allocation

To increase Tomcat's memory allocation, you should use the Tomcat Monitoring Tool. By completing the following tasks, you increase the memory allocation for a Tomcat web server. If you use a different web server, you should refer to the documentation for that web server.

1. From the Windows Start Menu, click Programs, Apache Tomcat, Monitor Tomcat. These actions insert an Apache icon in the System Tray as indicated in Exhibit 12.

Draft


**Exhibit 12.** Apache Icon in the Desktop's System Tray

2. Right click the Apache icon and select Configure, which displays the Apache Tomcat Properties window as depicted in Exhibit 13.

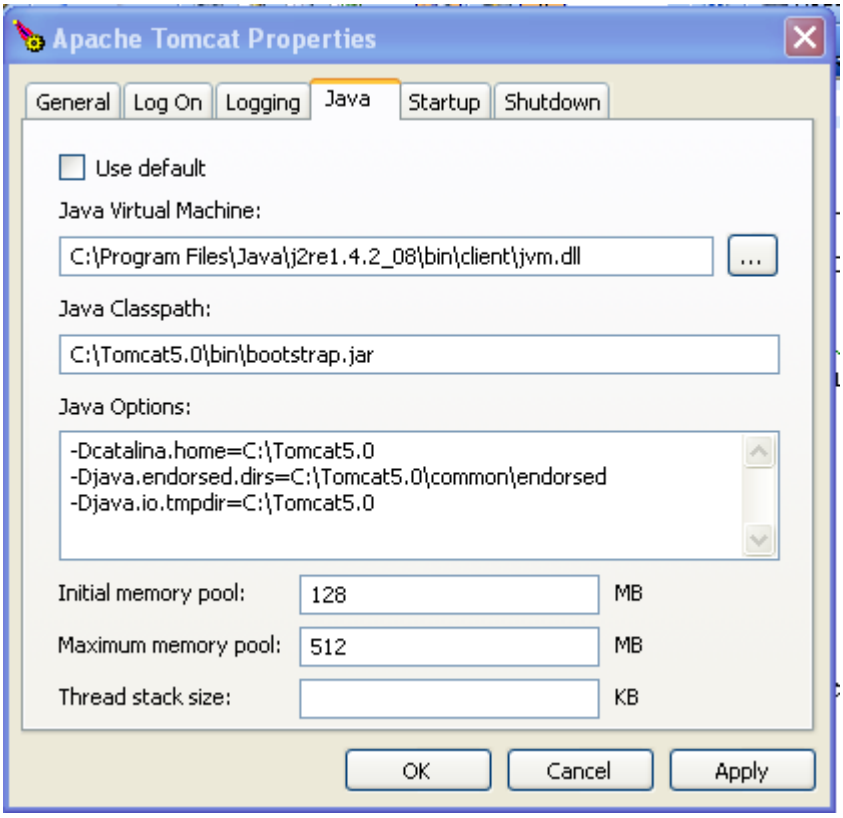



**Exhibit 13.** Tomcat Properties Window

3. Enter 128 for the initial memory pool. Enter 512 for the maximum memory pool. Click **OK**, which sets the memory allocation parameters.

4. Restart Tomcat, which allows the environment variables and memory parameters to be established.

#### 2.2.2.6 Configure and Start Up SDWIS/LabToState

After completing Steps 1 through 5, the installation of SDWIS/LabToState is complete. You should continue to Section 3.0, Site Customization Procedures of this document to configure the SDWIS/LabToState software for your site.

After configuring SDWIS/LabToState, you need to complete the following activities:

- Start the application server.
- Enter URL of the application (e.g., http://localhost:8080/labtoState/jsp).

## 3.0 SITE CUSTOMIZATION PROCEDURES

This section describes how to configure SDWIS/LabToState at your site. It describes the text files used to assign various installation and run-time parameters. It also describes procedures for constructing the data files for submission to SDWIS/LabToState.

### 3.1 Site Customization Files

To customize its deployment, SDWIS/LabToState uses several text files and XML documents. SDWIS/LabToState does not offer any tools to administer these files. You may use any standard text editor to modify or change the content. All of the various configuration files are located in the following directory:

Stand alone Client Application: *[Unzipped Folder]*\docBase\web-inf\classes\properties
Web-Based Application: *[AppServerHome]*\webapps\labtostate\web-inf\classes\properties

The following sections describe each configuration file.

#### 3.1.1 Site Properties File

The Site Properties File is a text file formatted as name/value pairs. The System Administrator changes values in this file to reflect the operational environment at a particular site. Appendix A lists the name, description, and initial setting of each name/value pair. Column 2 (Set By) of Appendix A indicates whether the value of the element is "Set By" the software, system administrator, or through delivery (the file is delivered with a value that should not be changed).

**Physical File Name:** siteProperties.txt

#### 3.1.2 User Identification XML Document

SDWIS/LabToState provides a default implementation of user authentication. It is, however, anticipated that most states will implement their own authentication utility. The default implementation is provided to deliver an "out of the box" solution for initial start up and testing. That default implementation uses an XML document to identify each User ID and assign to it a

password, role, PIN number, and laboratory. Exhibit 14 lists the contents of the XML document delivered with SDWIS/LabToState.

Each user is assigned one of two roles. A user assigned to the Lab role is granted access to the sample data submitted by that laboratory. A user assigned to the State role is granted access to all submissions regardless of the laboratory.

To authorize a new user, add the element tags named UserName, Password, Role, Laboratory, PIN, and Email to the XML document as depicted in Exhibit 14. A PIN Number and laboratory are not required if the user is assigned to the "State" role.

**Physical XML Document File Name:** userList.xml

```
<UserList>
        <User>
                <UserName>user1</UserName>
                <Password>user1</Password>
                <Role>Lab</Role>
                <Laboratory>Lab-1</Laboratory>
                <PIN>1234</PIN>
                <Email>xyz@xyz.com</Email>
        </User>
        <User>
                <UserName>user2</UserName>
                <Password>user2</Password>
                <Role>Lab</Role>
                <Laboratory>Lab-2</Laboratory>
                <PIN>1234</PIN>
                <Email>xyz@xyz.com</Email>
        </User>
        <User>
                <UserName>admin</UserName>
                <Password>admin</Password>
                <Role>State</Role>
                <Email>xyz@xyz.com</Email>
        </User>
</UserList>
```

**Exhibit 14.** User Authentication XML Document

### 3.1.3 Release Statement File

The Release Statement File is a text file containing the text presented to the user as he/she releases the XML document to SDWIS/XML Sampling. This text will be presented on the Certification page.

**Physical File Name:** certificationStatement.txt

### 3.2 Data File Content Configuration

SDWIS/LabToState provides a Data-Source Plug-In to receive sample data formatted as comma separated values (CSV). The site configures the content and order of the sample data contained in the CSV file. First, the site identifies logical groupings of sample data referred to as logical objects. Each logical object is based on one of the following SDWIS/XML Sampling structure sets:

- Sample
- Sample Measure
- Result
- Sample Result Measure
- Sample Summary
- Sample Summary Result
- MDBP Summary

The Object Configuration File is a text file that maps logical objects to one and only one structure set. Each structure set, however, may be mapped to multiple logical objects. For example, the site may identify a logical object called TCRSample and map it to the Sample structure set. The site may also identify a logical object called GeneralSample and map it to the same Sample structure set. Even though both logical objects (TCRSample and GeneralSample) map to the same Sample structure set, they may contain different data in a different order. Exhibit 15 lists the content of the Object Configuration File delivered with SDWIS/LabToState.

**Physical File Name**: master.txt**.**

```
# This file contains the mapping between the object ordered elements file
# and the structure set names
# Valid structure set names are :
# {Sample,SampleMeasure,Result,ResultMeasure,MDBPSummary,SampleSummary,SampleSummaryResult}

sample=Sample
result=SampleResult
sampleMeasure=SampleMeasure
sampleResultMeasure=SampleResultMeasure
sampleSummary=SampleSummary
sampleSummaryResult=SampleSummaryResult
mdbpSummary=MDBPSummary
```

**Exhibit 15.** Object Configuration File

For each logical object identified in the Object Configuration File, the Element Configuration File lists the elements and their order in the CSV file. The element names are listed in the order in which they are submitted in the CSV file. The name of the element is the Staging Table Column Name as specified in the structure set documentation included in the *SDWIS/XML Sampling Mapping Document and Accompanying Schemas Document* (Systalex-SDWIS-01-10.d1b, July 3, 2007). The Element Configuration File is a text file and the site creates a

separate Element Configuration File for each logical object. Using the above example in which the site configured both a TCRSample and a GeneralSample logical object, the site must also configure two Element Configuration Files named TCRSample.txt and GeneralSample.txt. Exhibit 16 lists the Element Configuration Files delivered with SDWIS/LabToState. Appendix B contains a list of the content of each Object and Element Configuration file delivered with SDWIS/LabToState. The physical name of the Element Configuration File is *[logical object]*.txt. Appendix B also includes a zip file containing examples of CSV files that may be run using the default configuration files.

| Element Configuration Files |
|---|
| sample.txt |
| result.txt |
| mdbpSummary.txt |
| sampleSummary.txt |
| sampleSummaryResult.txt |
| sampleMeasure.txt |
| sampleResultMeasure.txt |

**Exhibit 16.** Element Configuration Files

**Physical File Name:** *[logicalobject]*.txt

To configure your own CSV file content and order, you assign a logical name, such as TCRSample, to the data you want to submit. You register the logical name by adding it to the Object Configuration file and assigning it to one of the SDWIS/XML Sampling Structure Sets, listed in Exhibit 17. Then, you create a text file containing the list of elements names representing the data to be submitted with that logical object. Appendix C contains a Microsoft Excel spreadsheet that lists the element names by business object. When you create the CSV file, the first column of the line of text is the name of the object followed by the data values for each of the elements named in the Element Configuration file.

| SDWIS/XML Sampling Structure Sets |
|---|
| Sample |
| SampleResult |
| MDBPSummary |
| SampleSummary |
| SampleSummaryResult |
| SampleMeasure |
| SampleResultMeasure |

**Exhibit 17.** SDWIS/XML Sampling Structure Sets

Exhibit 18 depicts an example of a CSV file that conforms to the default configuration delivered with SDWIS/LabToState. The first column in the CSV file is always the name of the logical object being submitted. The remainder of the line is the sample data in the order specified in the

Element Configuration File for that logical object. Also, if any alphanumeric data contains spaces, the alphanumeric data must be enclosed in quotation marks as depicted in the first line of Exhibit 18. If the alphanumeric data does not contain spaces, the quotation marks are optional as depicted in the remaining lines of Exhibit 18.

```
"sample","TEST1","statesample","IL0750100","Y","IL17549","ID1","DISTRIBUTION","10001-02","501 ST CHARLES ST   ",...[more data]
result,TEST1,01/01/2000,xx,23,nameX,0100,01/01/2000,080000,01/01/2000,080000,01/01/2000,01/01/2000,A,xx,xx,,...[more data]
sampleMeasure,TEST1,,,,nameX,1.23,ul
sampleResultMeasure,TEST1,,,,0100,,,bname,1,mg
sampleSummary,x,1,2,3,N,2/10/1976,2/10/2006,300,2/10/1976,2/10/1976,2/10/1976,"my, comments"
sampleSummaryResult,x,1,2,Y,01/01/2006,01/01/2006,RT,1,1,mg
mdbpSummary,A,xyz,abc,1,2/10/1976,2/10/2006,x,1,1,1,mohammed,fazal,saic,4-082,2/10/1976,10,8,Y,5,50,,,,,...[more data]
```

**Exhibit 18.** Example of CSV File

### 3.3 Data Source Registration File

The Data Registration File is a text file that maps the logical name of each Data-Source Plug-In to the name of the Java class implementing the functionality. The file is formatted as name value pairs. The name is the logical name of the Data-Source Plug-In and is the value listed in the drop-down list on the Upload And Validate Page from which the user selects the data format being uploaded. The logical name is mapped to the Java class implementing the data source specification.

To register your own data source specifications, you must identify a logical name and the Java class implementing your specification. The example depicted in Exhibit 19 registers a new Data-Source Plug-In with a logical name of “NewUserPlugIn.” If this data source is selected from the Upload and Validate page, SDWIS/LabToState invokes the “com.sdwis.labtostate.generator.NewUserPlugIn”. The registered Java class must be copied to the appropriate library in the hierarchy structure. For this example, that directory is [AppServerHome]\webapps\labtostate\web-inf\classes.

The Data-Source Plug-Ins delivered with SDWIS/LabToState and those developed by the site must adhere to the interface specifications documented in Appendix B.

**Physical File Name:** plugins.txt

```
# This file contains the generator plugins to class name mappings

ODBC-SQL=com.sdwis.labtostate.generator.SQLPlugin
CSV-File=com.sdwis.labtostate.generator.CSVPlugin
SDWIS-EDWR-XML=com.sdwis.labtostate.validator.EDWRValidator
MDBP-Summary-XML=com.sdwis.labtostate.validator.MDBPValidator
Sample-Summary-XML=com.sdwis.labtostate.validator.SummaryValidator
Sample-Summary-Result-XML=com.sdwis.labtostate.validator.SummaryResultValidator


#Example of User-Defined Plug-in
NewUserPlugIn=com.sdwis.labtostate.generator.NewUserPlugIn
```

**Exhibit 19.** Data-Source Plug-In Registration File

## 4.0 CROMERR IMPLEMENTATION

To support CROMERR, SDWIS/LabToState 2.0 implemented the requirements listed in the *SDWIS/LabToState 2.0 Requirements and Design Document* (Guident-SAIC-SDWIS-8-d1b, December 19, 2007). In addition to those software requirements, many operational requirements must be implemented at your site. To assist you with attaining CROMERR approval at your site, Appendix E contains a completed CROMERR checklist describing the software requirements and suggesting operational procedures. This CROMERR checklist was approved by EPA's Technical Review Committee (TRC) in August, 2007. The provided CROMERR Checklist is a template, which your primacy agency may use to obtain approval at your site. It is the responsibility of your primacy agency to review each item in the checklist for compliance with the described implementation or update the checklist describing implementation procedures at your site.

Exhibit 20 provides a brief description by checklist item number of SDWIS/LabToState's implementation. This description is intended as an overview of SDWIS/LabToState's CROMERR implementation; it is not intended as a replacement for reviewing the checklist included in Appendix E.

**Exhibit 20.** SDWIS/LabToState CROMERR Implementation Overview

| CROMERR Checklist Section | Item Numbers | Summary Implementation Description |
|---|---|---|
| Registration Section<br>Signature Process Section<br>(e-signature cases only) | 1 through 7 | If your site requires a digital signature for submissions of samples, you must complete the Registration and Signature Process Sections of the checklist. |
| Submission Process<br>(Transmission Error Chuckling) | 8 | If your primacy agency does not use SSL, you must establish your own business practices to support transmission error checking and describe those practices in the checklist. |

Draft

**Exhibit 20.** SDWIS/LabToState CROMERR Implementation Overview

| CROMERR Checklist Section | Item Numbers | Summary Implementation Description |
|---|---|---|
| Submission Process (Opportunity to Review Copy of Record) | 9 | SDWIS/LabToState automatically sends e-mails when the Copy of Record is generated to the e-mail address registered with the user account as well as the e-mail addresses identified during the approval process.<br><br>SDWIS/LabToState renders the XML documents using an XSLT stylesheet provided with the application. If your primacy agency wants a different presentation, you can create your own style sheet and configure the application to use it.<br><br>Although SDWIS/LabToState provides a download capability, your primacy agency must establish a retention period, which is the number of days the Copy of Record is available using SDWIS/LabToState. After the retention period has lapsed, your primacy agency must establish procedures to archive the Copy of Record and allow the laboratory to request and receive it. |
| Submission Process (Submitter/Signatory Repudiation of Copy of Record) | 10 | If your primacy agency allows changes to sample data, you must describe those procedures in your supplemental CROMERR checklist. |
| Submission Process (Flag Accidental Submissions) | 11 | SDWIS/LabToState allows the laboratory to review data and remove it before it is processed by SDWIS/STATE. |
| Automatic Acknowledgement of Submission | 12 | If your site requires a digital signature for submissions of samples, you must complete the Registration and Signature Process Sections of the checklist. |
| Signature Validation | 13 through 17 | If your site requires a digital signature for submissions of samples, you must complete the Registration and Signature Process Sections of the checklist. |
| Copy of Record (True and Correct Copy of Record Received) | 18 | SDWIS/LabToState generates, encrypts, and logs a digital hash of the submitted files, and the generated XML documents. The XML documents are displayed using XSLT stylesheets, which may be customized to your site. |

Draft

**Exhibit 20.** SDWIS/LabToState CROMERR Implementation Overview

| CROMERR Checklist Section | Item Numbers | Summary Implementation Description |
|---|---|---|
| Copy of Record (Timely Availability of Copy of Record Received) | 19 | SDWIS/LabToState sends an e-mail to the laboratory indicating the submitted data is ready for review and approval. After approval, the laboratory may view and/or download the Copy of Record. |
| Copy of Record (Maintenance of Copy of Record) | 20 | The primacy agency establishes a retention period, which specifies the time (e.g., number of days) the Copy of Record is available through SDWIS/LabToState. After the retention period, each primacy agency must establish and describe archival procedures consistent with its operational environment. |

This page is intentionally left blank

# APPENDIX A

Site Property File Name Descriptions

This page is intentionally left blank.

Draft

| SDWIS/LabToState Site Property Names and Description | | | |
|---|---|---|---|
| **Name** | **Description** | **Initial Setting** | **Set By** |
| TurnOnEmail | Contains an indicator allowing a site to eliminate the delivery of e-mail messages after upload and validation. The permitted values for this property are “true” and “false.” | true | Email Administrator |
| EmailHost | Contains the IP Address of the SMTP (Simple Mail Transfer Protocol) Mail Server. This value is used to connect to the mail server in automate the delivery of e-mails. This value must be set during installation. | sdc-ex2 | Email Administrator |
| EmailPort | Contains the port number of the e-mail server. The default value of 25 is used for a standard e-mail server installation. For non-standard e-mail server installations, this value should be changed to the port number of the e-mail server. This value is set during installation and can be changed by the System Administrator. | 25 | Email Administtrator |
| EmailAuthenticationRequired | Contains an indicator allowing a site to require e-mail authentication. The permitted values for this property are “true” and “false.” This value is set during installation and can be changed by the System Administrator. | false | Email Administrator |
| EmailUserid | Contains the user name that must be supplied if the e-mail server requires authentication. If e-mail server does not require authentication, the value is set to null. This value is set during installation and can be changed by the System Administrator. | <no value> | Email Administrator |
| EmailPassword | Contains the Password that must be supplied if the e-mail server requires authentication. If e-mail server does not require authentication, the value is set to null. This value is set during installation and can be changed by the System Administrator. | <no value> | Email Administrator |

Draft

| SDWIS/LabToState Site Property Names and Description | | | |
|---|---|---|---|
| **Name** | **Description** | **Initial Setting** | **Set By** |
| EmailFromAddress | Contains the e-mail address for the SDWIS/LabToState administrator. This e-mail address is used to send e-mails. | LabToState | Email Administrator |
| DateFormat | Contains a mask indicating the format in which date values are submitted. The default mask is MM/dd/yyyy. | mm/dd/yyyy | System Administrator |
| TimeFormat | Contains a mask indicating the format in which time values are submitted. The default mask is hhmmss. | HHmmss | System Administrator |
| EdwrXSL | Contains the name of the stylesheet used to provide the EDWR XML document in a human readable format. | edwr_view.xsl | System Administrator |
| SummaryResultXSL | Contains the name of the stylesheet used to provide the Summary Result document in a human readable format. | summaryResult_view.xsl | System Administrator |
| SummaryXSL | Contains the name of the stylesheet used to provide the Summary XML document in a human readable format. | summary_view.xsl | System Administrator |
| MdbpXSL | Contains the name of the stylesheet used to provide the MDBP XML document in a human readable format. | mdbp_view.xsl | System Administrator |
| HostProtocol | Contains the protocol portion used by the web server hosting the application. It is used to generate the URL identified in e-mails sent to the users. | http | System Administrator |
| HostName | Contains the host name of web server hosting the application. It is used to generate the URL identified in e-mails sent to the users. | localhost | System Administrator |
| HostPort | Contains the port number of web server hosting the application. It is used to generate the URL identified in e-mails sent to the users. | 8080 | System Administrator |
| HostContext | Contains the context of the LabToState application deployed on the web server. It is used to generate the URL identified in e-mails sent to the users. | Labtostate | System Administrator |

Draft

| SDWIS/LabToState Site Property Names and Description | | | |
|---|---|---|---|
| **Name** | **Description** | **Initial Setting** | **Set By** |
| DatabaseDriverClassName | Contains the JDBC-ODBC bridge class name, which connects to the client's database using the ODBC data source. This value must not be changed. If the site is running on UNIX, this value can be changed to the jdbc driver class name (Refer to your local database manual for "connecting using jdbc").<br><br>sun.jdbc.odbc.JdbcOdbcDriver | sun.jdbc.odbc.JdbcOdbcDriver | This value can be overriden by the Database Administrator |
| DatabaseConnectionURL | Contains the string used to establish a database connection The default DSN name is set to SDWIS. If the site is using a different DSN then this value must be changed. If the site is running on UNIX, this value can be changed to the jdbc connection url (Refer to your local database manual for "connecting using jdbc").<br><br>jdbc:odbc:SDWIS | jdbc:odbc:SDWIS | This value can be overriden by the Database Administrator |
| DatabaseUserID | Contains the User ID used to connect to the client's database schema. This value must be set during installation if the site is using the SQL Plug-in. | <no value> | Database Administrator |
| DatabasePassword | Contains the Password used to connect to the client's database schema. This value must be set during installation if the site is using the SQL Plug-in. | <no value> | Database Administrator |
| PasswordAuthenticatorClass | Contains the name of the Java class used to authenticate User IDs and Passwords. If the site implements its own authentication utilities, the implementation must follow the interface specifications documented in the SDWIS/LabToState Installation Guide. SDWIS/LabToState includes a default implementation of user authentication in the Java class com.sdwis.labtostate.utility.DefaultPasswordAuthenticator. | com.sdwis.labtostate.utility.DefaultPasswordAuthenticator | This value can be overriden by the System Administrator |

Draft

| SDWIS/LabToState Site Property Names and Description | | | |
|---|---|---|---|
| **Name** | **Description** | **Initial Setting** | **Set By** |
| PINAuthenticatorClass | Contains the name of the Java class used to verify a User's PIN Number. If the site implements its own PIN Number verification utilities, the implementation must follow the interface specifications documented in the SDWIS/LabToState Installation Guide. SDWIS/LabToState includes a default implementation of user PIN Number verification in the Java class com.sdwis.labtostate.utility.DefaultPINAuthenticator. | com.sdwis.labtostate.utility.DefaultPINAuthenticator | This value can be overriden by the System Administrator |
| NodeURL | Contains the URL to the State Node offering the web services defined by the WSDL document. This parameter must be used when using the State Node for integrating with SDWIS/XMLSampling. | http://localhost:10000/DNC/services/NetworkNodePortType_V10 | Node Administrator |
| NodeUserId | Contains the UserID required to connect to the State Node. This parameter must be used when using the State Node for integrating with SDWIS/XMLSampling. | <no value> | Node Administrator |
| NodePassword | Contains the password used to connect to the State Node. This parameter must be used when using the State Node for integrating with SDWIS/XMLSampling. | <no value> | Node Administrator |
| DataFlowNameForWebService | This value will be used as the dataflow name when invoking the web service. This parameter must be used when using the State Node for integrating with SDWIS/XMLSampling. | XMLSampling | Node Administrator |

Draft

| SDWIS/LabToState Site Property Names and Description | | | |
|---|---|---|---|
| **Name** | **Description** | **Initial Setting** | **Set By** |
| XMLSamplingIntegratorClass | Contains the name of the Java class that delivers the XML documents to the State Node, which in turn delivers the XML document to SDWIS/XMLSampling. This name/value pair and the name/value pairs that identify the SDWIS/XML Sampling Inbox are mutually exclusive. Therefore, for delivery, this name/value pair is commented. To implement this feature, you remove the comment from this line and comment the lines associated with the SDWIS/XML Sampling Inbox (SDWIS_EDWR_Inbox, Sample_Summary_Inbox, Sample_Summary_Result_Inbox, and MDBP_Summary_Inbox). | com.sdwis.labtostate.utility.XMLSamplingFileSystemIntegrator | System Administrator |
| ZIPArchivorClass | Contains the name of the Java class that provides archival support of the submitted files, generated files, and the digital hash. This implementation zips all of the files and saves the zip file specified in the ZIP_File_Folder name/value combination. | com.sdwis.labtostate.utility.ZIPFileFileSystemIntegrator | System Administrator |
| ZIP_File_Folder | Contains the full path and folder name of the directory to which the zip file is archived by the ZIPArchivorClass. | C:\\My Downloads | System Administrator |
| XML_Sampling_Inbox | Contains the full path and folder name of the SDWIS SDWIS/XML Sampling Inbox. This name/value pair and the name/value pair (XMLSamplingIntegratorClass) that identifies the Java Class for the State Node submission are mutually exclusive. Therefore, if you implement the State Node approach to SDWIS/XML Sampling integration, you must comment this line and remove the comment from the XMLSamplingIntegratorClass line. | C:\\My Downloads\\XMLSamplingInbox | System Administrator |

This page is intentionally left blank.

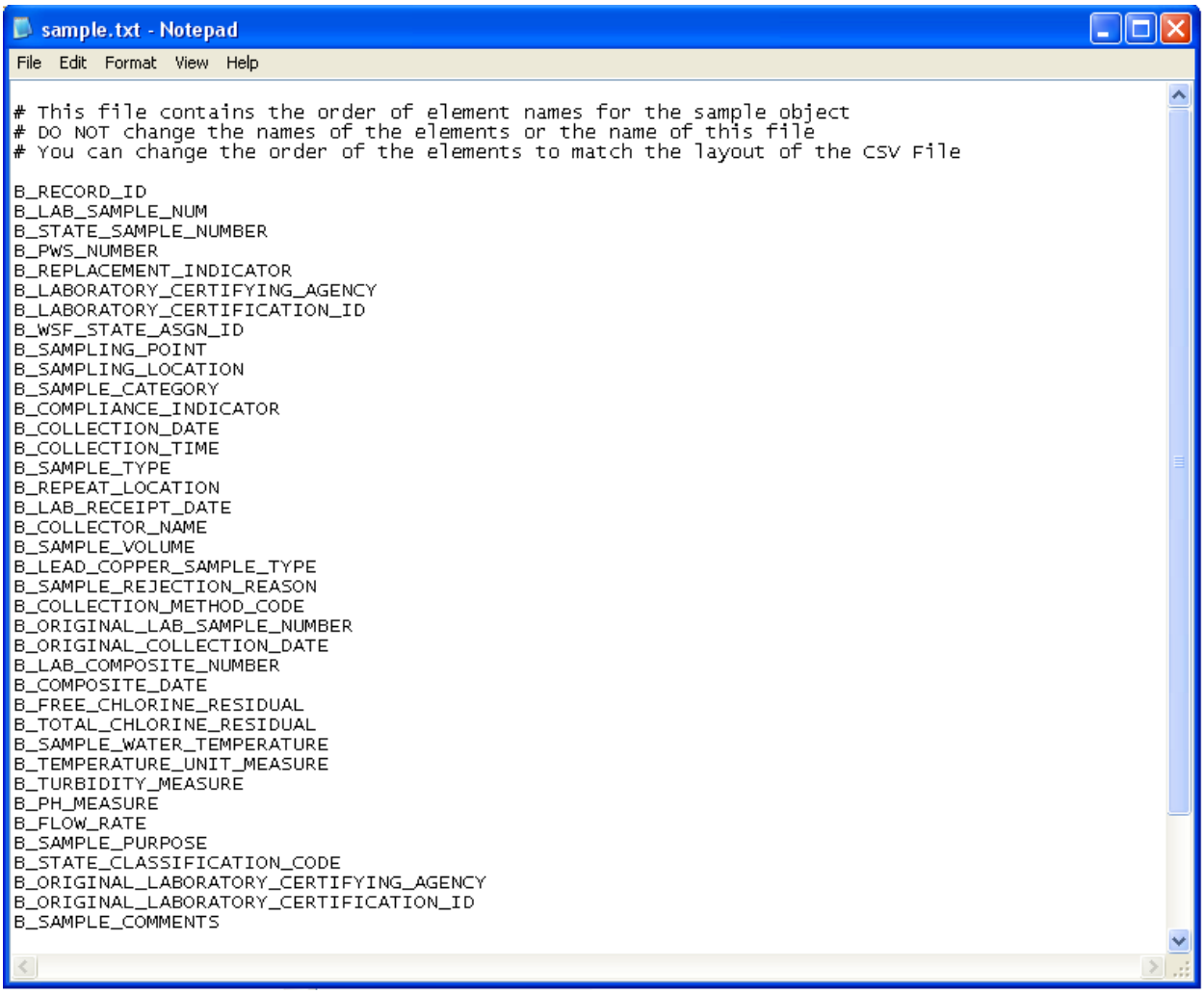
# APPENDIX B

Object and Element Configuration Files

This page is intentionally left blank.

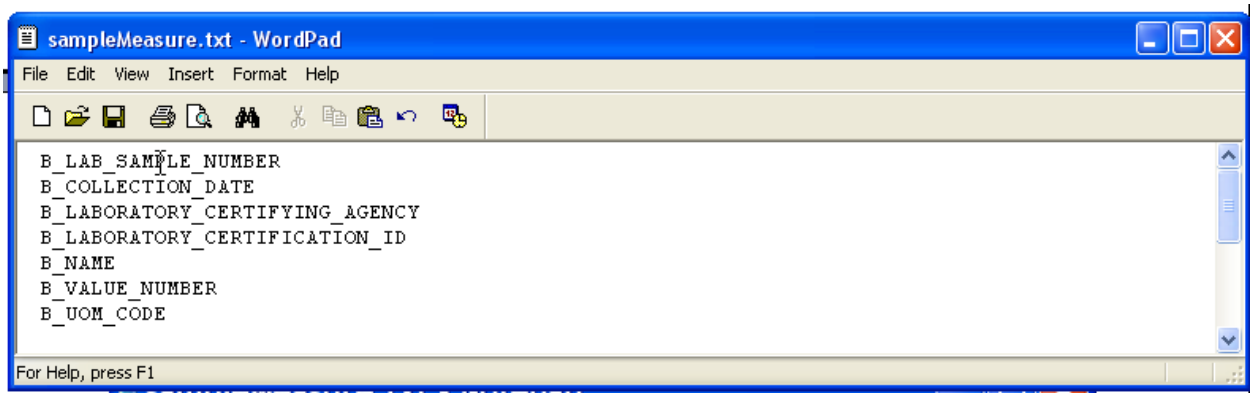
Draft

# Sample Object



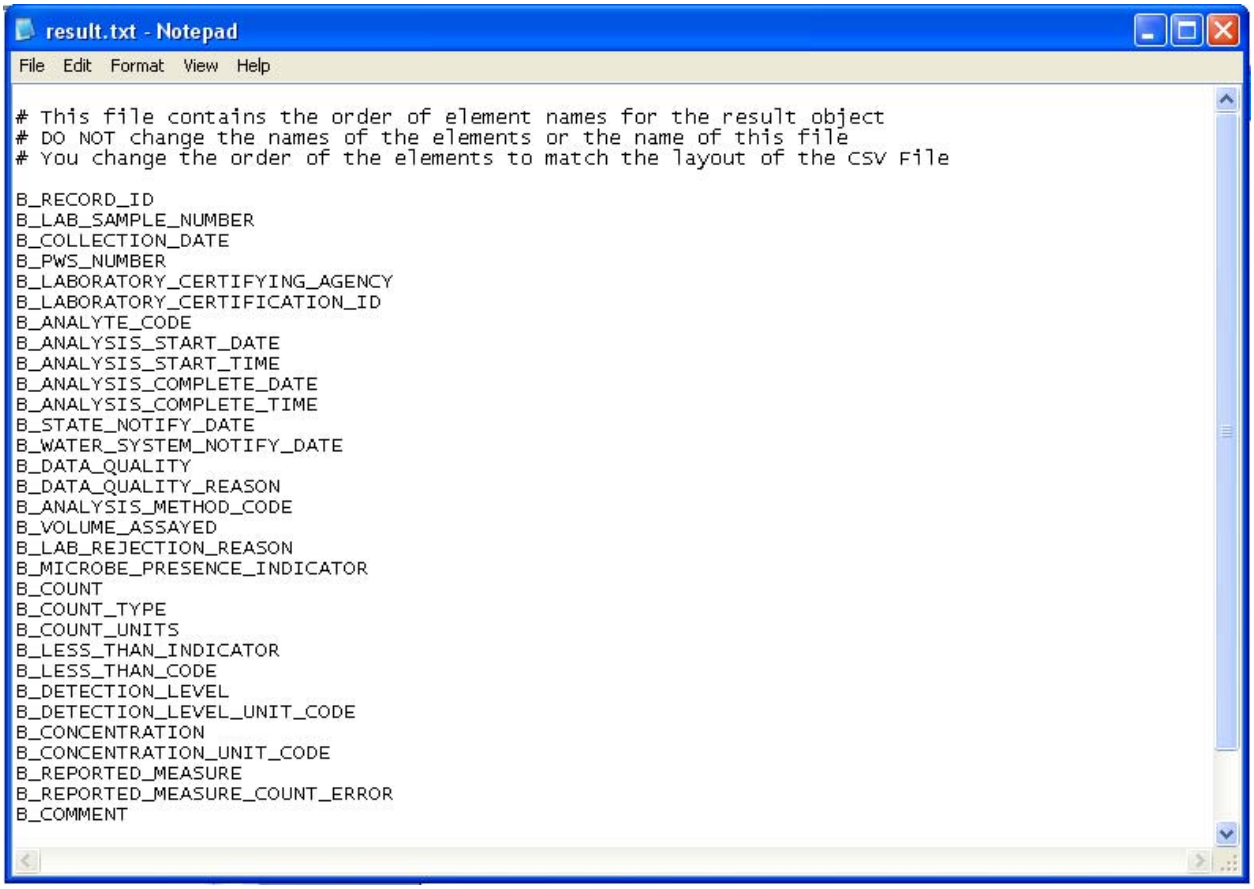
Draft

# Sample Measure Object



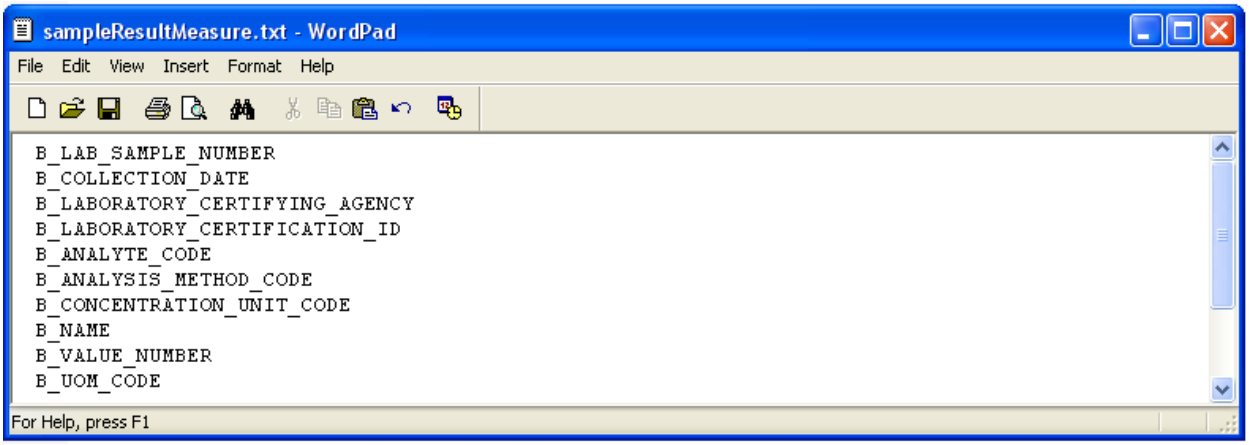
Draft

# Result Object



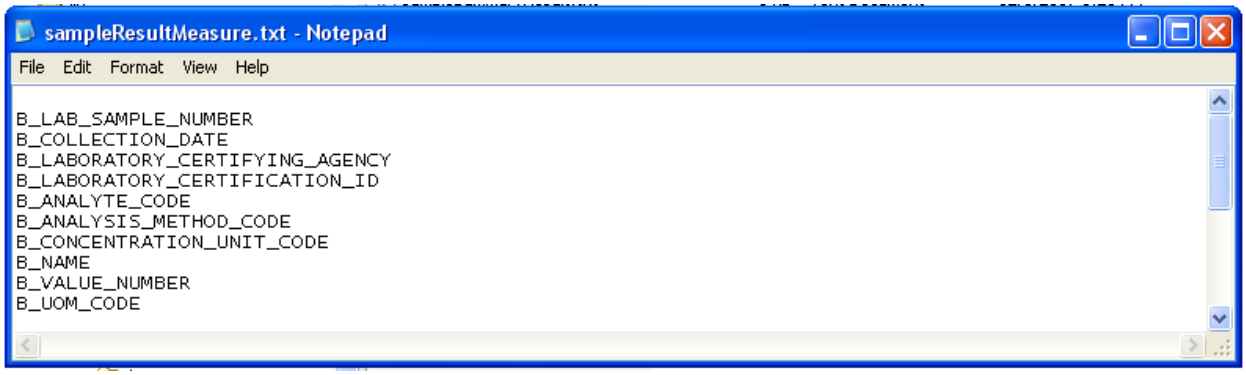
Draft

# Sample Result Measure Object



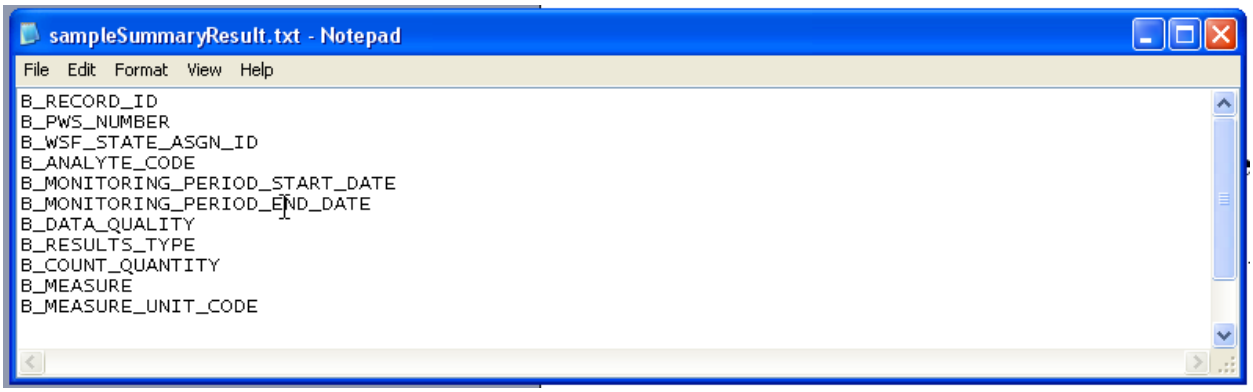
Draft

# Sample Summary Object



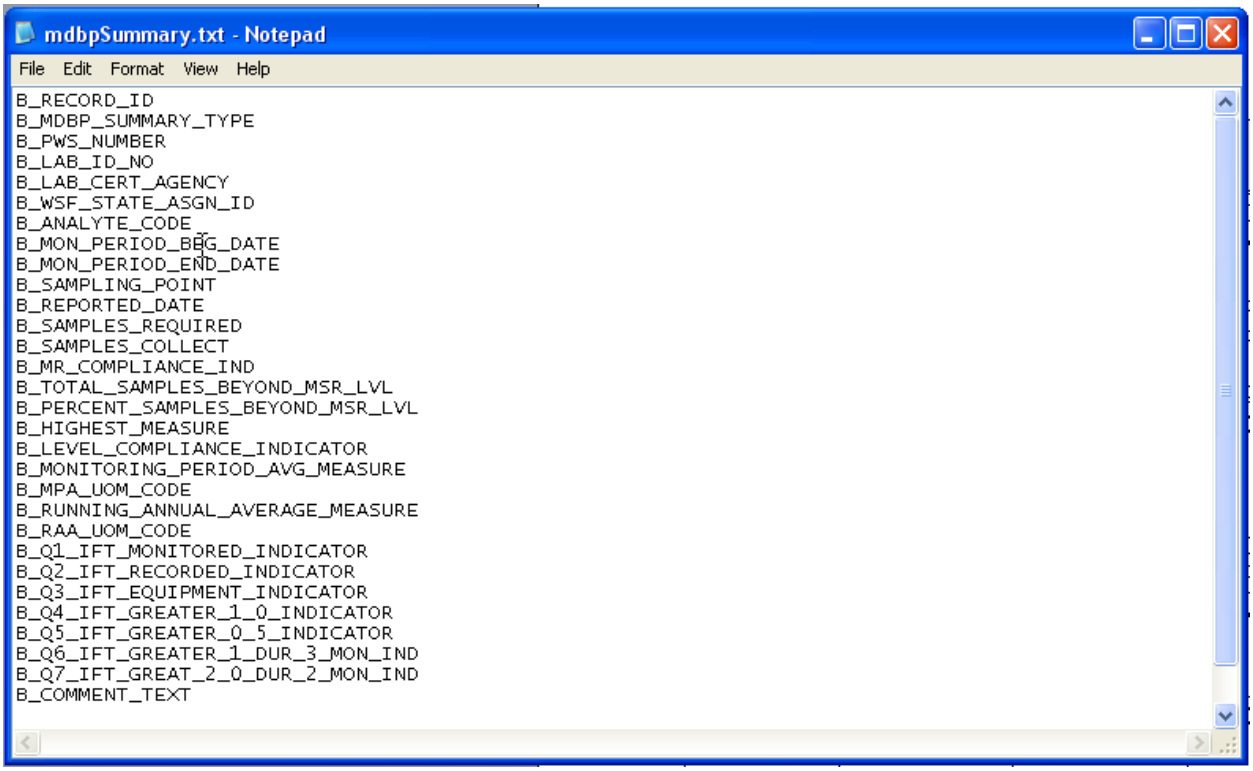
Draft

# Sample Summary Result Object



Draft

# MDBP Summary Object



Draft

The embedded zip file contains examples of CSV files, the error reports, XML documents, and XML Document views generated by SDWIS/LabToState based on the CSV files.

**Click on the icon below to extract files form the embedded zip file for Appendix B**

**Appendix B**


csvZips.zip

# APPENDIX C

Element Names by Object

This page is intentionally left blank.

Draft

**Click on the icon below to use viewing the electronic file for Appendix C**

LabToStateElementNames.xls

This page is intentionally left blank.

# APPENDIX D

Interface Specifications

This page is intentionally left blank.

Draft

# Data-Source Plug-Ins Interface Specifications

To implement a Data-Source Plug-In, the implementer will complete the following steps to create, register, and integrate it with SDWIS/LabToState.

**STEP 1: Develop a site-specific Data-Source Plug-In**

The Java class that implements the site-specific Data-Source Plug-In extends an abstract class delivered with the standard SDWIS/LabToState software. The abstract class is com.sdwis.labtostate.generator.Plugin. The new Java class implements one method with the following signature:

public abstract void execute(UIStructure structure) throws Exception;

The argument passed to this method is the UIStructure class, which contains all the data entered by the user in the upload and validate page. It is the responsibility of the concrete data source plug-in class to create the DataObject class for each logical business object for which it is receiving data. The DataObject class contains an attribute command which holds the name of the logical business object and a string array containing the data values in the order specified by the Element Configuration File. The abstract class provides a pointer to the handler, which the custom class uses to invoke the XML Generation tool of the SDWIS/LabToState application. After the new Java class creates the DataObject, it invoked the handler's processObject method denoted by the following method signature:

abstract public void processObject(DataObject object) throws Exception;

After creating and processing all the DataObjects, the new Java class creates a special DataObject containing the "Marshall" command. This command notifies the handlers to create the XML documents from the in-memory representation of the DataObjects and to invoke the XML Validation tool to check the XML documents against the XML Schemas.

**STEP 2: Copy the site-specific Data-Source Plug-In**

The user copies the compiled java classes under the WEB-INF/classes folder or creates a jar file from the compiled classes and copies the jar file to the WEB-INF/lib folder. This step makes the site-specific Data-Source Plug-In available to the SDWIS/LabToState application allowing it to dynamically invoke site-specific implementations.

**STEP 3: Register the site-specific Data-Source Plug-In**

To register the Data-Source Plug-In with SDWIS/LabToState, add a new entry to the plugins.txt file in the properties folder. The new entry contains a user-friendly logical name and the fully qualified java class name of the custom data source implementation. The user-friendly name is listed as a new option under the file format in the upload and validate page.

Draft

## User Authentication Interface Specifications

To implement a site-specific User Authentication routine, you will complete the following steps to create, register, and integrate it with SDWIS/LabToState.

### STEP 1: Create the site-specific User Authentication implementation

The custom user authentication implementation class implements the com.sdwis.labtostate.utility.PasswordAuthenticator interface class. The concrete class implements one method with the following signature:

public boolean authenticate(String userName, String password) throws Exception;

The arguments passed to this method contain the username and password entered by the user in the login page. The implementation class can validate this information using its own logic by accessing the local user repository (eg., Windows NT, LDAP etc). The return type from this method is a boolean indicating whether the user was successfully authenticated.

### STEP 2: Copy the site-specific User Authentication implementation

Copy the compiled java classes under the WEB-INF/classes folder or create a jar file from the compiled classes and copies the jar file to the WEB-INF/lib folder. This step makes the site-specific User Authentication implementation available to the SDWIS/LabToState application allowing it to dynamically invoke site-specific implementations.

### STEP 3: Register the site-specific User Authentication implementation

You update the siteProperties.txt file in the properties folder to change the value of the PasswordAuthenticatorClass property to contain the name of the newly created custom class. The value must be the fully qualified java class name of the custom implementation class.

You change the value of the PasswordAuthenticatorClass property in the siteProperties.txt file in the properties folder. The value contains the name of the newly created custom class. The value must be the fully qualified java class name of the custom implementation class.

Draft

## PIN Authentication Interface Specifications

To implement a site-specific PIN Authentication routine, you complete the following steps to create, register, and integrate it with SDWIS/LabToState.

### STEP 1: Create the site-specific PIN authentication implementation

The custom PIN authentication implementation class implements the com.sdwis.labtostate.utility.PasswordAuthenticator interface class. The concrete class implements one method with the following signature:

public boolean validatePIN(String userName, String laboratory, String PIN) throws Exception;

The arguments passed to this method contain the username, laboratory, and the PIN entered by the user on the certification page. The implementation class validates this information using its own logic to verify the PIN number for a specific user. The return type from this method is a boolean indicating whether the PIN was successfully verified.

### STEP 2: Copy the site-specific PIN Authentication implementation

Copy the compiled java classes under the WEB-INF/classes folder or create a jar file from the compiled classes and copies the jar file to the WEB-INF/lib folder. This step will make the site specific PIN Authentication available to SDWIS/LabToState application allowing it to dynamically invoke site-specific implementation.

### STEP 3: Register the custom PIN authentication class to the LabToState application

You change the value of the PINAuthenticatorClass property in the siteProperties.txt file in the properties folder. The value contains the name of the newly created custom class. The value must be the fully qualified java class name of the custom implementation class.

# APPENDIX E

SDWIS/LabToState CROMERR Checklist

This page is intentionally left blank.

Draft

**Click on the icon below to view the electronic file for Appendix E**

LabToState Checklist
Release 2.0.doc

# APPENDIX F

SDWIS/LabToState Example Files

This page is intentionally left blank.

# Draft

The embedded zip file contains examples of CSV files, the error reports, XML documents, and XML Document views generated by SDWIS/LabToState based on the CSV files.

LabToStateExampleFiles.zip