Cancer Trials Support Unit

CTSU – A Service of the National Cancer Institute

CTSU SSO (Java) Installation and Integration Guide

Revision 1.0

18 October 2011

Document Information

Revision Information for the CTSU SSO Framework Installation Guide

Revision History			
#	**Date**	**By**	**Description**
01	24-Aug-2011	P Whitefield	Guidelines on how to install and integrate CTSU SSO Framework to application using SAML for login and logout.
02	14-Oct-2011	Jayan Nair	Reviewed and Updated
Last Saved on 10/18/2011 6:26:00 PM			
File Location: **\\westat.com\DFS\CTSU8339\Tasks\8339_14_CDMS\07_IT\Releases\CTSU_SSO\Java\1.0\Documents\pdfs\CTSU_SSO_Java_StarterKit_Installation_V1.0.docx**			

This document was prepared for:

Template Help:

Use the **File→Properties** option to update footer and key values in document.

This document was prepared by:

WESTAT
1650 Research Boulevard
Rockville, Maryland 20850   Phone: (301) 251-1500

Phoebe Whitefield   *(301) 314-2448*

Contents

Figures

Tables

1. Introduction

1.1 Overview

As part of the integration of Rave as the CDMS (Clinical Data Management System) for the Cooperative Groups and other lead organizations, a CTEP-IAM based Single Sign-On (SSO) authentication system is planned to be used for authenticating users by the existing web based systems at the Cooperative Groups. The CTSU SSO authenticator implements Security Assertion Markup Language (SAML) for login authentication.

To enable the incorporation of this SSO federated authentication system to existing Cooperative Group applications, CTSU created a SSO software framework and a starter kit to integrate CTEP IAM SSO authenticator to existing JAVA and .NET applications of the cooperative groups.

This document details the installation and usage instructions for the CTSU SSO Framework for Java. A similar document will be available for .NET developers for using the CTSU SSO Framework for .NET.

The CTSU SSO framework provides the following advantages:

1. Object oriented approach by providing a class based implementation to hide the SAML complexity.
2. Simple APIs to construct SAML request.
3. Simple APIs to extract SAML response.
4. Ability to verify the digital signature of the Identity Provider.
5. Optional roster data integration for authorization.
6. Two options for integrating SSO to the existing web applications
   a. iFrame Based Approach: In this approach the developers can embed the CTEP IAM login content to the existing login page. The displayed IAM login content can be customized by using a CSS file integrated with the implementation.
   b. Page Redirection Based Approach: In this approach when the users login to the existing login page, they will be redirected the URL of the Identity Provider (CTEP-IAM) and after they successfully login the user will be sent back to the welcome page of the original system they tried to access. Ability to replace the CTEP IAM image header with cooperative group's specific image header is also incorporated with this implementation.

1.2 Terminologies

1. IdP: Identity Provider – It is a centralized system that creates, maintains, and manages identity information for principals (users) and provides primary authentication to other service providers within a federation.
2. SAML: Security Assertion Markup Language is an XML-based open standard for exchanging authentication and authorization data between security domains, that is, between an identity provider (a producer of assertions) and a service provider (a consumer of assertions).
3. SP: Service Provider – A system that provides services (such as patient enrollment service or clinical data collection service) to the end users.
4. SSO: Single Sign-On – With Single Sign-On a user logs into one system once and gains access to all systems within a federation without being prompted to log in again at each of them.

1.3 Purpose

The purpose of this document is to illustrate the following:

1. How to install the SSO framework or the developer's work environment.
2. How to integrate the SSO framework to existing application that requires SAML as the

authentication mechanism.

1.4   Audience

This document is intended for the development team at the various Cooperative Groups and Lead Organizations.

1.5   How to Use this Document

This section provides the logical steps that can be followed to integrate the CTSU SSO framework into the cooperative groups' application.

1. Review this document as well as the companion document CTSU_SSO_API_Java_Design_V1.0.docx which describes the CTSU SSO framework architecture and design principles.

2. Access the CTSU public URL (https//test.ctsu.org/ctsusso) to test out the starter kit functionalities.

3. Download and unzip the ctsusso.zip file on your development machine.

4. Build and deploy the starter kit without modification.

5. Invoke the starter kit UI located in your deployment machine to test the installation.

6. Integrate the CTSU SSO starter kit your web application.

2. Requirements to use CTSU SSO Starter Kit

This section contains the requirements to run the SSO starter kit and to integrate the SSO starter kit with the cooperative groups' systems.

2.1   Software Requirements

A Java servlet container such as JBoss or tomcat and Java runtime environment are required to run the CTSU SSO framework (Java). Table 1 shows the required software.

Software	Version	Description	URL
Java	JDK 1.6	Java Development Kit	http://www.oracle.com/technetwork/java/javase/downloads/index.html
Eclipse	3.0 or higher	An open platform integration tool that provides developers with flexibility and control over their software code	http://www.eclipse.org/downloads/index.php
JBoss	4.0.4 or higher	Open source Java servlet container	http://www.jboss.org
Tomcat	1.6 or higher	Open source Java servlet container	http://tomcat.apache.org/download-60.cgi

Table 1: Software Requirements

2.2   Publicly Accessible Files for Display Customization

As stated in the overview, there are two approaches to integrate the CTEP IAM login page to the cooperative groups' application. This section defines the optional files needed to run the iFrame and page redirection based implementations.

2.2.1   iFrame Based Implementation

In this approach, the CTEP IAM login page is embedded within the cooperative groups' login page. CTEP IAM login content can be customized using the cooperative groups' CSS file. This CSS file has to be deployed on a public URL. Please note that a CSS template is provided with the SSO starter kit and the styles in this template can be customized and hosted by the organization to get a custom look and feel for the iFrame based implementation. The CSS class names should not be changed. The CSS style

specifications can be changed to get the appropriate look and feel.

A default CSS provided by CTSU is available for the cooperative group to define the appearance of the CTEP IAM login UI.

2.2.2   Page Redirection Based Implementation

In this approach when the users access the existing login page, they will be redirected to the URL of the Identity Provider (CTEP IAM). On the CTEP IAM login page to which the user is redirected, the header image can be replaced by the cooperative groups' header image for preserving the user experience. The following specifications are applicable for the header image:

1. The image file has to be of jpg format.
2. The image height is to be 100px and the width is to be 525px.
3. The image file should be deployed to a public URL.

A CTSU image jpg file is provided to the cooperative groups to use as the template for generating their own image file.

2.2.3   CTEP Login Account

To utilize the CTSU SSO framework, a valid CTEP IAM account is required. This account should be created in CTEP IAM production site for logging into CTEP IAM production and CTEP IAM beta site for logging into CTEP IAM beta site. The CTEP IAM URLs are defined in section 2.2.1.

2.2.4   SAML Federation in CTEP IAM

The service provider is required to register with CTEP IAM as one of the authorized consumers of the CTEP IAM authentication service. The registration process will be provided in a separate document.

2.2.5   Identity Certificate

This certificate is required for receiving SAML response from CTEP IAM. This certificate validates the CTEP IAM SSO signature provided in the SAML response. The certificates for CTEP IAM beta and CTEP IAM production will be included in the ctsusso.zip file.

2.3   Required URLs

This section provides the CTEP IAM URL to create user account, CTEP IAM SSO authenticator URL, CTSU public CSS file URL and CTSU public header image file URL.

2.3.1   NCI CTEP IAM URLs

The following table contains the CTEP IAM SAML beta and production URLs.

#	Description	URL
1.	CTEP IAM user production site for creating user account.	https://eapps-ctep.nci.nih.gov/iam/index.jsp
2.	CTEP IAM SSO production site	https://iapps-ctep.nci.nih.gov:443/sso-web/singleSignOn.action
3.	CTEP IAM user beta site for creating user account	https://betapps-ctep.nci.nih.gov/iambeta/index.jsp
4.	CTEP IAM SSO beta site	https://betapps-ctep.nci.nih.gov:443/sso-web/singleSignOn.action

Table 2: CTEP IAM SAML URL

2.3.2   CTSU Public Files URLs

The following table contains the CSS URL for iFrame based implementation and the header image URL.

Description	URL
CTSU public CSS file	https://www.ctsu.org/ctsusso/css/miniLogin.css
CTSU public header image	https://www.ctsu.org/ctsusso/image/ctsulogo.jpg
CTSU SSO framework test URL	https://test.ctsu.org/ctsusso

Table 3: CTSU SSO URL

3. CTSU SSO Starter Kit Installation Instructions

This section provides the following details:

1. Steps for installing the SSO starter kit on the application development environment.
2. Steps to verify the installation and the requirements to utilize CTEP IAM authentication.
3. Required Jars for JBoss version 4.
4. Required Jars for JBoss version 5.
5. Required Jars for tomcat version 6.

NOTE: CTSU currently supports JBoss version 4, 5 and tomcat version 6.

3.1 Steps for Installation

3.1.1 Assumption: one has active CTEP IAM Account

If one does not already have CTEP-IAM Account, please create an account using CTEP IAM production and beta URLs as per items 1 and 3 listed in Table 2 above.

3.1.2 Download and Unzip

(1) Download CTSU SSO Framework and starter kit (ctsusso.zip) from this URL:
https://www.ctsu.org/ctsusso/default.asp?fName=ctsusso/java/ver_1.0

(2) Unzip ctsusso.zip file. When you unzip, the directory structure should look like the figure shown below.
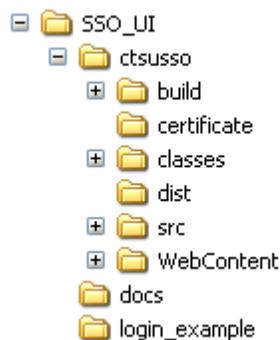
Figure 1: Folder structure of CTSU SSO package

3.1.3 Setting up Project in Eclipse IDE

The following steps show how to build the CTSU SSO starter kit project within Eclipse:

1. Create a new Java project using Eclipse-Europa. The steps below outline how to create the project
    a. From the main workbench window, click File->New->Java Project.

Input the value shown below. (Value in the **Directory** field should be the path to the extracted folder
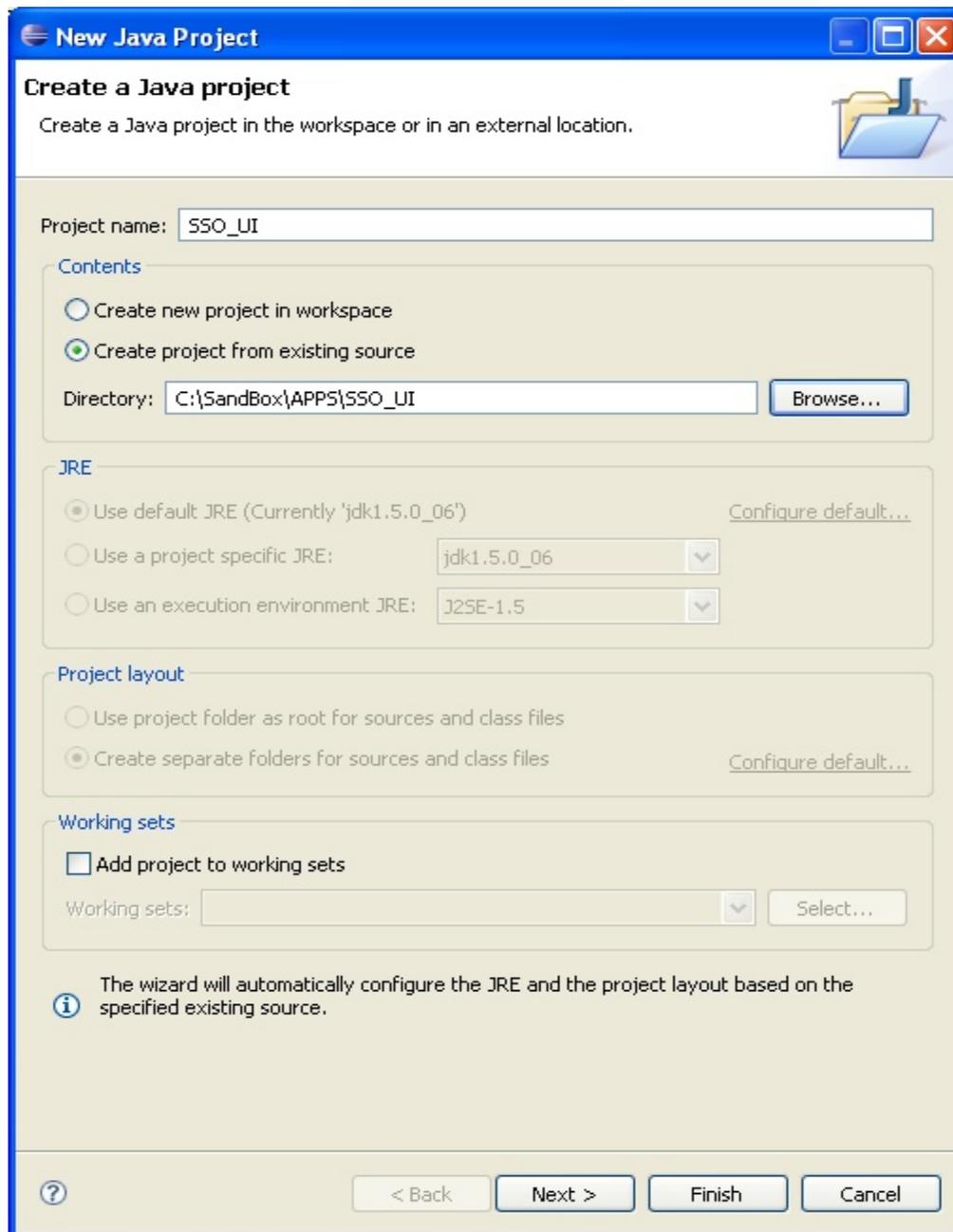
SSO_UI).

Figure 2: Create project in Eclipse

2. Use the build.xml to build the project. The steps below outline how to run the build.xml

    a. Expand the project path to locate the build.xml. The build.xml path is defined in Eclipse's Navigator tab as SSO_UI->test->build.xml.

    b. Right click on build.xml->Run As-> Ant Build.

    c. Select the build target as shown in Figure 3
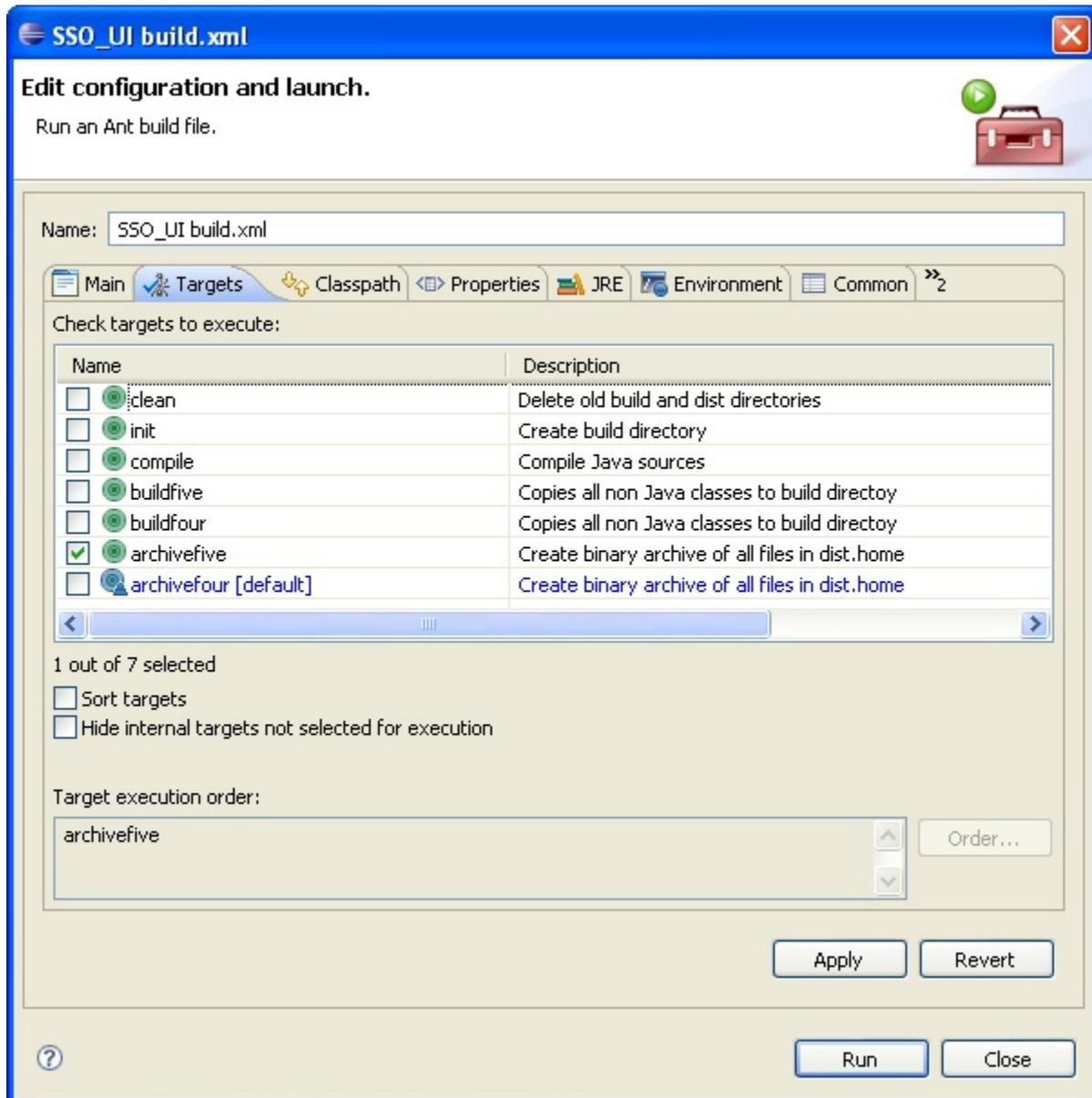
Figure 3: Build Target in Eclipse

3. Verify that the war file is deployed to the JBoss 5 deploy directory as specified in **build.properties**. (shown in Table 4 below). Note: the build target "archivefive" builds JBoss 5 war, and the build target "archivefour" builds JBoss 4 war.

4. The war is deployed to the servlet container during the build process if using the build.xml provided in the code. Before starting the JBoss server, Please confirm that ctsusso.war is deployed to the servlet deployment directory.

5. Run/Start JBoss 5. (Check the jar files required at section 3.3)

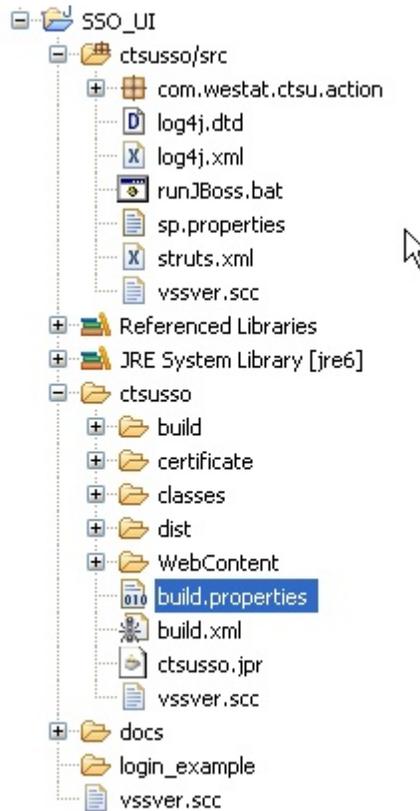The **build.properties** file is located as shown in the image below.

Figure 4: Location of build.properties in eclipse workspace

The table below shows the definition of elements in build.properties. One needs to change the value of **deploy.dir4/ deploy.dir5** according to their local JBoss deployment folder configuration.

Property Name	Description	Value
app.name	Name of the war file.	ctsusso
deploy.dir4	JBoss 4 deploy directory where the war is copied to. The value should be modified to reflect the server location.	C:/Servers/jboss-4.0.5.GA/server/default/deploy [Note: This value needs to be changed based on your JBoss configuration]
deploy.dir5	JBoss 5 deploy directory where the war is copied to. The value should be modified to reflect the server location.	C:/Servers/jboss-5.1.0.GA/server/standard/deploy [Note: This value needs to be changed based on your JBoss configuration]
source.home	Java source files directory relative to the build.xml file directory	./src

lib.home	Jars directory relative to the build.xml directory	./WebContent/WEB-INF/lib
webapp.home	Web content directory relative to the build.xml directory. The web content directory has the jsp files, the graphic files, CSS files.	./WebContent
build.home	The relative directory to the build.xml where all the files that make up the war are located.	./build
dist.home	The relative directory to the build.xml directory where the war file is located.	./dist

Table 4: build.properties Definition

3.1.4   Identity Certification Installation

This section describes how to install the certificate for testing starter kit and for integrating the certificate to the cooperative groups' application.

3.1.4.1   CTEP IAM Certificate Installation

After the certificate is extracted from ctsusso.zip, please install this certificate on your server machine. The certificate file directory depends on the ${CTSU_HOME} environment system variable. The full path to the certificate should be ${CTSU_HOME}/CTSUSSO/certificate/IAMSAML.cer. The CTEP IAM beta site uses IAMSAML.cer. The production certificate will be provided.

Note: If the certificate is not installed in the cooperative groups' server machine during integration, or during starter kit installation testing, an CertificateException will be thrown.

3.2   Verification

This section describes how to verify the CTSU SSO starter kit is installed without any error, the SP is registered with CTEP IAM and the user has CTEP IAM account in CTEP IAM production and/or in CTEP IAM beta.

The starter kit provides an UI as a tool to perform the verification. The steps below outline the UI functionalities:

  a. To verify that the installation is completed without any errors.
  b. To test the SSO functionalities.
  c. To test the CTEP IAM SSO authenticator to make sure it is responding correctly with the login request options such as returning CTSU person roster as requested.
  d. To display the CTEP IAM login page content in either iFrame based or page redirection based implementation.

The verification process assumes that the starter kit is extracted and installed correctly as in Section 3.1 and the JBoss 5 is the Java servlet container.

3.2.1   CSS and Custom Image Header Setup

The CSS file or the image header jpg file has to be deployed to a publicly accessible URL since these files will be read by the CTSU SSO implementation. CTSU provides default CSS and image header if the cooperative groups choose not to create these files.

3.2.2 Verification steps

After the completion of the steps described, the starter kit is ready for verification. The starter kit can be invoked in Internet Explorer, Chrome and Firefox. The verification process includes following steps:

1. Invoke the SSO starter kit UI.
2. Send in login request to CTEP IAM production and beta URL.
3. Verify the login response from CTEP IAM.

Please see below verification steps.

3.2.2.1 Invoking the UI

The URL to invoke the UI is http://ServerName:ServerPort/ctsusso where the ServerName is the name of the server machine such as localhost, and the ServerPort is the port which the Java servlet container is listening to such as 8080. When the UI is invoked the webpage should appear as shown in, Figure 4.
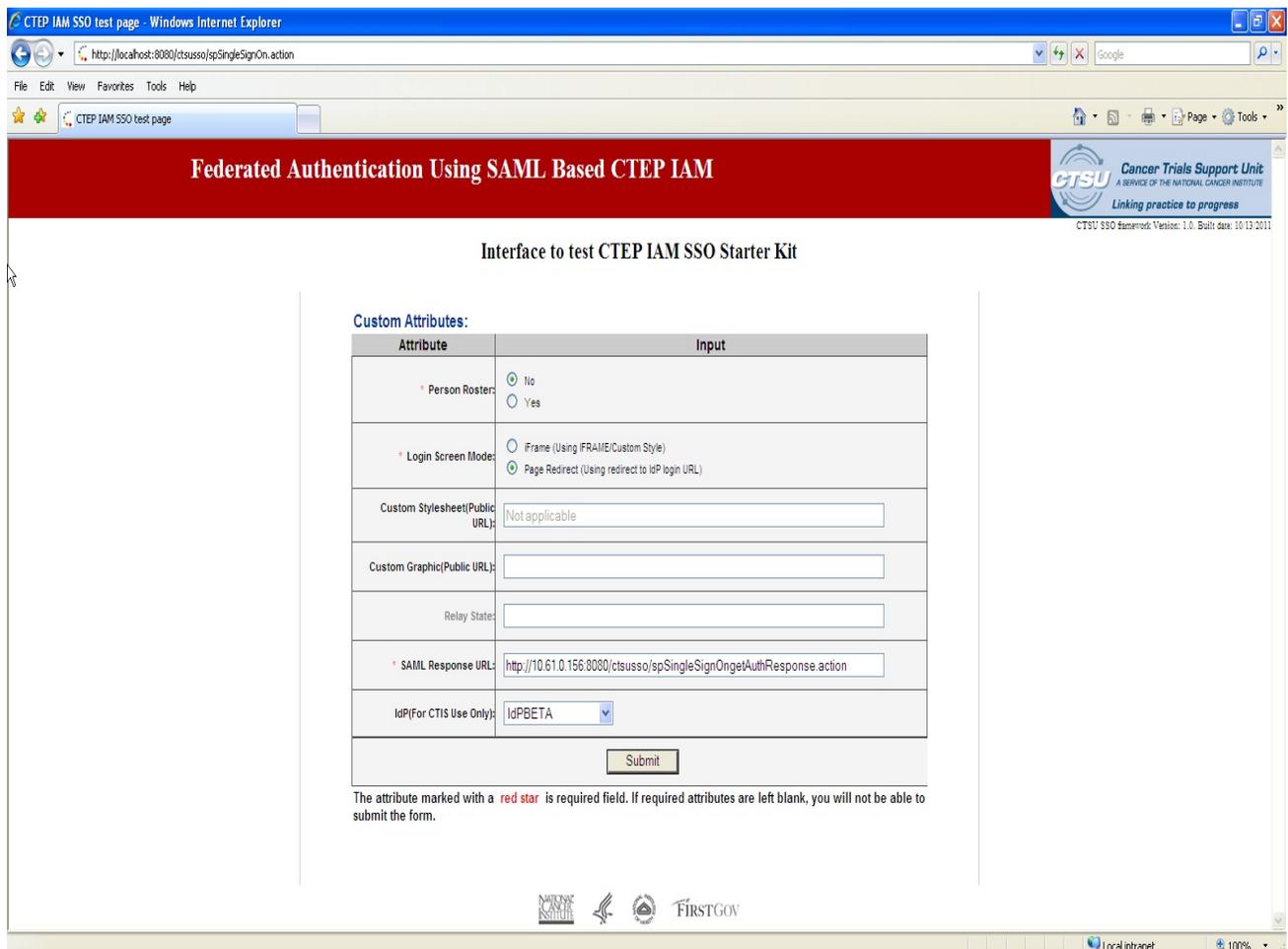

Figure 4: Starter Kit test user Interface page

The table below (Table 5) contains the custom attribute definitions that are listed in Figure 4.

Attribute	Meaning	Value to be selected
Person Roster	Cooperative group requests the CTSU person roster information to be returned from CTEP IAM for the login user after successful login.	Yes
Person Roster	The CTSU person roster information is not requested to be returned.	No
Login Screen Mode	Use page redirection based implementation where the browser is redirected to CTEP IAM login URL.	Page Redirect
Login Screen Mode	Use iFrame based implementation.	iFrame
Custom Style Sheet	Specify the CSS URL. This is used in conjunction with iFrame based implementation.	Example: https://www.ctsu.org/ctsusso/css/miniLogin.css
Custom Graphic	Specify the image header jpg that replaces the CTEP IAM header.	Example: https://www.ctsu.org/ctsusso/image/ctsulogo.jpg
SAML Response URL	Response URL invokes by CTEP IAM SSO authenticator to return the SAML response. This is defaulted to the deployed server IP address appended with "ctsusso/spSingleSignOngetAuthResponse.action".	http://localhost:8080/ctsusso/spSingleSignOngetAuthResponse.action
IdP	Use CTEP IAM beta site as SSO authenticator.	IdpBETA
IdP	Use CTEP IAM production site as SSO authenticator.	IdpPRODUCTION

Table 5: Attribute Definitions for the test UI

3.2.2.2 SSO Login Request

Please input the custom attribute values as in Figure 5 using the above UI and click on the submit button. Figure 6 should display in the browser.
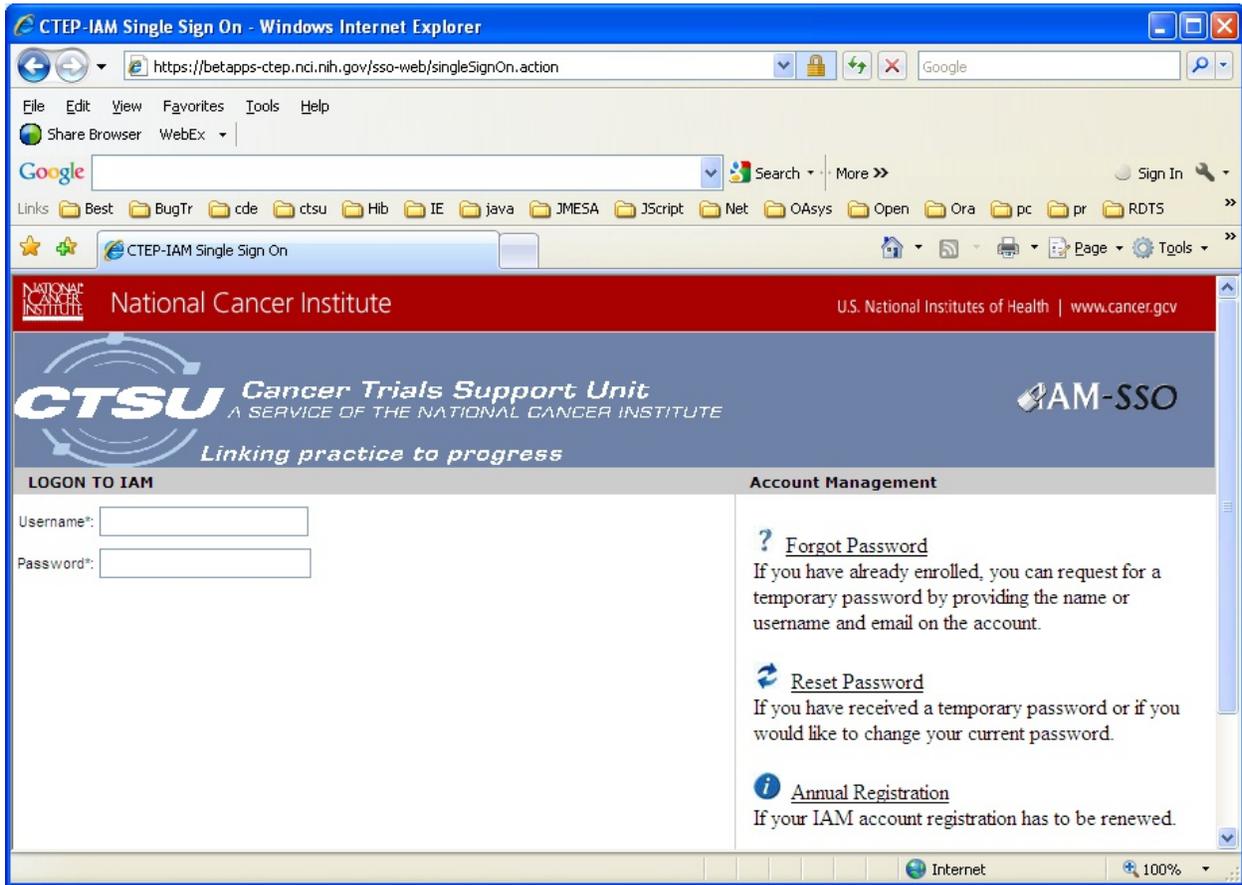
Figure 5: CTEP IAM Login with Page Redirect (Using redirect to IdP's login URL)
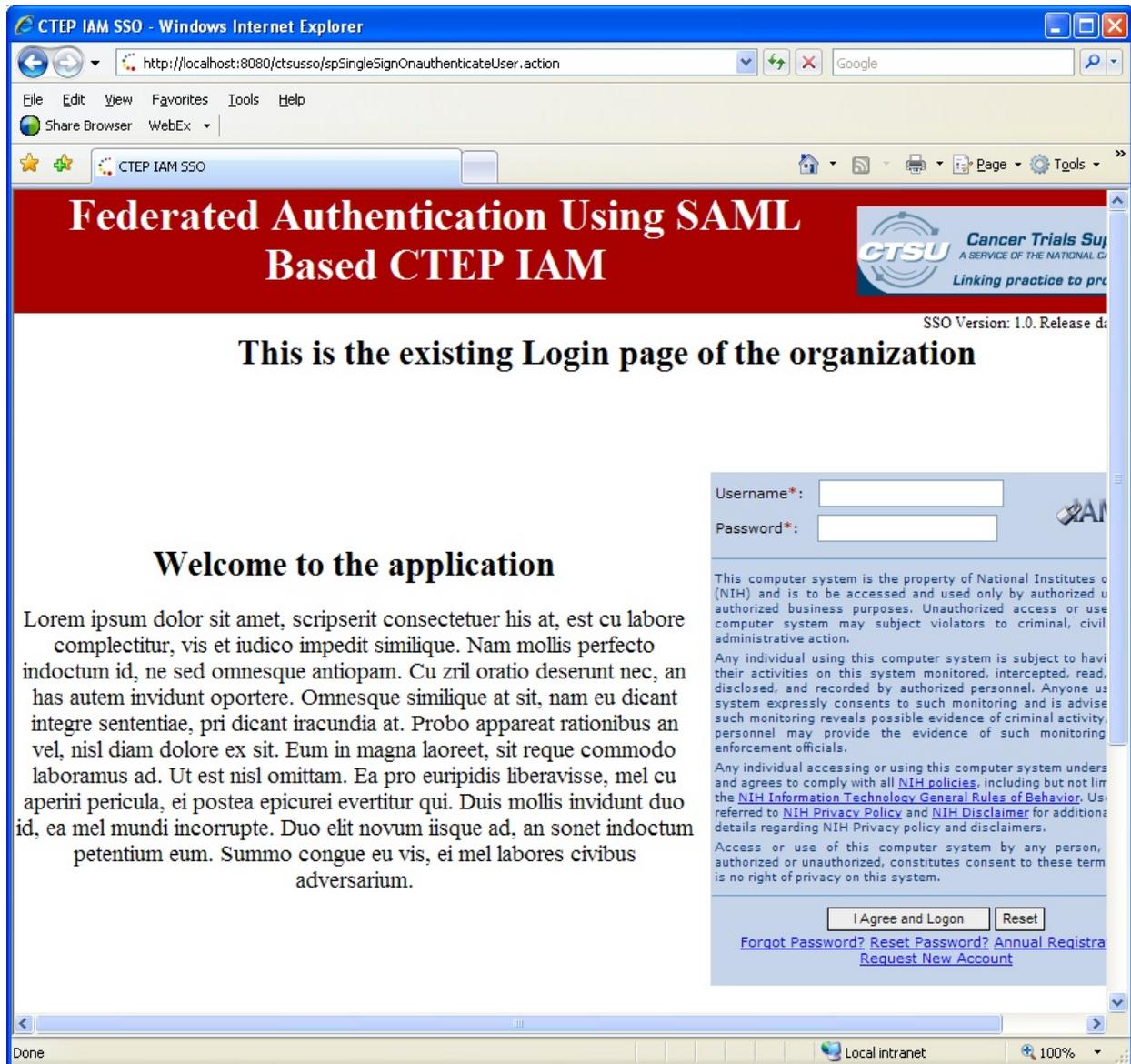
Figure 6: CTEP IAM Login with iFrame embedded within the existing application

3.2.2.3   SSO Login Response

Please input the CTEP IAM credential created in CTEP IAM beta and login. The following Figure 7 should display in the browser after successful login. This page contains the SAML response. This can be used to verify the response from CTEP IAM system.
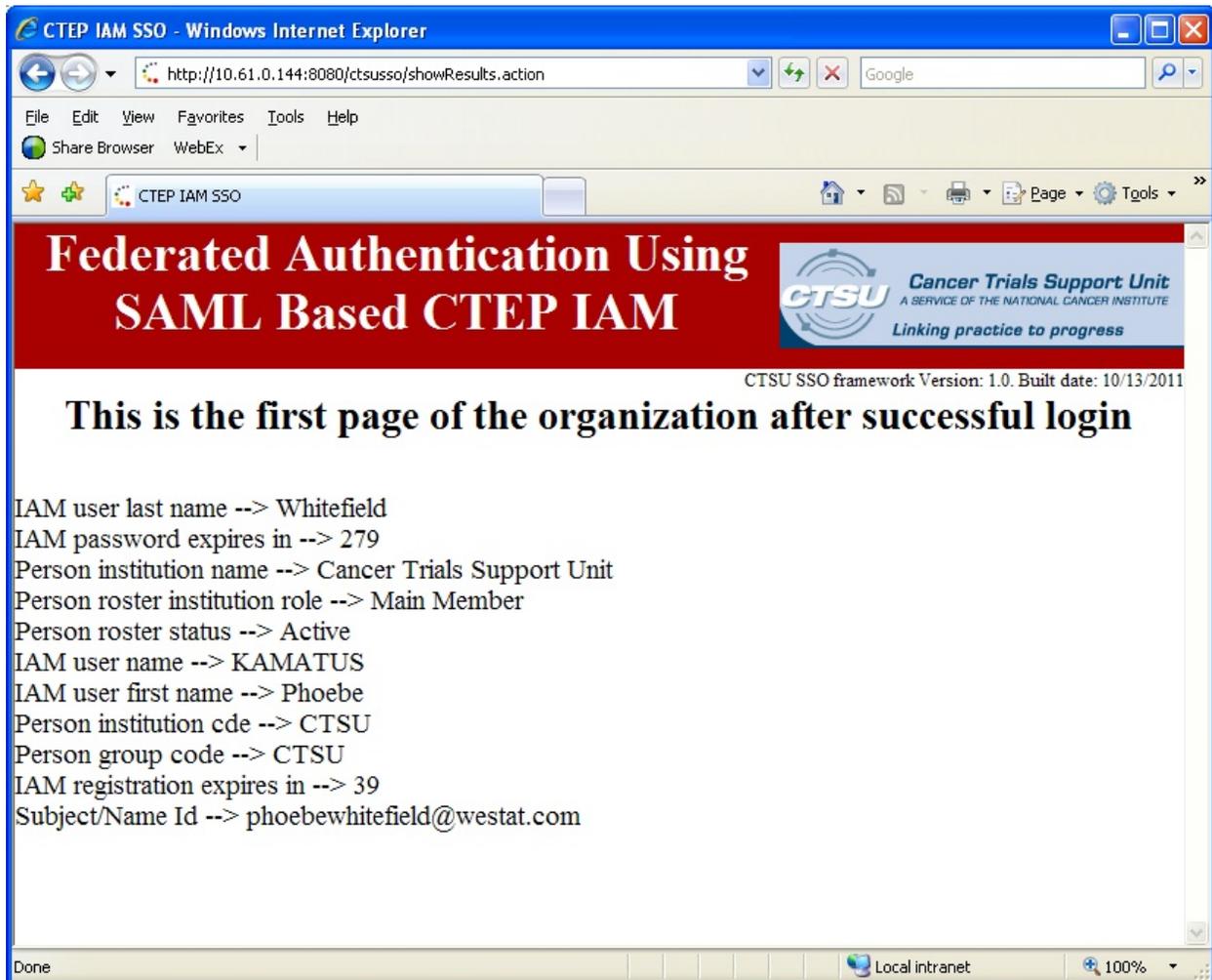
Figure 7: SAML Response with Successful Login

3.3   Jars for JBoss

This section lists the jars required to run the SSO starter kit. There are a few additional jars required to run in JBoss 4 which are listed separately.

3.3.1   Common Jars for JBoss

The figure below shows the common required JBoss jars. The jars should be installed with the ctsusso.jar.

```
bcprov-ext-jdk15-1.40.jar
commons-collections-3.1.jar
commons-collections-3.2.1.jar
commons-collections-3.2.jar
commons-fileupload-1.2.1.jar
commons-io-1.3.2.jar
commons-lang-2.4.jar
commons-logging-1.1.1.jar
freemarker-2.3.16.jar
javassist-3.7.ga.jar
joda-time-1.6.jar
jstl-1.2.jar
log4j-1.2.14.jar
ognl-3.0.jar
opensaml-2.2.3.jar
openws-1.2.2.jar
slf4j-api-1.6.1.jar
slf4j-log4j12-1.6.1.jar
struts2-core-2.2.1.jar
velocity-1.5.jar
xalan-2.5.0.jar
xmlsec-1.4.2.jar
xmltooling-1.2.0.jar
xwork-core-2.2.1.jar
```

Figure 8: Common JBoss Jars

3.3.2 Additional Jars for JBoss 4

The figure below shows the additional jars required to run JBoss 4. The jars should be installed to the same directory as ctsusso.jar.

```
servlet.jar
xercesImpl-2.9.1.jar
xml-apis.jar
```

Figure 9: JBoss Version 4 Jars

4. SSO Framework Integration

This section illustrates how to integrate the CTSU SSO framework to a generic web application that requires login credential using CTEP-IAM SSO.

4.1   Group Implementation Steps

This section highlights the steps that need to be taken to integrate the starter kit into an existing website.

- Install the IdP certificate and CTSUSSO.jar/dll
- Instantiate the CTEPSSOAuthenticator object
- Construct the CTEPSAMLRequest attributes
- Invoke the sendLoginRequest with the CTEPSAMLRequest
- Wait for the request post to come back from the Idp

- Invoke the getSAMLResponse for the request
- Invoke the getSSOUser for the obtained CTEPSAMLResponse

4.2   Implementation Options

There are two examples covering the two options for integration as described below:

1. **iFrame based implementation:** In this implementation the CTEP IAM login page is embedded within the Group's web application's login page. This example does not replace the original login page, instead a section of the application login page will contain an iFrame which is provided to display the CTEP login information (without the NCI CTEP logo). The URL of the web browser will still be the URL of the login page of the group's web application.

2. **Page redirection based implementation:** Here CTEP IAM login page is the application login page. In this example the CTEP-IAM login page replaces the original login page content through a page redirection. The user browser is redirected to the CTEP IAM login URL and once validated will get redirected back to the Group's welcome page.

4.3   iFrame Based Implementation

This example assumes that the following pages already exist:

1. Login page – this page is invoked when a user requests service from the service provider.

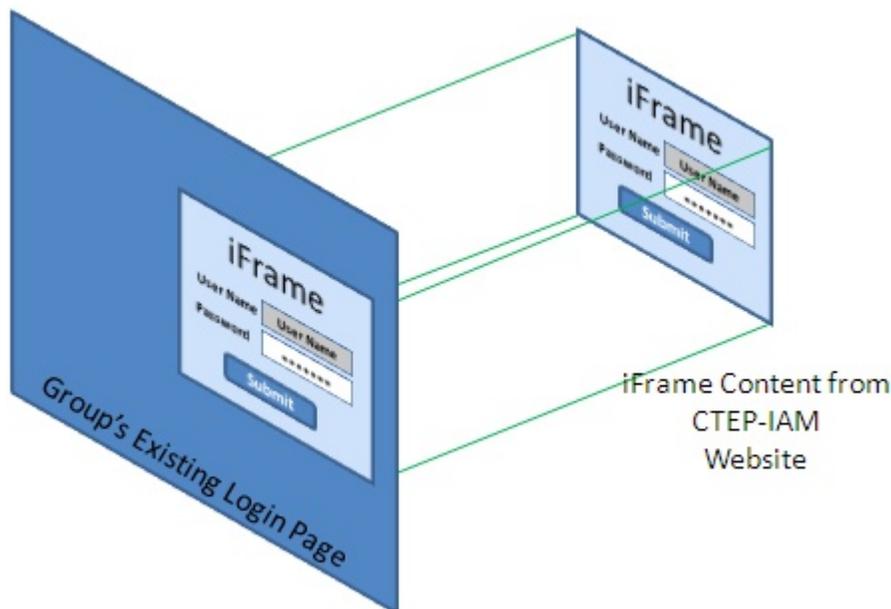2. Welcome page – this page is displayed after a user is successfully authenticated.

Figure 10: iFrame conceptual diagram

The above figure, (Figure 10) shows the concept of embedding an iFrame within a web application. The concept here is that the **iFrame** would be sitting in the place of the login area of the original web application of the group. The iFrame content will be coming from CTEP-IAM's website.

The following (Figure 11) shows current approach versus modified approach.

In the current approach, the user will enter the login details in the login page of the group's web application. Upon success, the user will be directed to the group's welcome page.

In the modified approach, the user will enter the login details in the iFrame which comes from the IdP. Here the user will enter their CTEP-IAM credentials. Upon success, the user will be directed to the
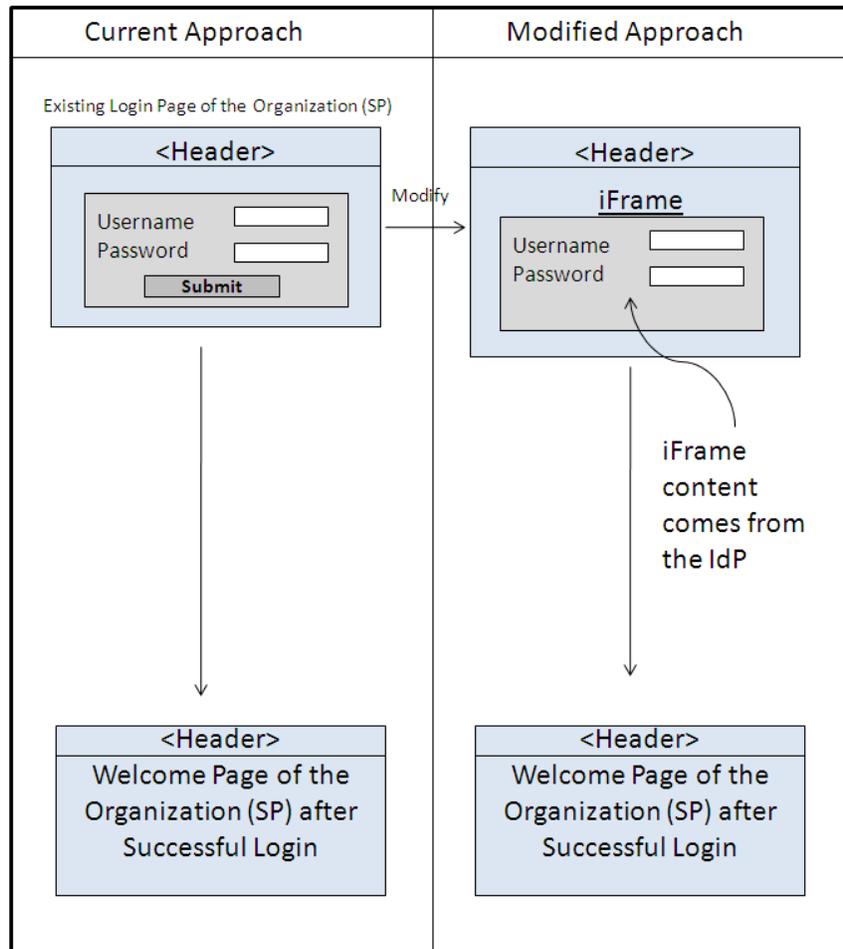
Group's welcome page.

Figure 11: iFrame Based Approach for SSO Starter Kit Integration

4.3.1   Login Modification

The login contains a jsp page and a login action backend code. The following sections explain the changes to the jsp page and the login action backend code.

4.3.2   Login jsp Modification

This example uses Form-based login authentication mechanism. The original login.jsp contains a UI section that requests the user name and password from the user. This section is no longer needed. It is replaced by an iFrame with the source that comes from the CTEP IAM SSO login URL after a post is made to the CTEP server. The following figures illustrate the code modification.

The code shown in Figure 12 is to be replaced by the code shown in Figure 14.

```
   <tr>
           <td align="right">IAM User: </td>
           <td align="left"><input type="text" id="username" name="username" class="formFieldSized"
size="25" /></td>
           </tr>
           <tr>
            <td align="right">Password: </td>
            <td align="left"><input type="password" id="password" name="password"
class="formFieldSized" size="25" /></td>
           </tr>

   <tr>
           <td colspan="2" align="center">
            <input type="submit" name="logon" value="I Agree and Logon" class="randoButtonGreen"
title='I accept the agreement. Log me in.'>
             <input type="reset" name="reset" value="Reset" class="randoButtonO" title='Clear my
entries'>
           </td>
           </tr>
```

Figure 12: Sample of the existing Login.jsp

Figure 13 shows the login.jsp user interface.

Figure 13: Sample of Login.jsp UI

```
<!-- src is the backend code. In this example, response.open is handled
     by LoginAction.java
-->
<iframe id="loginFrame" width="100%" style="width: 700px;height: 630px"
     src="<%=request.getContextPath()%>/response.open">
</iframe>
```

Figure 14: iFrame Based Login.jsp Modification

4.3.3 Login Action Backend Code Modification

The original login action backend code extracts the user credential and validates the user credential. This code is to be replaced by setting up the CTSU SSO framework to invoke the IdP. Figure 15 illustrates the CTSU SSO framework integration to the login action backend code.

```
//
// Create SAMLRequest object to hold the SAML information
//
CTEPSAMLRequest samlRequest = new CTEPSAMLRequest();

//
// IdP returns the SAML response to the below URL
//
samlRequest.setResponseURL("your URL");

//
// request is the HttpServletRequest
// response is the HttpServletResponse
//
samlRequest.setHttpRequest(request);
samlRequest.setHttpResponse(response);

//
// Below specify the IdP URL, please replace with your IdP URL
//
samlRequest.setIdpURL("https://iappsbeta-ctep.nci.nih.gov:443/sso-web/singleSignOn.action");

//
// Below specify who initiate this request, please replace with your
// issuer value
//
samlRequest.setIssuer("http://test.sp.com");

//
// Request CTSU Person Roster to be returned from CTEP IAM
//
samlRequest.setNeedPersonRoster(true);

//
// Use below setting for iFrame based implementation with provided CSS file.
//
samlRequest.setCustomCSSURL("https://www.ctsu.org/ctsusso/css/miniLogin.css");

//
// Invoke CTSU SSO to send SAML request
//
sso.sendLoginRequest(samlRequest);
```

Figure 15: Integrated CTSU SSO to Login Backend

4.3.4   CSS Style Sheet Creation

CSS file is needed for iFrame based implementation. The starter kit provides a CSS template for the iFrame based implementation. Please use this template to specify the CTEP IAM login iFrame style. The style class names should not be changed since these class names are agreed upon with CTEP IAM development team. Please only change the attribute values within the class.

4.4   Page Redirection Based Implementation

This example assumes that the following pages already exist:

1. Login page – this page is invoked when a user requests service from the service provider.

2. Welcome page – this page is displayed after a user is successfully authenticated.

The following figure (Figure 16) shows the modification steps for this approach. In the modified approach, when the user tries to login from the group's web application, It will be automatically redirected to IdP's login page (CTEP-IAM's login page). After the user enter authentication details and on success it would again be redirected to the group's welcome page.
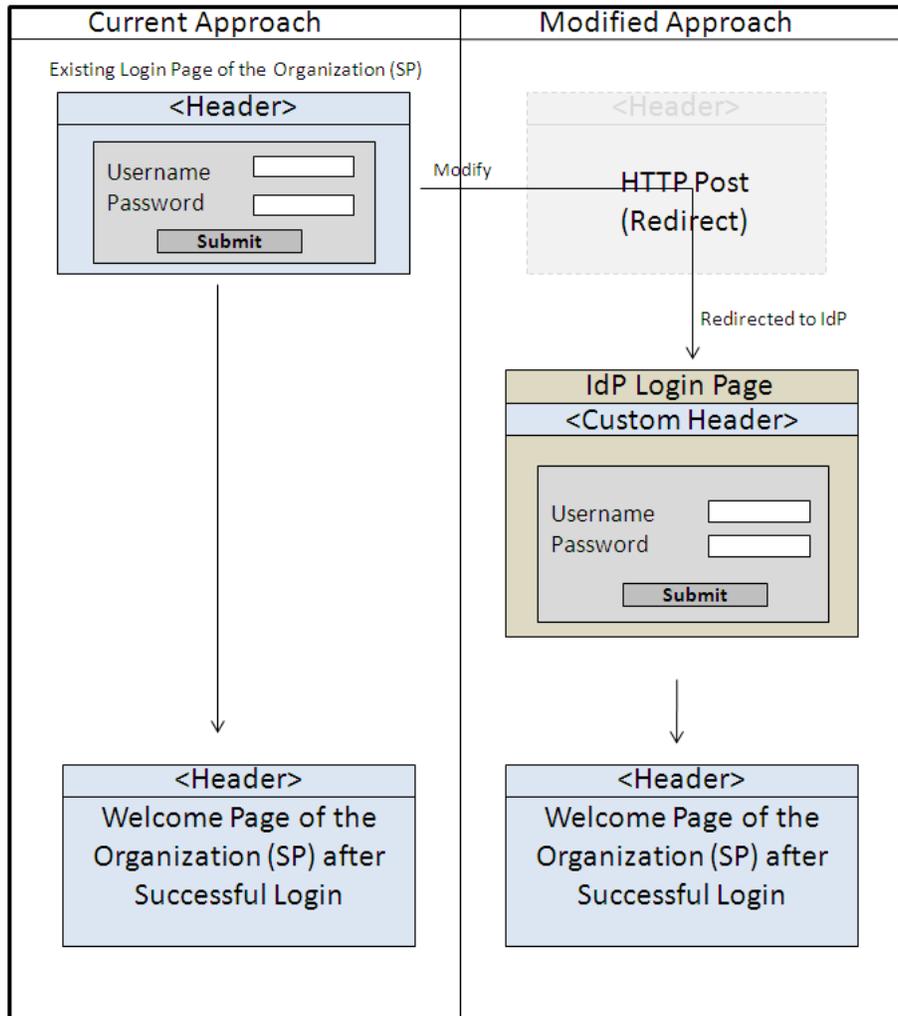

Figure 16: Page Redirection Approach for SSO Starter Kit Integration

4.4.1   Login Modification

The login contains a jsp page and a login action backend code. The following sections explain the changes to the jsp page and the login action backend code. Please note that the login page used in iFrame based approach will be used for this example.

4.4.2   Login jsp Modification

The original login.jsp contains a UI section that requests name and password from the user. This section is no longer needed. It is replaced by a submit call to the login action backend code when this page is loaded to the user's browser.

The code shown in Figure 12 is to be replaced by the code shown in Figure 17.

```
<script language="javascript" type="text/javascript">
   window.onload = function() {
              document.forms[0].submit();
            }
 </script>
```

Figure 17: Modified Login.jsp to Invoke CTSU SSO

4.4.3 Login Action Backend Code Modification

The original login action backend code extracts the user credential and validates the user credential. This code is to be replaced by setting up the CTSU SSO framework to invoke the IdP. Figure 18 illustrates the CTSU SSO framework integration to the login action backend code.

```
//
// Create SAMLRequest object to hold the SAML information
//
CTEPSAMLRequest samlRequest = new CTEPSAMLRequest();

//
// IdP returns the SAML response to the below URL
//
samlRequest.setResponseURL("your URL");

//
// request is the HttpServletRequest
// response is the HttpServletResponse
//
samlRequest.setHttpRequest(request);
samlRequest.setHttpResponse(response);

//
// Below specify the IdP URL, please replace with your IdP URL
//
samlRequest.setIdpURL("https://iappsbeta-ctep.nci.nih.gov:443/sso-web/singleSignOn.action");

//
// Below specify who initiate this request, please replace with your
// issuer value
//
samlRequest.setIssuer("http://test.sp.com");

//
// Request CTSU Person Roster to be returned from CTEP IAM
//
samlRequest.setNeedPersonRoster(true);

//
// Use below setting for redirect based implementation with provided header
// image jpg. Not needed for iFrame based or not to replace the CTEP IAM
// header
// Inform CTEP IAM to replace CTE IAM header with provided header image
// The image file should be jpg format and height = 145px and width = 755px
//
samlRequest.setCustomGraphicURL("https://www.ctsu.org/ctsusso/image/ctsulogo.jpg");

//
// Invoke CTSU SSO to send SAML request
//
sso.sendLoginRequest(samlRequest);
```

Figure 18: Integrated CTSU SSO to Login Backend code