SDWIS/LabToState 3.0 Installation Guide and Release Notes

United States Environmental
Protection Agency

Office of Ground Water and
Drinking Water

Contract No.
GS-35F-4461G
SDWIS Project
Product Control No.
SAIC-SDWIS-5.4d2c

January 14, 2011

SDWIS/LABTOSTATE 3.0
INSTALLATION GUIDE
AND
RELEASE NOTES

CONTRACT NO. GS-35F-4461G
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
8301 Greensboro Drive
McLean, VA 22102

CONTENTS

EXHIBITS

1.0 OVERVIEW

To electronically submit drinking water sample results to SDWIS/XML Sampling for processing and permanent storage in a SDWIS/STATE 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. SDWIS/LabToState 3.0 targets the following schemas:

- SDWIS_eDWR_v3.0.xsd
- SDWIS_Summary_v3.0.xsd
- SDWIS_SummaryResult_v3.0.xsd
- SDWIS_MDBPSummary_v3.0.xsd

The *SDWIS/LabToState 3.0 Installation Guide and Release Notes* contains instructions for installing SDWIS/LabToState 3.0. It comprises the following sections:

- Section 1.0, Overview, describes the content of the document, the software and hardware test environment, and user support procedures.

- Section 2.0, Release Notes, describes the acquisition method and delivery package of the product release. It also contains a section on advisories and additional items of importance about the release.

- Section 3.0, Installation Guide, describes the steps required to install SDWIS/LabToState 3.0 as a stand-alone client application and as a web-based application using a J2EE application server.

- Section 4.0, Functional Changes, describes the enhancements implemented in this release.

- Section 5.0, Resolved System Incident Reports (SIR), describes the defects corrected in this release.

- Section 6.0, Known Discrepancies, describes unresolved issues of this release.

- Section 7.0, CROMERR Implementation, describes the SDWIS/LabToState's aid to implementation of the CROMERR requirements.

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

1.1 Prerequisites

SDWIS/LabToState 3.0 requires the supporting software components and 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, SQL Server, and Microsoft Access.

Exhibit 1. SDWIS/LabtoState 3.0 Test Environment

Platform	Operating System	Software	Hardware
Application Server/Web Server	Windows 2003 Server with Service Pack 1	Windows XP Java 2 SDK, Standard Edition (J2SE), v1.6 with Oracle and SQL Server JDBC drivers Tomcat 6.0.20	Virtual Hardware using VMWare Infrastructure 3 on Dell 2950 1 virtual CPU 2 GB virtual RAM Virtual NIC Virtual hard drive with 20 GB
Database Server	Windows 2003 Server with Service Pack 1	Oracle 11g Enterprise Edition Microsoft SQLServer 2005	Virtual Hardware using VMWare Infrastructure 3 on Dell 2950 1 virtual CPU 2 GB virtual RAM Virtual NIC Virtual hard drive with 20 GB
E-Mail Server	Windows 2003 Server	MS Exchange 2000	Compaq Proliant ML370 G2, two 1.3 MHz CPU, 2 MB of memory

Exhibit 1. SDWIS/LabtoState 3.0 Test Environment

Platform	Operating System	Software	Hardware
Client Workstation	Windows XP Professional Version 2002 with Service Pack 3	Microsoft Internet Explorer v8.0	Dell OptiPlex GX620 Pentium(R) D CPU 2.80GHz 1 GB RAM Broadcom Net Xtreme 57xx Gigabit Controller Screen Resolution: 1024 x 768 and Small Fonts

1.2 User Support

Should you require additional support beyond this guide, you may direct questions or requests for further assistance to SDWIS User Support at (703) 676-4880 or e-mail **sdwis@saic.com**. A SDWIS team member answers hotline calls and e-mails between 9 a.m. and 5 p.m., Eastern Time on weekdays (except for federal holidays). During evenings, weekends, or times when SDWIS User Support is speaking with another SDWIS customer, you can leave a detailed message. Questions requiring the expertise of other team members (e.g., developers, subject matter experts) will be answered as soon as possible by the appropriate team member.

For program-wide support of SDWIS products, contact Mr. Clint Lemmons, EPA/OGWDW, at the following address:

Clint Lemmons
USEPA
1200 Pennsylvania Avenue, NW
Washington, DC 20460
Phone: (202) 564-4623
E-mail: lemmons.clint@epamail.epa.gov

Contact SDWIS User Support:

E-mail:sdwis@saic.com

Phone: 703-676-4880

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. SDWIS User Support is intended as technical support for the operation of SDWIS products such as SDWIS/LabToState.

2.0 RELEASE NOTES

2.1 Download Instructions

1. Obtain the LabToState3.0.zip file listed for SDWIS/LabtoState 3.0 from ASDWA's FTP site or SDWIS User Support:

2. Unzip the LabToState3.0.zip file to a folder of your choosing.

- LabToState3.0.zip – All files needed for SDWIS/LabToState 3.0. The .zip uncompresses to the following files:
    - labtostate.war
    - labtostate-embedded.zip

- LabToState_3.0_Doc.zip
    - SDWIS/LabToState 3.0 Installation Guide and Release Notes
    - SDWIS/LabToState 3.0 Requirements and Design Document
    - SDWIS/LabToState 3.0 User Guide

2.2 System Advisories

Please note the following advisories pertaining to the SDWIS/LabToState 3.0:

- The documentation provided with SDWIS/LabToState is intended for use by primacy agencies. It is the responsibility of the primacy to provide site-specific instructions for the laboratories.

- A CROMERR Checklist is included in Appendix E. You may use this checklist as a template for your primacy agency to obtain CROMERR approval. 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.

- To submit large amounts of data, you may need to increase the memory allocation assigned to SDWIS/LabToState by the application server (Tomcat) – section 3.2.2.4 covers Java memory recommendations.

- When submitting Sample Measures, those logical objects must be included in the same CSV file as the Sample to which it refers. This constraint also applies to Other Result Measures, which must be submitted with Sample Results.

- Microsoft Excel is the most widely used tool for creating .csv files. Be advised, however, that when you open a .csv file into Microsoft Excel, that Excel attempts to identify the domain of a column based upon its content. Care must be taken to reformat columns properly so that cell content is not altered when saving back to a .csv format. Below are a few examples.

- Analyte Codes are text fields. Although the format of an analyte code appears to be numeric, it must be formatted as a text column in excel so that leading zeros are retained. An example is analyte 0100, when saved as a numeric, will be saved as 100.

- Date fields are most easily formatted as text fields. This too is to retain leading zeros on the month fields. If you choose to use a date edit mask, the month and day should include the respective DD and MM custom format, supplied in the site properties file. This custom format should match the date format.

- When using the ODBC-SQL file format to Upload and Validate, do not supply a semi-colon at the end of the SQL statement.

- Before running a SQL statement when using the embedded SDWIS/LabToState application, ensure you download the respective .jar file for your DBMS (i.e., ojdbc14.jar for Oracle or sqljdbc.jar for SQL Server) to the following location: \labtostate\docBase\WEB-INF\lib

- Before running a SQL statement, you must supply database connection parameters on the Configuration page for the database in which you intend to extract from. Below are examples:

<u>For Oracle:</u>
Database Driver Class Name: oracle.jdbc.OracleDriver
Database Connection URL: jdbc:oracle:thin:@*yourServer*:1521:*yourDatabase*
Database User ID: *username*
Database Password: *password*

<u>For SQL Server 2005:</u>
Database Driver Class Name: com.microsoft.sqlserver.jdbc.SQLServerDriver
Database Connection URL:
jdbc:sqlserver://*yourServer:*1433;DatabaseName=*yourDatabase*;integratedSecurity=false;SelectMethod=cursor
Database User ID: *username*
Database Password: *password*

<u>For MSAccess:</u> You must have an MSACCESS ODBC System Data Source established on the application server that provides the location of the .mdb you wish to extract from. In the example below, DBTEST is the name of the MSACCESS ODBC System Data Source created on the application server.

Database Driver Class Name: sun.jdbc.odbc.JdbcOdbcDriver

Database Connection URL: jdbc:odbc:*DBTEST*

- To successfully certify a job when using the embedded SDWIS/LabToState application, you must supply an email address in the userList.xml file located under ...\labtostate\docBase\WEB-INF\classes\properties\userList.xml

- To receive email notifications, you must supply a valid email address in the userList.xml file located under ...\labtostate\docBase\WEB-INF\classes\properties\

3.0 INSTALLATION GUIDE

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 3.1, Stand-Alone Client Application Installation. To install it as a web-based application, proceed to Section 3.2, Web Based Application Installation.

3.1 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 3.1.1, Upgrade Installation – Stand -Alone Client Application. If you have never installed it at your site, you should proceed to Section 3.1.2, First Time Installation - Stand-Alone Client Application.

3.1.1 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 3.1.1.1, Archive SDWIS/LabToState Files
2	Remove the SDWIS/LabToState Software	Section 3.1.1.2, Remove the SDWIS/LabToState Software
3	Install SDWIS/LabToState Software	Section 3.1.1.3, Install SDWIS/LabToState Software

Exhibit 2. SDWIS/LabToState Upgrade Installation Checklist

3.1.1.1 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\jobFolders	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.
Directory	[InstalledDirectory]\docBase\WEB-INF\classes\db\labtostate	Contains the embedded database containing historical processed job information.

Exhibit 3. SDWIS/LabToState Files To Archive

The Job 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.

The db\labtostate directory is archived so that it may be reloaded to maintain continuity of processed jobs after the software upgrade.

3.1.1.2 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.

3.1.1.3 Install SDWIS/LabToState Software

After the existing release of SDWIS/LabToState is removed, you can install the SDWIS/LabToState software using the installation procedures described in Section 3.1.2, First Time Installation - Stand-Alone Client Application.

At the completion of Section 3.1.2, you may reinstate the Job directory and the db\labtostate directory from the archive created in Section 3.1.1.1.

3.1.2 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	Install the SDWIS/LabToState software	Section 3.1.2.1, Install the SDWIS/LabToState Software
2	Start SDWIS/LabToState	Section 3.1.2.2, SDWIS/LabToState Startup
3	Shut down SDWIS/LabToState	Section 3.1.2.3, SDWIS/LabToState Shut Down
4	Configure SDWIS/LabToState	Section 3.1.2.4, Configure SDWIS/LabToState

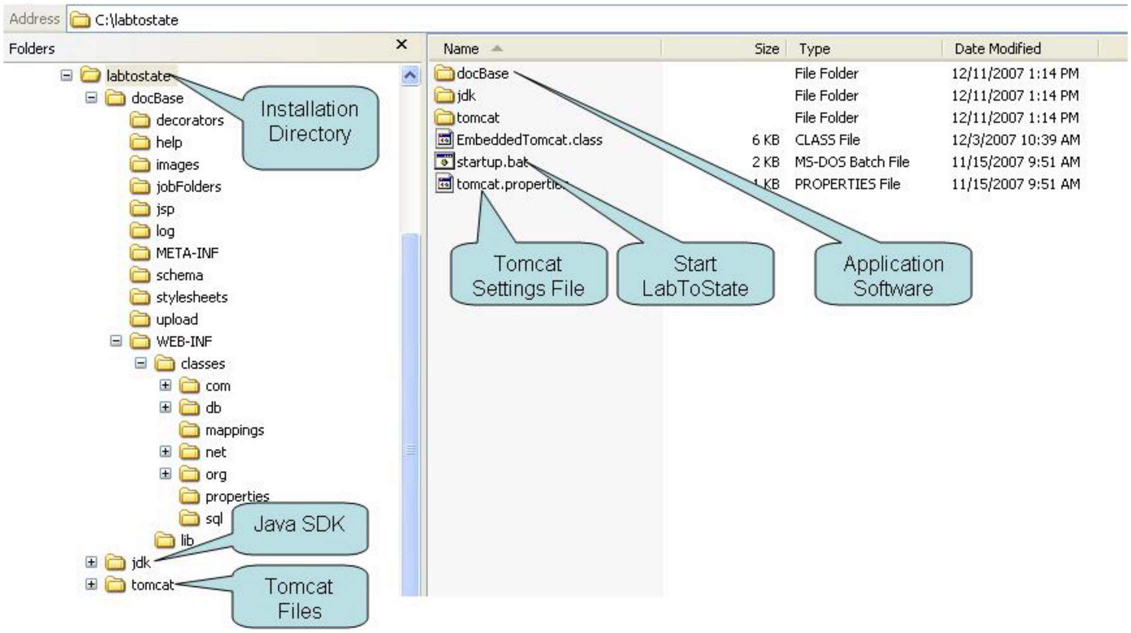
Exhibit 4. Install SDWIS/LabToState – Stand-Alone Client Application

3.1.2.1 Install the SDWIS/LabToState Software

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). Unzip the embedded-labtostate.zip using folder names to your desired deployment folder. At the end of the unzip, the software will be in a folder structure similar to:

C:\labtostate-embedded\labtostate

Exhibit 5 provides an annotated list of the directory structure after unzipping the installation package.



Address C:\labtostate

Name	Size	Type	Date Modified
docBase		File Folder	12/11/2007 1:14 PM
jdk		File Folder	12/11/2007 1:14 PM
tomcat		File Folder	12/11/2007 1:14 PM
EmbeddedTomcat.class	6 KB	CLASS File	12/3/2007 10:39 AM
startup.bat	2 KB	MS-DOS Batch File	11/15/2007 9:51 AM
tomcat.properties	1 KB	PROPERTIES File	11/15/2007 9:51 AM

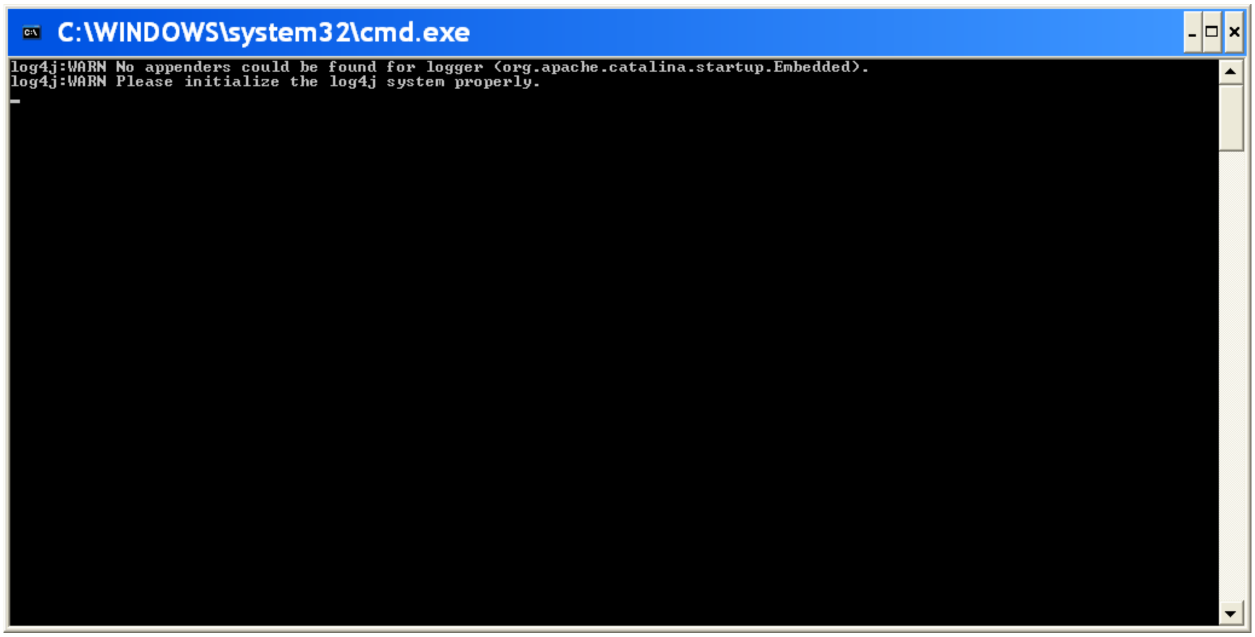
Exhibit 5. Install SDWIS/LabToState-Web Based Application

3.1.2.2 SDWIS/LabToState Startup

Start the application by double clicking on the startup.bat located in the folder in which you installed the application, such as C:\labtostate-embedded\labtostate\startup.bat. 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 warnings 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.



```
log4j:WARN No appenders could be found for logger (org.apache.catalina.startup.Embedded).
log4j:WARN Please initialize the log4j system properly.
_
```

Exhibit 6. Stand-Alone Client Application DOS Window

3.1.2.3 SDWIS/LabToState Shut Down

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.

3.1.2.4 Configure SDWIS/LabToState

After completing the above steps, the installation of SDWIS/LabToState software is complete. You should continue to Section 3.33, Site Customization Procedures of this document to configure the SDWIS/LabToState software for your site.

3.2 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) 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 3.2.1, Upgrade Installation-Web Based Application. If you have never installed it at your site, you should proceed to Section 3.2.2, First Time Installation Web-Based Application.

3.2.1 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 3.2.1.1, Archive SDWIS/LabToState Files
2	Remove the SDWIS/LabToState Software	Section 3.2.1.2, Remove the SDWIS/LabToState Software
3	Install SDWIS/LabToState Software	Section 3.2.1.3, Install SDWIS/LabToState Software

Exhibit 7. SDWIS/LabToState Upgrade Installation Checklist

3.2.1.1 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.
Directory	*[Tomcat_Home]*\ webapps\labtoState\WEB-INF\classes\db\labtostate	Contains the embedded database containing historical processed job information.

Exhibit 8. SDWIS/LabToState Files To Archive

The Job 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.

The db\labtostate directory is archived so that it may be reloaded to maintain continuity of processed jobs after the software upgrade.

3.2.1.2 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).

3.2.1.3 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 3.2.2, First Time Installation Web-Based Application.

At the completion of Section 3.2.2, you may reinstate the Job directory and the db\labtostate directory from the archive created in Section 3.2.1.1.

3.2.2 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	Install support software (i.e., Java SDK and Tomcat application server) used by SDWIS/LabToState	3.2.2.1 Install Support Software Used by SDWIS/LabToState
2	Install the SDWIS/LabToState software	3.2.2.2 Install the SDWIS/LabToState Software
3	Set Java environment variable	3.2.2.3 Set Environment Variables
4	Set Memory Allocation	3.2.2.4 Set Memory Allocation

Exhibit 9. SDWIS/LabToState First Time Installation Checklist

3.2.2.1 Install Support Software Used by SDWIS/LabToState

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 3.0 may be deployed using the JDK v 1.5 but has not been certified in this environment. To attempt deployment using JDK 1.5, you must copy the jaxb-api.jar from the *[Tomcat_Home]*\webapps\labtoState\WEB-INF\lib to *[Tomcat_Home]*\common\endorsed.

3.2.2.2 Install the SDWIS/LabToState Software

Load the SDWIS/LabToState software on the Tomcat application server. The installation package delivers the SDWIS/LabToState software as a single packed file named labtostate.war. 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.

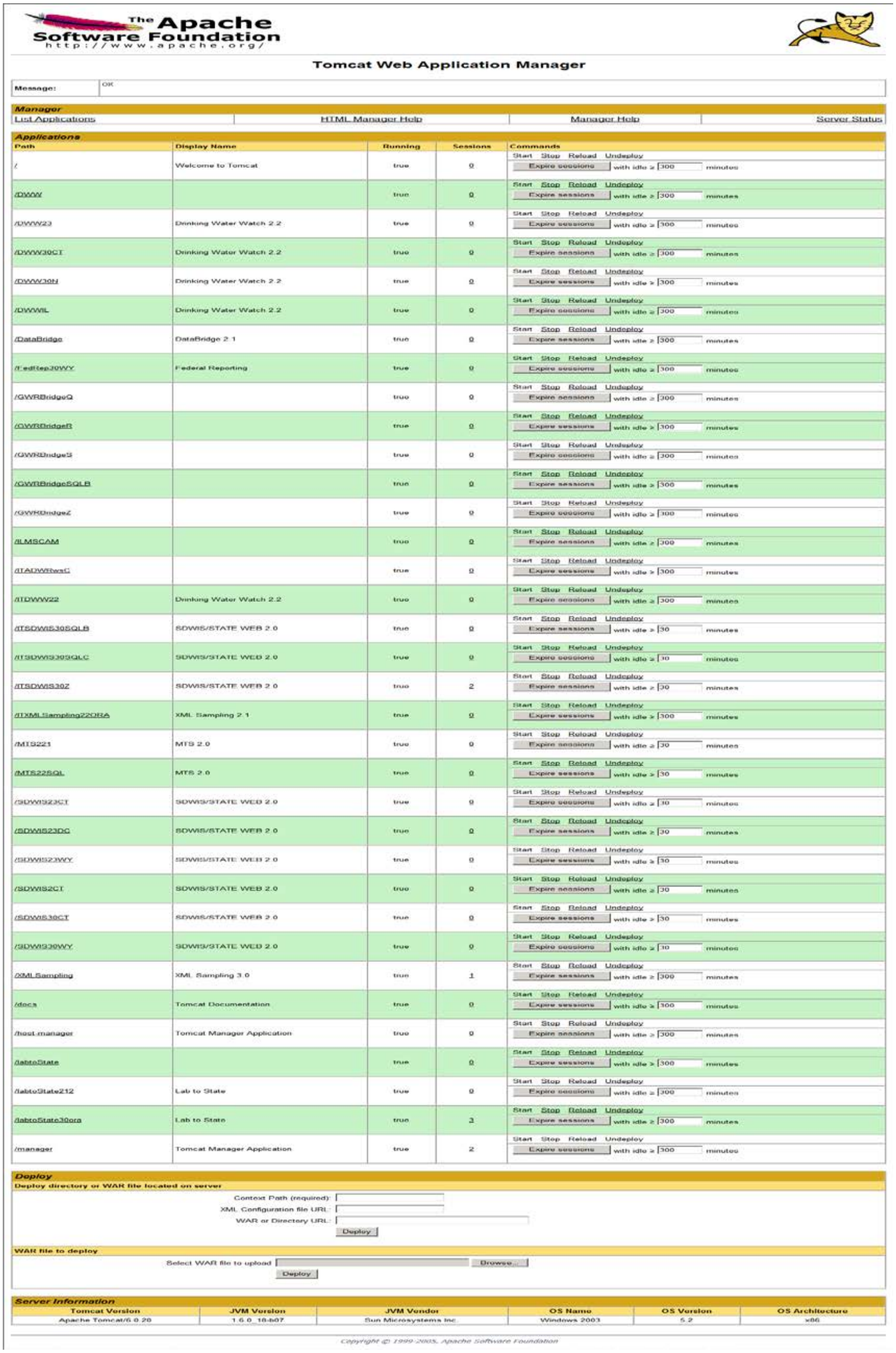
2. Open a web browser and type **http://localhost:8080** in the address bar. The Tomcat home page will be displayed.

3. Click on the Tomcat Manager link, located on the left side of the page in the box titled Administration. Login as the Tomcat administrator. 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.



Tomcat Web Application Manager

Message:	OK

Manager

List Applications	HTML Manager Help	Manager Help	Server Status

Applications

Path	Display Name	Running	Sessions	Commands
/	Welcome to Tomcat	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/DWW		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/DWW23	Drinking Water Watch 2.2	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/DWW30CT	Drinking Water Watch 2.2	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/DWW30N	Drinking Water Watch 2.2	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/DWWIL	Drinking Water Watch 2.2	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/DataBridge	DataBridge 2.1	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/FedRep30WY	Federal Reporting	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/GWRBridgeQ		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/GWRBridgeR		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/GWRBridgeS		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/GWRBridgeSQLB		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/GWRBridgeZ		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/ILMSCAM		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/ITADWRwsC		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/ITDWW22	Drinking Water Watch 2.2	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/ITSDWIS30SQLB	SDWIS/STATE WEB 2.0	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/ITSDWIS30SQLC	SDWIS/STATE WEB 2.0	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/ITSDWIS30Z	SDWIS/STATE WEB 2.0	true	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/ITXMLSampling22ORA	XML Sampling 2.1	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/MTS221	MTS 2.0	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/MTS22SQL	MTS 2.0	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/SDWIS23CT	SDWIS/STATE WEB 2.0	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/SDWIS23DC	SDWIS/STATE WEB 2.0	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/SDWIS23WY	SDWIS/STATE WEB 2.0	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/SDWIS2CT	SDWIS/STATE WEB 2.0	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/SDWIS30CT	SDWIS/STATE WEB 2.0	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/SDWIS30WY	SDWIS/STATE WEB 2.0	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 30 minutes
/XMLSampling	XML Sampling 3.0	true	1	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/docs	Tomcat Documentation	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/host-manager	Tomcat Manager Application	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/labtoState		true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/labtoState212	Lab to State	true	0	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/labtoState30ora	Lab to State	true	3	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes
/manager	Tomcat Manager Application	true	2	Start Stop Reload Undeploy Expire sessions with idle ≥ 300 minutes

Deploy

Deploy directory or WAR file located on server

Context Path (required):
XML Configuration file URL:
WAR or Directory URL:

WAR file to deploy

Select WAR file to upload

Server Information

Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture
Apache Tomcat/6.0.20	1.6.0_18-b07	Sun Microsystems Inc.	Windows 2003	5.2	x86

Exhibit 10. Tomcat Web Application Manager — Before Deployment

3.2.2.3 Set Environment Variables

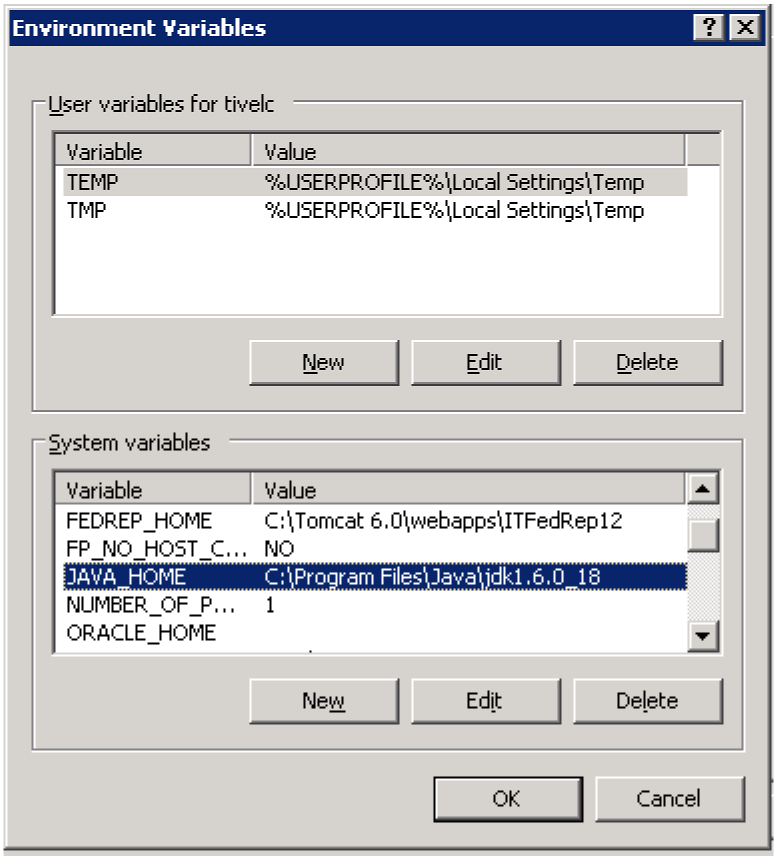
Set the environment variable used by the Apache Tomcat. The "JAVA_HOME" variable is set to the location of the java compiler software and is used by the Apache Tomcat server. By completing the following tasks, you set these environment variables:

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.



Variable	Value
TEMP	%USERPROFILE%\Local Settings\Temp
TMP	%USERPROFILE%\Local Settings\Temp

Variable	Value
FEDREP_HOME	C:\Tomcat 6.0\webapps\ITFedRep12
FP_NO_HOST_C...	NO
JAVA_HOME	C:\Program Files\Java\jdk1.6.0_18
NUMBER_OF_P...	1
ORACLE_HOME	

Exhibit 11. SDWIS/LabToState System Environment Variables

3.2.2.4 Set Memory Allocation

You may need to adjust your Java memory settings in Apache Tomcat to optimize SDWIS product performance – particularly if your input files are large. The two most important JVM settings are MaxPermSize and Max JVM Heap Allocation. In general, the following guidelines apply:

- MaxPermSize + Max JVM Heap Allocation < 75% of web server's total RAM.
- Initial PermSize = Minimum PermSize = MaxPermSize
- Initial JVM Heap = Minimum JVM Heap = Max JVM Heap

Note: The official term Sun uses is "minimum" but in the context of JVM settings, Apache Tomcat uses the term "initial". Also, Windows 2000/2003 Server documentation (excluding Enterprise Edition and 64-bit OS) indicates the maximum amount of memory that can be supported is 4GB with a maximum of 2GB allocated to any application (e.g., Tomcat).

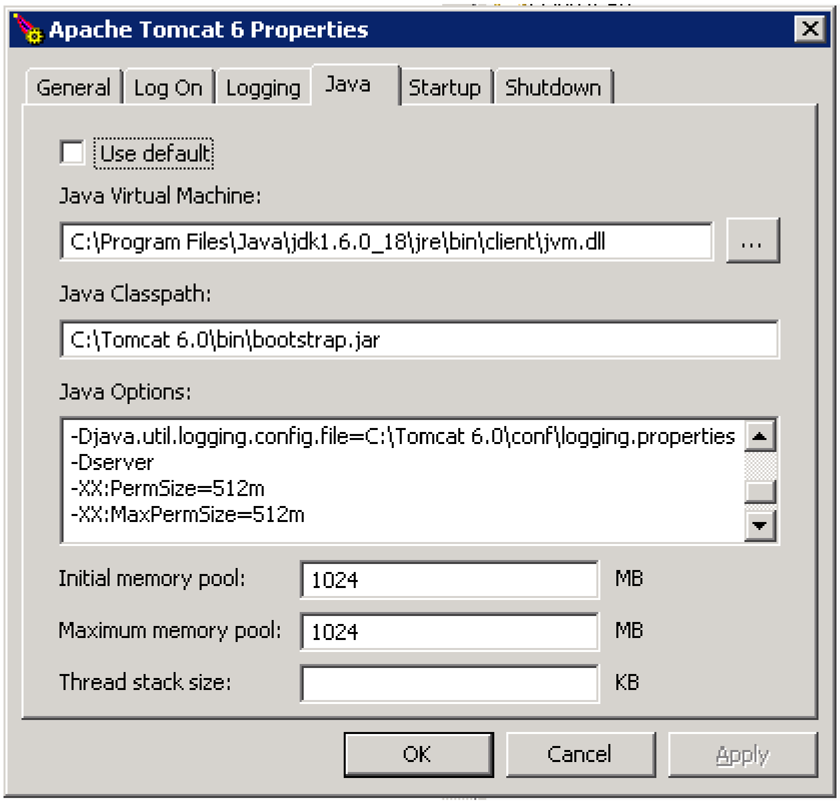
If your web server has 1 GB RAM, the following JVM settings are suggested:

- Initial PermSize = Minimum PermSize = MaxPermSize = 256 MB
- Initial JVM Heap = Minimum JVM Heap = Max JVM Heap = 512 MB

If your web server has 2 GB RAM (or more), the following JVM settings are suggested:

- Initial PermSize = Minimum PermSize = MaxPermSize = 512 MB
- Initial JVM Heap = Minimum JVM Heap = Max JVM Heap = 1024 MB

If you are running Tomcat as a service, to set or change the JVM settings, access the Apache Tomcat Properties page by selecting Desktop Start->Programs->Apache Tomcat 6.0->Configure Tomcat or by going directly to the folder where Tomcat is installed (i.e. C:\Tomcat-6.0.\bin) and clicking on tomcat6w.exe.



□ Use default

Java Virtual Machine: C:\Program Files\Java\jdk1.6.0_18\jre\bin\client\jvm.dll

Java Classpath: C:\Tomcat 6.0\bin\bootstrap.jar

Java Options:

```
-Djava.util.logging.config.file=C:\Tomcat 6.0\conf\logging.properties
-Dserver
-XX:PermSize=512m
-XX:MaxPermSize=512m
```

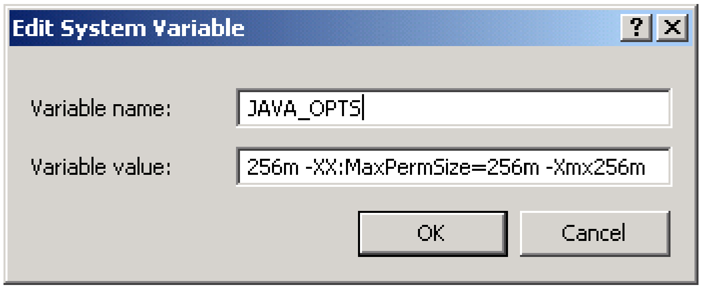
Initial memory pool: 1024 MB

Maximum memory pool: 1024 MB

Thread stack size: KB

Exhibit 12. Tomcat as a Service Memory Settings

If you are running Tomcat as a .bat, you will need to create an environmental variable called JAVA_OPTS to set or change the JVM settings.



Variable name: JAVA_OPTS

Variable value: 256m -XX:MaxPermSize=256m -Xmx256m

Exhibit 13. Tomcat as an Application Memory Settings

A. To set JVM PermSize, add the following under the Java Options box of the Apache Tomcat Properties page (Ensure that there are no spaces when you enter these settings.):

(1) If your web server has 1 GB RAM

- -Dserver

- -XX:PermSize=256m (this is initial or minimum PermSize)
- -XX:MaxPermSize=256m

(2) If your web server has 2 GB RAM (or more)

- -Dserver
- -XX:PermSize=512m (this is initial or minimum PermSize)
- -XX:MaxPermSize=512m

B. To set JVM Heap Size, use the same Apache Tomcat Properties page

(1) If your web server has 1 GB RAM

- Set the Initial/Minimum memory pool to 512MB
- Set the Maximum memory pool to 512MB.

(2) If your web server has 2 GB RAM (or more)

- Set the Initial/Minimum memory pool to 1024MB
- Set the Maximum memory pool to 1024MB.

The changed settings will take affect the next time Tomcat is started.

Continue to Section 3.3, Site Customization Procedures, to configure the SDWIS/LabToState software for your site.

After configuring SDWIS/LabToState in section 3.3, you should complete the following activities:

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

3.3 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.

3.3.1 Site Customization Files

To customize its deployment, SDWIS/LabToState uses several text files and XML documents. With the exception of the siteProperties.txt, SDWIS/LabToState does not offer any tools to administer these files. You may use any standard text editor to modify or change the content. To make changes to the siteProperties.txt file use the Configuration page available off the SDWIS/LabToState homepage. 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.

3.3.1.1 Site Properties File

The Site Properties File is a text file formatted as name/value pairs. The System Administrator should update values using the Configuration page 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

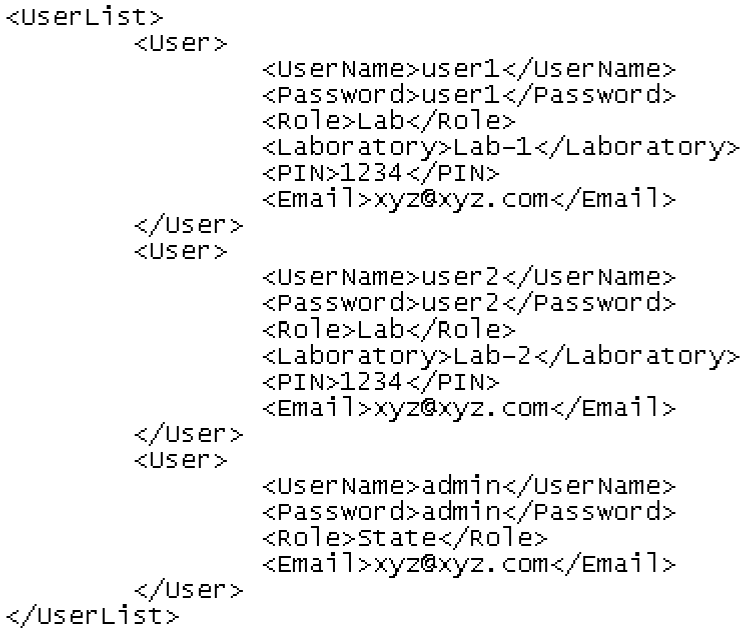
3.3.1.2 User Identification XML Document

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. Please see Appendix J (Recommendations for Making SDWIS Applications Deployed on External Facing Server More Secure) for specific secure procedure recommendations including changing default passwords.

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

3.3.1.3 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

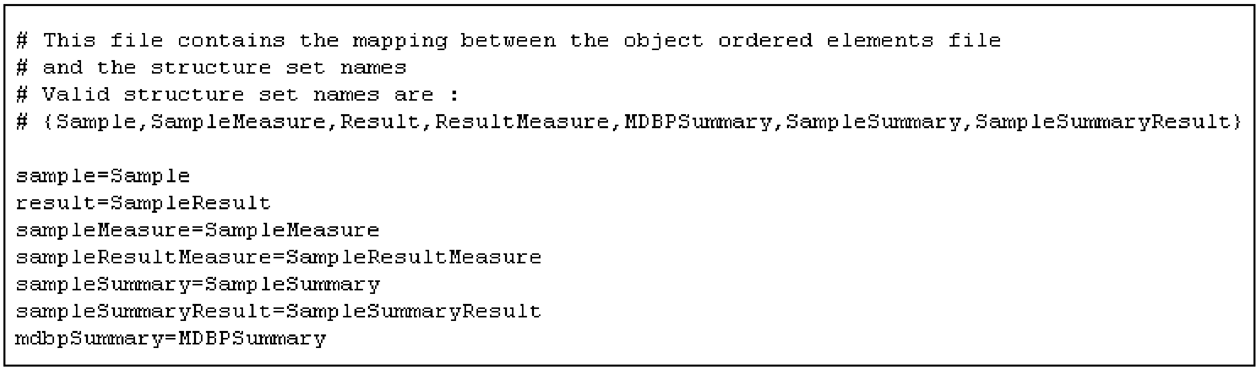
3.3.2 Data File Content Configuration

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 Appendix A of the *SDWIS/XML Sampling 3.0 Design Document.* The Element Configuration File is a text file and the site creates a separate Element Configuration File for each logical object. If the site configures both a TCRSample and a GeneralSample logical object in the master.txt file, 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

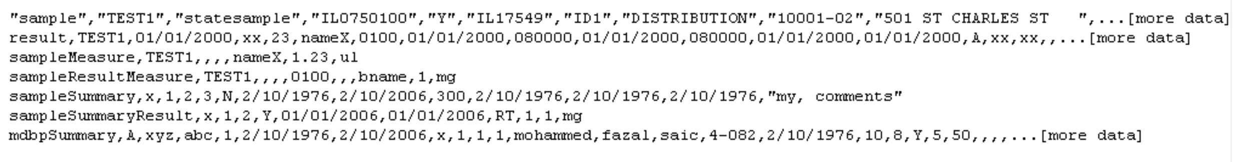
Physical File Name: *[logicalobject]*.txt

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

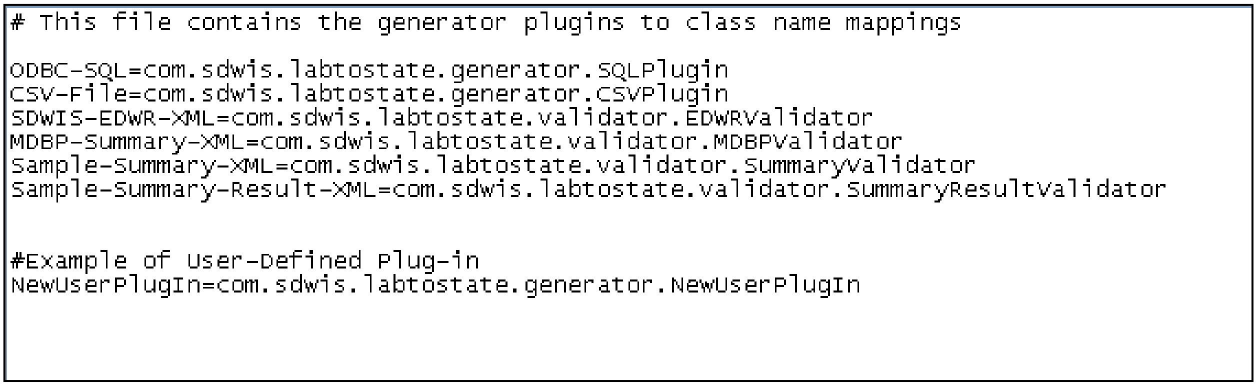
3.3.3 Data Source Registration File

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 FUNCTIONAL CHANGES

4.1 Schema Changes

SDWIS_eDWR_v3.0

- Add ‘TG’ enumerated value to eDWR Submission/LabReport/ Sample/SampleIdentification/ SampleMonitoringTypeCode.
- Add new tag eDWR Submission/LabReport/ Sample/SampleIdentification/ SampleLocationCollectionAddress.
- Removed minimum inclusion of 0 from eDWR Submission/LabReport/ SampleAnalysisResult/AnalysisResult/Result/MeasurementValue.

4.2 Other Changes

- Added mapping support for new Sample Collection Address.

- Upload function inspects file extension and only allows types of .xml, .csv, and .zip. (Note that it does not inspect the contents of a .zip file. This should be accomplished using site virus protection software.)

5.0 RESOLVED SYSTEM INCIDENT REPORTS (SIR)

SIR 18787 and 19143: User unable to delete Job from Job List page if Job has been previously certified.

SIR 19473: Certification function will consistently provide certification statement and deliver certified file to XML Sampling inbox.

6.0 KNOWN SOFTWARE DISCREPANCIES

SIR 19130: A known defect is identified is the JAXB technology implemented in LabToState where content with decimal places greater than 8 significant digits is converted to scientific notation. Two suggested work arounds until a product fix is available are:

1 Use online Sampling to enter results with reporting limit of 8 or 9 decimal places.

2 Modify the XML file generated by LabToState. For all reporting limits written in scientific notation, manually replace it with plain numeric value. For example, replace 1E-8 with 0.00000001, replace 5E-9 with 0.000000005. Save the XML file and submit it using XMLSampling.

7.0 CROMERR IMPLEMENTATION

To support CROMERR, SDWIS/LabToState originally 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 Signature Process Section (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 (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.

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. 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. 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.

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.

SDWIS/LabToState Site Property Names and Description			
Name	**Description**	**Initial Setting**	**Set By**
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 Administrator
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
EmailFromAddress	Contains the e-mail address for the SDWIS/LabToState administrator. This e-mail address is used to send e-mails.	LabToState	Email Administrator

SDWIS/LabToState Site Property Names and Description			
Name	**Description**	**Initial Setting**	**Set By**
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

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"). sun.jdbc.odbc.JdbcOdbcDriver	sun.jdbc.odbc.JdbcOdbcDriver	This value can be modified 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"). jdbc:odbc:SDWIS	jdbc:odbc:SDWIS	This value can be modified 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

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

SDWIS/LabToState Site Property Names and Description			
Name	**Description**	**Initial Setting**	**Set By**
XMLSamplingIntegrator Class	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

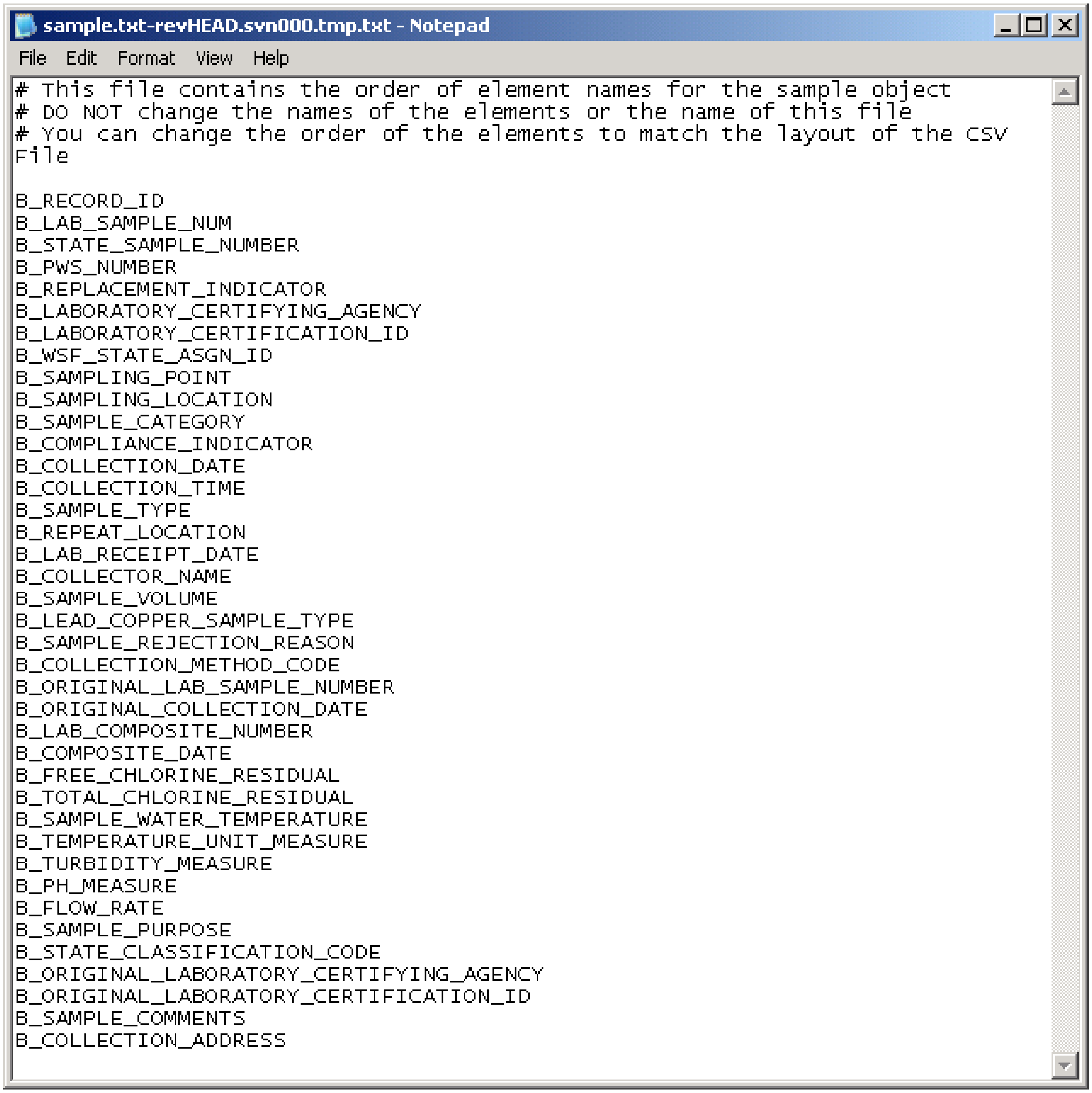
SDWIS/LabToState Site Property Names and Description			
Name	**Description**	**Initial Setting**	**Set By**
XML_Sampling_Inbox	Contains the full path and folder name of the SDWIS SDWIS/XML Sampling Inbox. This name/value pair and the name/value pair (XMLSamplingIntegratorClass) that identifies the Java 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
EdwrSchema	Contains the name of the current schema for lab sample reporting.	SDWIS_eDWR_v3.0.xsd	On Delivery
SummaryResultSchema	Contains the name of the current schema for reporting sample summary results.	SDWIS_SummaryResult_v3.0.xsd	On Delivery
SummarySchema	Contains the name of the current schema for reporting sample summaries.	SDWIS_Summary_v3.0.xsd	On Delivery
MdbpSchema	Contains the name of the current schema for reporting MDBP Sample Summaries.	SDWIS_MDBPSummary_v3.0.xsd	On Delivery

APPENDIX B

Object and Element Configuration Files

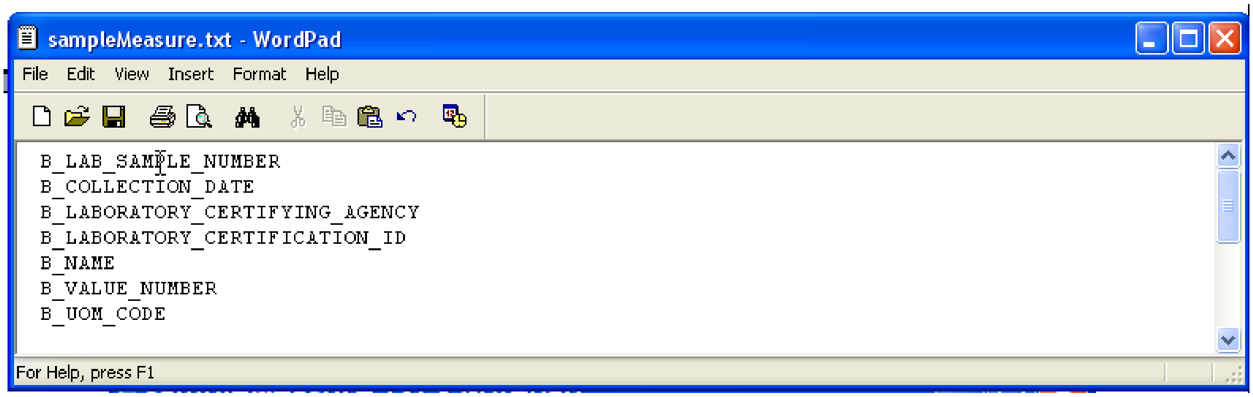
This page is intentionally left blank.

Sample Object


sample.txt-revHEAD.svn000.tmp.txt - Notepad
File Edit Format View Help

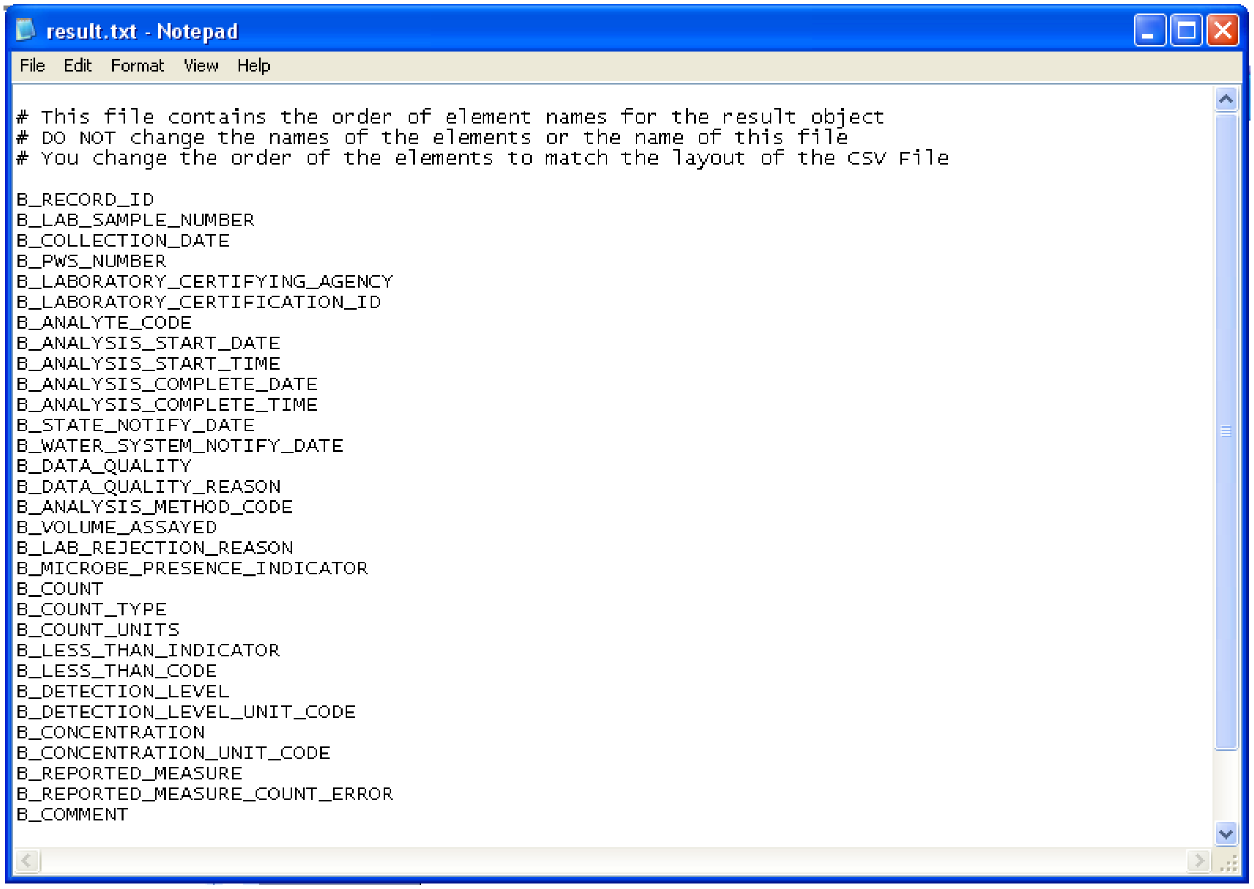
```
# This file contains the order of element names for the sample object
# DO NOT change the names of the elements or the name of this file
# You can change the order of the elements to match the layout of the CSV
File

B_RECORD_ID
B_LAB_SAMPLE_NUM
B_STATE_SAMPLE_NUMBER
B_PWS_NUMBER
B_REPLACEMENT_INDICATOR
B_LABORATORY_CERTIFYING_AGENCY
B_LABORATORY_CERTIFICATION_ID
B_WSF_STATE_ASGN_ID
B_SAMPLING_POINT
B_SAMPLING_LOCATION
B_SAMPLE_CATEGORY
B_COMPLIANCE_INDICATOR
B_COLLECTION_DATE
B_COLLECTION_TIME
B_SAMPLE_TYPE
B_REPEAT_LOCATION
B_LAB_RECEIPT_DATE
B_COLLECTOR_NAME
B_SAMPLE_VOLUME
B_LEAD_COPPER_SAMPLE_TYPE
B_SAMPLE_REJECTION_REASON
B_COLLECTION_METHOD_CODE
B_ORIGINAL_LAB_SAMPLE_NUMBER
B_ORIGINAL_COLLECTION_DATE
B_LAB_COMPOSITE_NUMBER
B_COMPOSITE_DATE
B_FREE_CHLORINE_RESIDUAL
B_TOTAL_CHLORINE_RESIDUAL
B_SAMPLE_WATER_TEMPERATURE
B_TEMPERATURE_UNIT_MEASURE
B_TURBIDITY_MEASURE
B_PH_MEASURE
B_FLOW_RATE
B_SAMPLE_PURPOSE
B_STATE_CLASSIFICATION_CODE
B_ORIGINAL_LABORATORY_CERTIFYING_AGENCY
B_ORIGINAL_LABORATORY_CERTIFICATION_ID
B_SAMPLE_COMMENTS
B_COLLECTION_ADDRESS
```

Sample Measure Object


sampleMeasure.txt - WordPad
File Edit View Insert Format Help
For Help, press F1

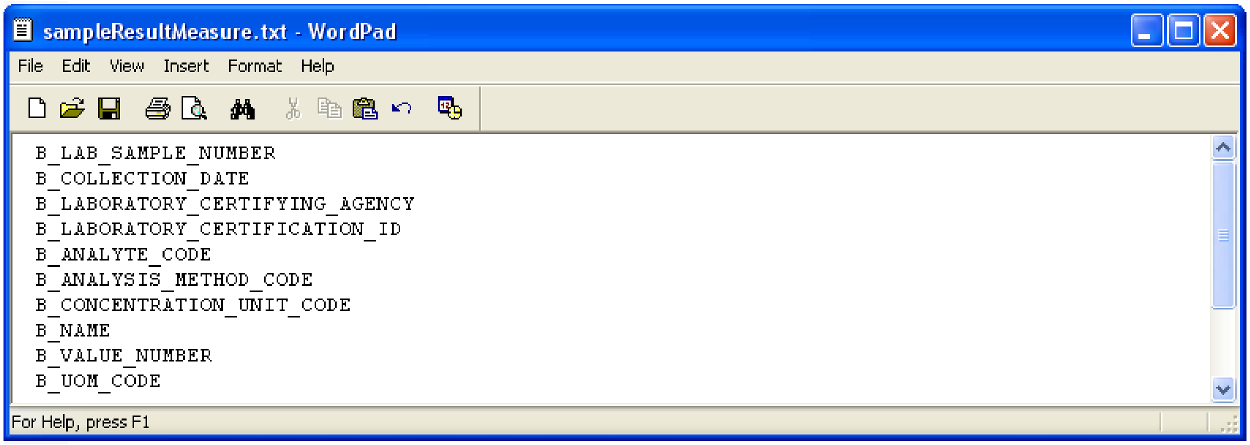
```
B_LAB_SAMPLE_NUMBER
B_COLLECTION_DATE
B_LABORATORY_CERTIFYING_AGENCY
B_LABORATORY_CERTIFICATION_ID
B_NAME
B_VALUE_NUMBER
B_UOM_CODE
```

Result Object



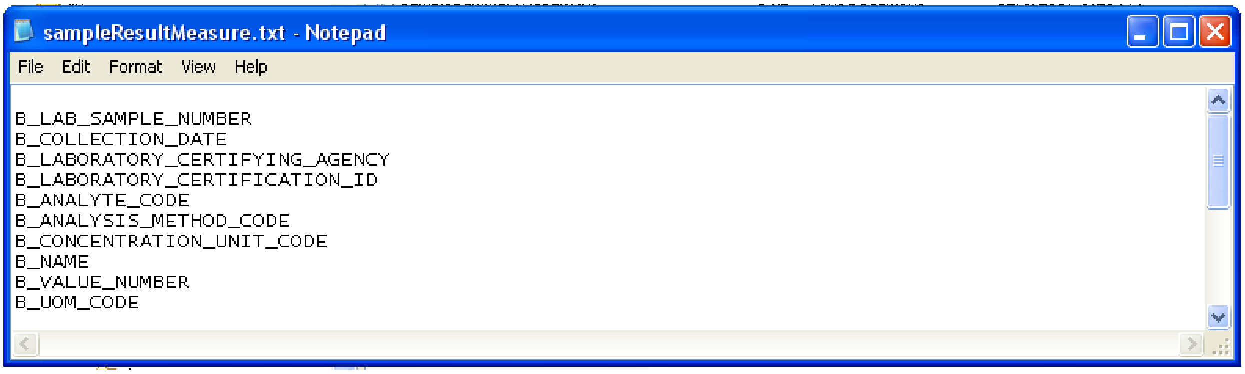
```
# This file contains the order of element names for the result object
# DO NOT change the names of the elements or the name of this file
# You change the order of the elements to match the layout of the CSV File

B_RECORD_ID
B_LAB_SAMPLE_NUMBER
B_COLLECTION_DATE
B_PWS_NUMBER
B_LABORATORY_CERTIFYING_AGENCY
B_LABORATORY_CERTIFICATION_ID
B_ANALYTE_CODE
B_ANALYSIS_START_DATE
B_ANALYSIS_START_TIME
B_ANALYSIS_COMPLETE_DATE
B_ANALYSIS_COMPLETE_TIME
B_STATE_NOTIFY_DATE
B_WATER_SYSTEM_NOTIFY_DATE
B_DATA_QUALITY
B_DATA_QUALITY_REASON
B_ANALYSIS_METHOD_CODE
B_VOLUME_ASSAYED
B_LAB_REJECTION_REASON
B_MICROBE_PRESENCE_INDICATOR
B_COUNT
B_COUNT_TYPE
B_COUNT_UNITS
B_LESS_THAN_INDICATOR
B_LESS_THAN_CODE
B_DETECTION_LEVEL
B_DETECTION_LEVEL_UNIT_CODE
B_CONCENTRATION
B_CONCENTRATION_UNIT_CODE
B_REPORTED_MEASURE
B_REPORTED_MEASURE_COUNT_ERROR
B_COMMENT
```

Sample Result Measure Object


sampleResultMeasure.txt - WordPad
File Edit View Insert Format Help
For Help, press F1

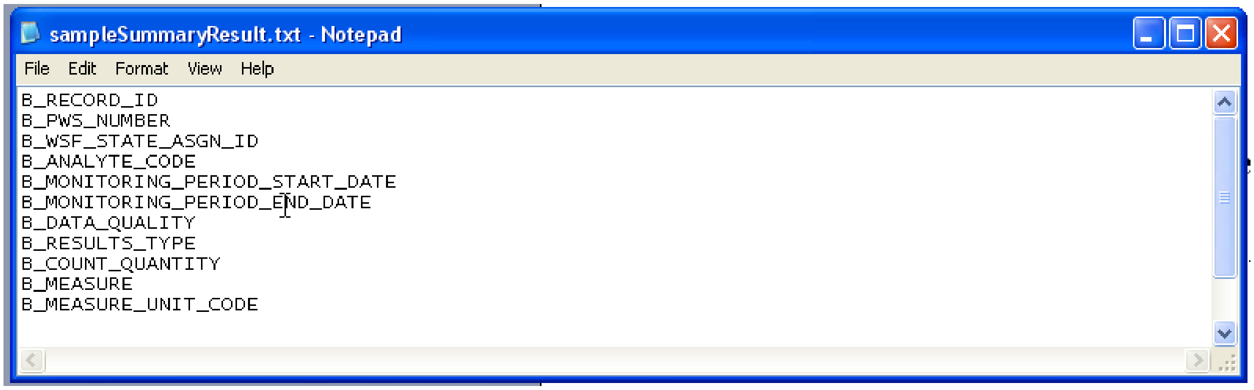
```
B_LAB_SAMPLE_NUMBER
B_COLLECTION_DATE
B_LABORATORY_CERTIFYING_AGENCY
B_LABORATORY_CERTIFICATION_ID
B_ANALYTE_CODE
B_ANALYSIS_METHOD_CODE
B_CONCENTRATION_UNIT_CODE
B_NAME
B_VALUE_NUMBER
B_UOM_CODE
```

Sample Summary Object



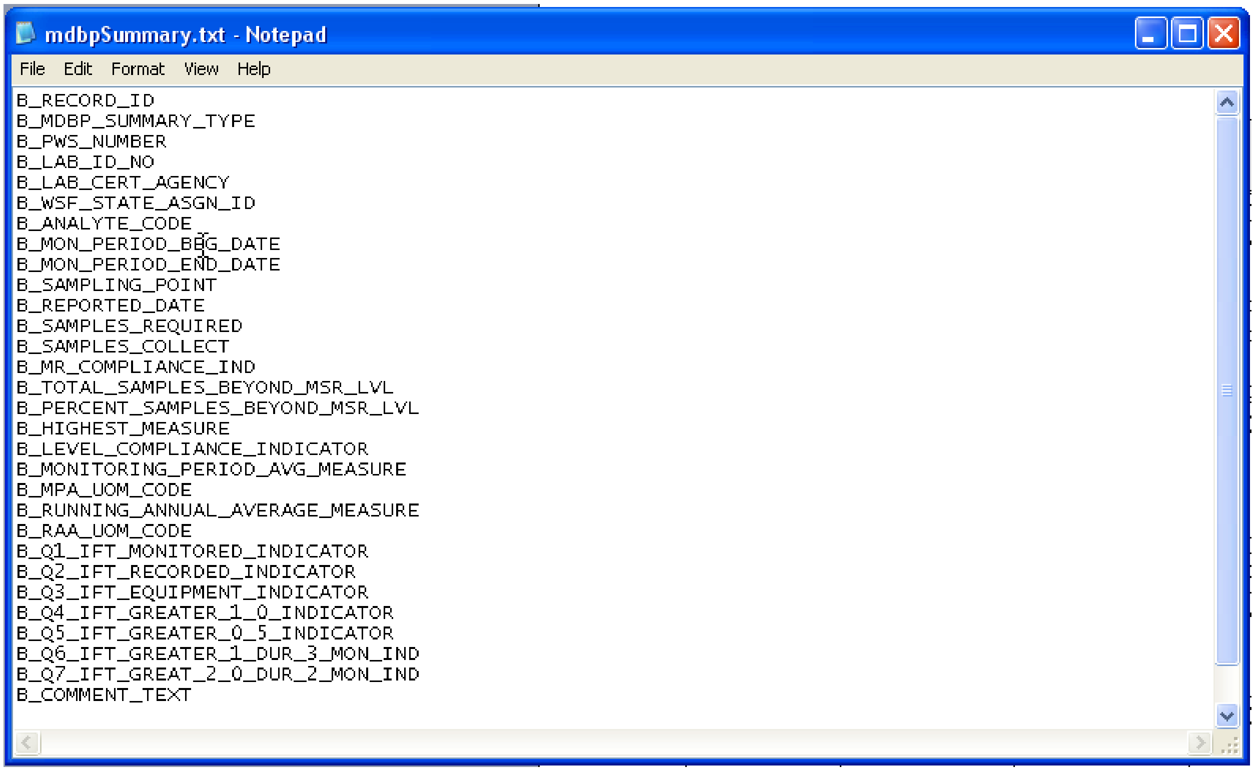
```
B_LAB_SAMPLE_NUMBER
B_COLLECTION_DATE
B_LABORATORY_CERTIFYING_AGENCY
B_LABORATORY_CERTIFICATION_ID
B_ANALYTE_CODE
B_ANALYSIS_METHOD_CODE
B_CONCENTRATION_UNIT_CODE
B_NAME
B_VALUE_NUMBER
B_UOM_CODE
```

Sample Summary Result Object



```
B_RECORD_ID
B_PWS_NUMBER
B_WSF_STATE_ASGN_ID
B_ANALYTE_CODE
B_MONITORING_PERIOD_START_DATE
B_MONITORING_PERIOD_END_DATE
B_DATA_QUALITY
B_RESULTS_TYPE
B_COUNT_QUANTITY
B_MEASURE
B_MEASURE_UNIT_CODE
```

MDBP Summary Object



```
B_RECORD_ID
B_MDBP_SUMMARY_TYPE
B_PWS_NUMBER
B_LAB_ID_NO
B_LAB_CERT_AGENCY
B_WSF_STATE_ASGN_ID
B_ANALYTE_CODE
B_MON_PERIOD_BEG_DATE
B_MON_PERIOD_END_DATE
B_SAMPLING_POINT
B_REPORTED_DATE
B_SAMPLES_REQUIRED
B_SAMPLES_COLLECT
B_MR_COMPLIANCE_IND
B_TOTAL_SAMPLES_BEYOND_MSR_LVL
B_PERCENT_SAMPLES_BEYOND_MSR_LVL
B_HIGHEST_MEASURE
B_LEVEL_COMPLIANCE_INDICATOR
B_MONITORING_PERIOD_AVG_MEASURE
B_MPA_UOM_CODE
B_RUNNING_ANNUAL_AVERAGE_MEASURE
B_RAA_UOM_CODE
B_Q1_IFT_MONITORED_INDICATOR
B_Q2_IFT_RECORDED_INDICATOR
B_Q3_IFT_EQUIPMENT_INDICATOR
B_Q4_IFT_GREATER_1_0_INDICATOR
B_Q5_IFT_GREATER_0_5_INDICATOR
B_Q6_IFT_GREATER_1_DUR_3_MON_IND
B_Q7_IFT_GREAT_2_0_DUR_2_MON_IND
B_COMMENT_TEXT
```

APPENDIX C

Element Names by Object

Click on the icon below to view the electronic file for Appendix C

SDWIS LabToState
3.0 Logical Object Ma

This page is intentionally left blank.

APPENDIX D

Interface Specifications

This page is intentionally left blank.

D-1. 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.

D-2. 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. Please note Appendix J (Recommendations for Making SDWIS Applications Deployed on External Facing Server More Secure) for specific secure procedure recommendations including changing default passwords.

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 boolean, 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.

D-3. 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.

This page intentionally left blank.

APPENDIX E

SDWIS/LabToState CROMERR Checklist

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

LabToState Checklist
Release.doc

This page intentionally left blank.

APPENDIX F

SDWIS/LabToState Example Files

The embedded zip file contains examples of CSV files.

LabToState_EXAMPLE_CSV_FILES.zip

This page intentionally left blank.

APPENDIX G

SDWIS/LabToState 3.0
File Listing

This page intentionally left blank.

SDWIS/LabToState 3.0 File Listing

```
labtoState.war              12/03/2010  02:29 PM        21,881,211
labtostate-embedded.zip     12/03/2010  02:26 PM       114,686,865

     Total Files Listed:
               2 File(s)    136,568,075 bytes
```

APPENDIX H

Tips for creating CSV Files

The following tips are from http://www.creativyst.com/Doc/Articles/CSV/CSV01.htm and maybe useful in understanding the creation of a CSV file.

- **Fields are separated with commas.**
  Example: *John,Doe,120 any st.,"Anytown, WW",08123*
- **Fields with embedded commas must be delimited with double-quote characters.**
  In the above example. "Anytown, WW" is delimited in double quotes because it has an embedded comma.
- **Fields that contain double quote characters must be surrounded by double-quotes, and the embedded double-quotes must each be represented by a pair of consecutive double quotes.**
  Example: *John "The Man" Doe* must be represented as *"John ""The Man""",Doe, 120 any st.,...*
- **Leading and trailing space-characters adjacent to comma field separators are ignored.**
  Examples: *John , Doe ,...* resolves to "John" and "Doe", etc. Space characters can be spaces, or tabs.
- **In most instances, each line represents a single record and a line is delimited by a record separator.** A record separator may consist of a line feed (ASCII/LF=0x0A), or a carriage return - line feed pair (ASCII/CRLF=0x0D 0x0A).

- **A field that contains embedded line-breaks must be surrounded by double-quotes**
  Example:

> **Field 1:** Conference room 1
> **Field 2:**
> John,
> Please bring the M. Mathers file for review
> -J.L.
> **Field 3:** 10/18/2002
> ...

would convert to:

> *Conference room 1, "John,*
> *Please bring the M. Mathers file for review*
> *-J.L.*
> *",10/18/2002,...*

Note that this is a *single* CSV record, even though it takes up more than one line in the CSV file. This works because the line breaks are embedded inside the double quotes of the field.

> *Implementation note: In Excel, leading spaces between the comma used for a field separator and the double quote will sometimes cause fields to be read in as unquoted fields, even though the first non-space character is a double quote. To avoid this quirk, simply remove all leading spaces after the field-separator comma and before the double quote character in your CSV export files.*

- **Fields with leading or trailing spaces must be delimited with double-quote characters.**
  Example: to preserve the leading and trailing spaces around the last name above: *John ," Doe ",...*
- **Fields may always be delimited with double quotes.**
  The delimiters will always be discarded.
- **The first record in a CSV file *may* be a header record containing column (field) names**
  There is no mechanism for automatically discerning if the first record is a header row, so in the general case, this will have to be provided by an outside process (such as prompting the user). The header row is encoded just like any other CSV record in accordance with the rules above. A header row for the multi-line example above, might be:
  Location, Notes, "Start Date", ...

APPENDIX I

Summary of SDWIS/LabToState 3.0 ADA Section 508 Testing

Technical Specification of §1194.22	Implementation	Screen Reader Test Result
EPA HIGH PRIORITY		
(a) A text equivalent for every non-text element shall be provided (e.g., via "alt", "long desc", or in element content).	(1) A text description has been implemented for the faucet image.	All images and graphical buttons located either in the navigation bar or the main pages were read by the screen reader.
(c) Web pages shall be designed so that all Information conveyed with color is also available without color, for example from context or markup.	SDWIS/LabToState contains no pages where color by itself conveys information.	Not Applicable.
(e) Redundant text links shall be provided for each active region of a server-side image map.	Not Applicable. SDWIS/LabToState does not use server-side image maps.	Not Applicable.
(f) Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.	Not Applicable. SDWIS/LabToState does not use server-side image maps.	Not Applicable.
(k) A text-only page, with equivalent information or functionality, shall be provided to make a web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only page shall be updated whenever the primary page changes.	Not Applicable.	Not Applicable.
(m) When a web page requires that an applet, plug-in or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with 1194.21 (a) through (l).	Not Applicable. SDWIS/LabToState does not require applets or plug-ins to interpret page content.	Not Applicable.
EPA MEDIUM PRIORITY		
(d) Documents shall be organized so they are readable without requiring an associated style sheet.	All pages are readable without a style sheet.	SDWIS/LabToState web pages were read without any problem when the style sheet is removed.
(g) Row and column headers shall be identified for data tables.	HTML tags for row and column headers were added	JAWS 10 successfully reads header column and row information, as expected.
(h) Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.	Tags were added for row and column headers.	JAWS 10 successfully reads header column and row information for each data cell, as expected.
(l) When pages use scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by assistive technology.	Not Applicable, SDWIS/LabToState does not use scripting	Not Applicable.
(n) When electronic forms are designed to be	Assistive technologies/screen readers such as IBM	Not Applicable.

Technical Specification of §1194.22	Implementation	Screen Reader Test Result
completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.	Home Page Reader was used to verify that the information.	There are no electronic forms to be filled on-line in this website.
(o) A method shall be provided that permits users to skip repetitive navigation links.	Not Applicable, the navigation links are unique for each page.	Not Applicable.
EPA LOW PRIORITY		
(b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.	Not Applicable. SDWIS/LabToState does not have nor use multimedia content.	Not Applicable.
(i) Frames shall be titled with text that facilitates frame identification and navigation.	Two frames are rendered, a navigation and application frame, and both frames are identified appropriately.	The screen reader identifies the start and the end of each of the two frames, as expected.
(j) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.	Not Applicable. SDWIS/LabToState does not have content that will cause the screen to flicker.	Not Applicable.
(p) When a timed response is required, the user shall be alerted and given sufficient time to indicate more time is required.	Not Applicable. SDWIS/LabToState does not require timed responses.	Not Applicable.

APPENDIX J

Recommendations for Making SDWIS Applications Deployed on External Facing Server More Secure

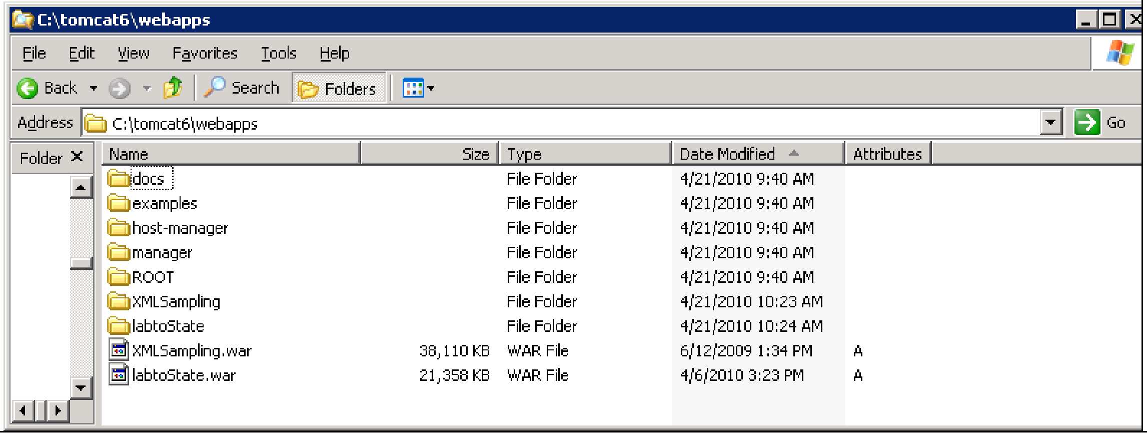
RECOMMENDATIONS FOR MAKING SDWIS APPLICATIONS DEPLOYED ON EXTERNAL FACING SERVER MORE SECURE

If your agency is deploying SDWIS applications (e.g., SDWIS/DWW, SDWIS/LabToState) from outside your firewall (e.g. on an external facing server), consider the following basic recommendations to make your environment more secure.

	Recommendation	**Description**	
	Use Strong Passwords (Oracle, SQL Server, Tomcat Users etc…)	Do not use same username as password Do not use common words, names of spouses, kids etc… Use a combination of uppercase, lowercase, alpha and numeric and special characters. The longer the length the better (e.g., 15 characters) *In your Tomcat_Home\conf\tomcat-users.xml file, please ensure you have changed the password for the user with admin and/or manager role. If you have left the default username/password as admin/admin from a default installation, your application server is vulnerable. While those usernames with a SDWIS role (in the tomcat-user.xml file), can only be 8 digits in length, use a longer and strong password that is different from the username.* *Ensure that the DBMS Administrator/System password is not a standard password (i.e., administrator, server, oracle, manager, system, etc…)*	
	Stay Current with Web and App Server Patches	http://tomcat.apache.org/whichversion.html It is good practice to check for the latest version of your Application and/or Web Server and apply this to your server. Newer releases often tighten up security holes and resolve memory leaks. Only upgrade to release versions of the software not to beta versions.	
	Consider using Apache as Web	Consider adding Apache as additional layer to your Application Server (4-Tier Architecture) or implementing similar architecture that seamlessly redirects a user-	

	<u>Server</u>	supplied url to the location where SDWIS/LabtoState resides. Adding this front end, prevents your application server URL from being exposed. For example, using Apache, all calls to SDWIS products would flow to the Apache Web Server, which then would "redirect" the url to your Application server (seamlessly to the user). *To download Apache 2.2:* • *Go To: <u>http://www.apache.org/dist/httpd/binaries/</u>* • *Select your operating environment:* *For Example under win32/ select either* • *httpd-2.2.15-win32-x86-no_ssl.msi (to install without SSL)* • *httpd-2.2.15-win32-x86-openssl-0.9.8m-r2.msi (to install with SSL capability)* *After installation: In the httpd.conf file under C:\Program Files\Apache Software Foundation\Apache2.2\conf, uncomment the following:* *LoadModule proxy_module modules/mod_proxy.so* *LoadModule proxy_http_module modules/mod_proxy_http.so* *LoadModule rewrite_module modules/mod_rewrite.so* *LoadModule ssl_module modules/mod_ssl.so (only uncomment if you installed Apache using .msi with SSL)* *Add the following lines to the very end of the file (not within in a tag ---just at the end of the file by itself)* *SSLProxyEngine on (only include this line if you installed Apache using .msi with SSL)* *ProxyPreserveHost on*	

		RewriteEngine on _RewriteRule ^/DWW$ /DWW/ [R,L]_ _RewriteRule ^/DWW/(.*) http://AppServer:PortNumber/DWW/$1 [P,L]_ _RewriteRule ^/labtoState$ /labtoState/ [R,L]_ _RewriteRule ^/labtoState/(.*) AppServer:PortNumber/labtoState/$1 [P,L]_ _For added security add the following lines to the end of the file:_ _Timeout 45_ _TraceEnable off_ _LimitRequestBody 1048576_	
	Consider running App and Web Server using SSL	You can install Tomcat and Apache with Secure Sockets Layer (SSL). SSL establishes an encrypted link between the web server and browser, ensuring that the data passed remains private and secure. _When opening up your firewall, generally port 443 is used for SSL (i.e., https) and port 80 is used for http requests._	
	Consider changing the Tomcat default port	Default Tomcat installations use 8080 as the port number. Consider changing this default port number to another (less obvious) number. You can change the port number during installation or in the server.xml file located under Tomcat_Home\conf\server.xml Change the HTTP Connector port to an unused port number other than 8080 (e.g.,9080): <Connector port="8080" protocol="HTTP/1.1"	
	Consider changing the shutdown command	By default, Tomcat can be shut down by connecting to Tomcat on port 8005 (default) and sending the following character sequence	

		SHUTDOWN You might want to change the character sequence and port number to another (less obvious) number and text string. (e.g. <server port="8098" shutdown="goingdown">)	
	Remove default applications	A default Installation of Tomcat contains a number of web applications which are installed and ready to run • ROOT: contains the simple default welcome page • docs: Tomcat documentation • examples: Examples of JSPs and servlets demonstrating Tomcat • manager and host-manager: Two powerful system applications to make administrating virtual hosts and the Tomcat servers more convenient *manager and host-manager present the greatest security risk. If you are running on an external facing server or if you do not use the Tomcat Manager functions to administer your application, consider removing these folders from your production environment.*  C:\tomcat6\webapps File Edit View Favorites Tools Help Back Search Folders Address C:\tomcat6\webapps Go Folder Name / Size / Type / Date Modified / Attributes docs / File Folder / 4/21/2010 9:40 AM examples / File Folder / 4/21/2010 9:40 AM host-manager / File Folder / 4/21/2010 9:40 AM manager / File Folder / 4/21/2010 9:40 AM ROOT / File Folder / 4/21/2010 9:40 AM XMLSampling / File Folder / 4/21/2010 10:23 AM labtoState / File Folder / 4/21/2010 10:24 AM XMLSampling.war / 38,110 KB / WAR File / 6/12/2009 1:34 PM / A labtoState.war / 21,358 KB / WAR File / 4/6/2010 3:23 PM / A	

	Run Tomcat (or web/app server) under its own account Securing the File system	By maintaining a dedicated network user account (for starting Tomcat or web/app server), you can limit operating system privileges. By using a Tomcat (or web/app server) account and making sure the file permissions are set correctly (restrict those folders not used by SDWIS applications) you reduce the risk of affecting other file folders if Tomcat is compromised. *For those folders that are receiving uploaded files, (e.g., Tomcat_Home\webapps\labtoState\jobFolders) you should see that your virus scan targets this directory to detect compromised files.*	
	Configure SDWIS/LabToState with robust UserID and Passwords.	SDWIS/LabtoState delivers an “out of the box” solution for initial start up and testing. which includes an XML document with example UserID and Password, role, PIN number, and laboratory. You should change these example UserIDs, Passwords, PINs, etc. to follow more robust guidelines as discussed in item 1. You can make the necessary changes in userList.xml file located under Tomcat_Home\webapps\labtoState\WEB-INF\classes\properties You may also consider developing your own plug-in for password authentication.	
	Limit IP addresses that can target your SDWIS/LabToState installation	To ensure that only approved SDWIS/LabToState users can access your SDWIS/LabtoState installation, consider working with your IT department to apply firewall rules that only accept incoming requests to your SDWIS/LabToState application from the IP addresses of your users.	