

# Sun Portal Server 7.x

## Smart LDAP Login Channel

### For the Anonymous Page

**Author:** Jeff Nester  
Sun Microsystems  
*jeff.nester@sun.com*

**Version:** 1.1

**Date:** 10/11/2009



## *Table of Contents*

1	Introduction .....	3
2	Assumptions .....	4
3	Configuration Steps .....	4
3.1	Create the loginJSP Channel .....	4
3.2	Create the Portal Desktop Folder.....	8
4	Testing .....	8
5	Appendix – loginJSP.JSP .....	8

## Document Revisions

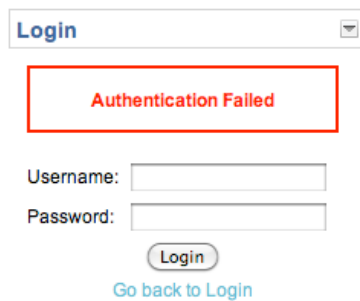
<b>Date</b>	<b>Editor</b>	<b>Description of Change</b>
10/11/09	Jeff Nester	Original Document Created

# 1 Introduction

The login provider on the Sun Portal Server 7.x anonymous desktop by default does not handle failure events. If an event like the user enters a bad password or the user's password has expired they are redirected to the Access Manager page that deals with the error. This document describes the steps and the code to create a JSPPProvider style login provider that handles the error conditions within the portlet.

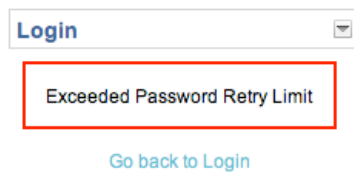
This provider has the ability to support the following conditions:

- 1) Authentication Failed due to a bad password or username



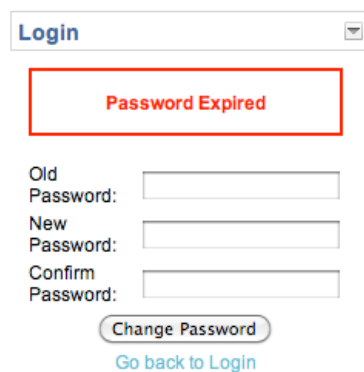
A screenshot of a login form. At the top is a portlet header with the text "Login" and a dropdown arrow. Below it is a red-bordered box containing the text "Authentication Failed" in red. Underneath are two input fields: "Username:" and "Password:". Below the password field is a "Login" button and a blue link "Go back to Login".

- 2) Password Retry Limit Exceeded



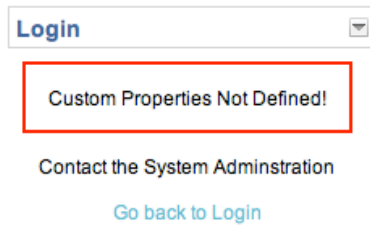
A screenshot of a login form. At the top is a portlet header with the text "Login" and a dropdown arrow. Below it is a red-bordered box containing the text "Exceeded Password Retry Limit" in red. Below the box is a blue link "Go back to Login".

- 3) Password Expired or Password Reset Required (The user can change their password from this channel)



A screenshot of a login form. At the top is a portlet header with the text "Login" and a dropdown arrow. Below it is a red-bordered box containing the text "Password Expired" in red. Underneath are three input fields: "Old Password:", "New Password:", and "Confirm Password:". Below the "Confirm Password" field is a "Change Password" button and a blue link "Go back to Login".

- 4) Missing customProperties (How to configure these properties is describe in this document)



## 2 Assumptions

- 1) The Portal Server is configured for anonymous desktop and the results of /portal/dt brings up the anonymous portal page that is being modified.
- 2) The User understands how to add channels to the anonymous desktop.
- 3) Assumes that LDAP is the login module being used.

## 3 Configuration Steps

The follow steps describe the configuration steps to add the new loginJSP channel to the Sun Portal Server.


### 3.1 Create the loginJSP Channel

- 1) Log into the Sun Portal Server PSConsole.
- 2) Select the portal
- 3) Select the DN for the anonymous user's desktop that you wish to add the capability too and add a JSPProvider channel. This can be added using either the "Design Desktop Layout" or the "Manage Container and Channels." In this example the channel is called myLoginJSP channel.
- 4) Change the following properties on the channel:
  - a. contentJSP to loginJSP.jsp
  - b. title to Login
  - c. save the changes

Like this:


## myLoginJSP

### Edit Properties

To change client type and locale, click the Table Preferences button  below

Client Type: default  
Locale: default


Property Path: myLoginJSP

Properties (15)				
Remove Customization Save Reset 				
<input type="checkbox"/>	Name	Value	Category	State
<input type="checkbox"/>	showExceptions	<input type="radio"/> True <input checked="" type="radio"/> False	Basic	Default
<input type="checkbox"/>	contentPage	<input type="text" value="loginJSP.jsp"/>	Basic	Default
<input type="checkbox"/>	editType	<input type="text" value="edit_subset"/>	Advanced	Default
<input type="checkbox"/>	width	<input type="text" value="thick"/>	Basic	Default
<input type="checkbox"/>	editPage	<input type="text" value="sampleedit.jsp"/>	Basic	Default
<input type="checkbox"/>	title	<input type="text" value="Login"/>	Basic	Default
<input type="checkbox"/>	productName	<input type="text" value="Sun Java™ System Portal Server 7"/>	Basic	Default
<input type="checkbox"/>	refreshTime	<input type="text" value="0"/>	Advanced	Default
<input type="checkbox"/>	helpURL	<input type="text" value="en/desktop/jspchann.htm"/>	Advanced	Default
<input type="checkbox"/>	<b>customProperties</b>	0 Values [Edit Values...]	Basic	Default
<input type="checkbox"/>	isTopLevel	<input type="radio"/> True <input checked="" type="radio"/> False	Advanced	Default
<input type="checkbox"/>	isEditable	<input type="radio"/> True <input checked="" type="radio"/> False	Advanced	Default
<input type="checkbox"/>	description	<input type="text" value="Login JSP Channel"/>	Basic	Default
<input type="checkbox"/>	processPage	<input type="text" value="sampledoedit.jsp"/>	Basic	Default
<input type="checkbox"/>	fontFace1	<input type="text" value="Sans-serif"/>	Basic	Default

- 5) Click on the customProperties hyper link high lighted in Green above. On the screen that is displayed click the **New Property** button


## myLoginJSP

### Edit Properties

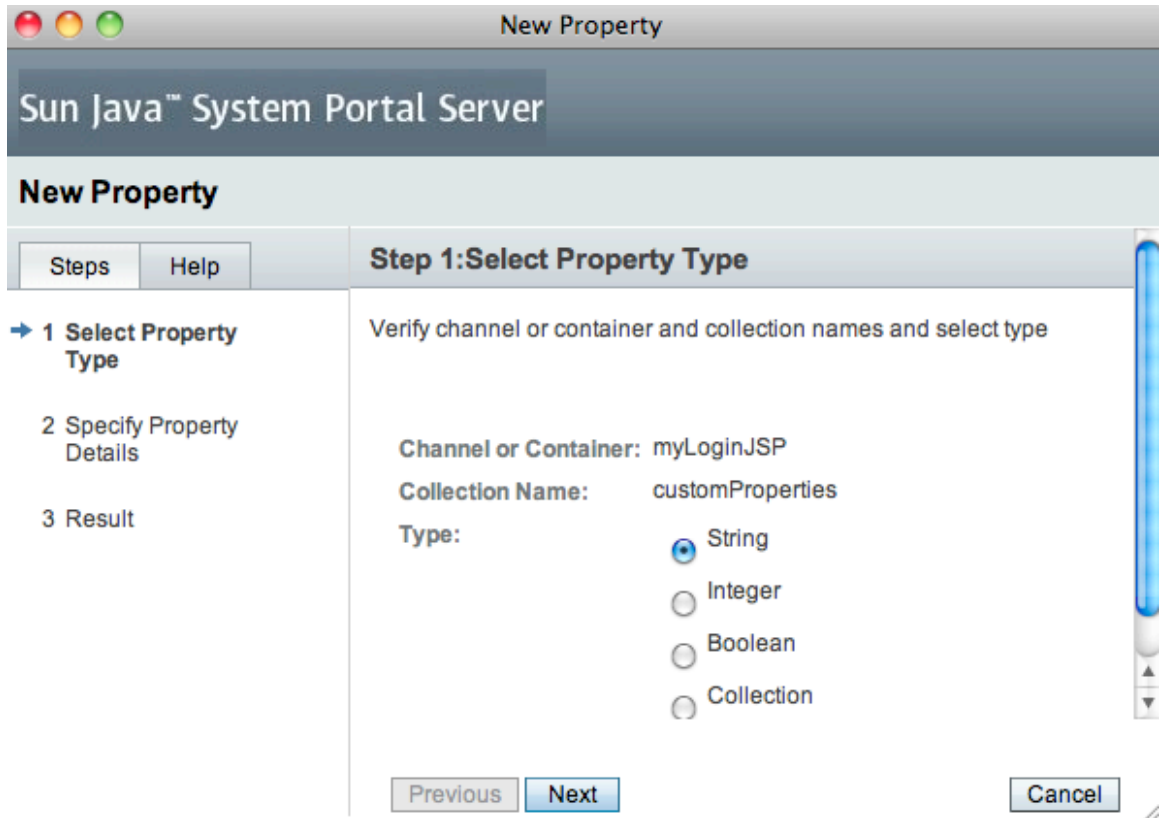
To change client type and locale, click the Table Preferences button  below

Client Type: default  
Locale: default

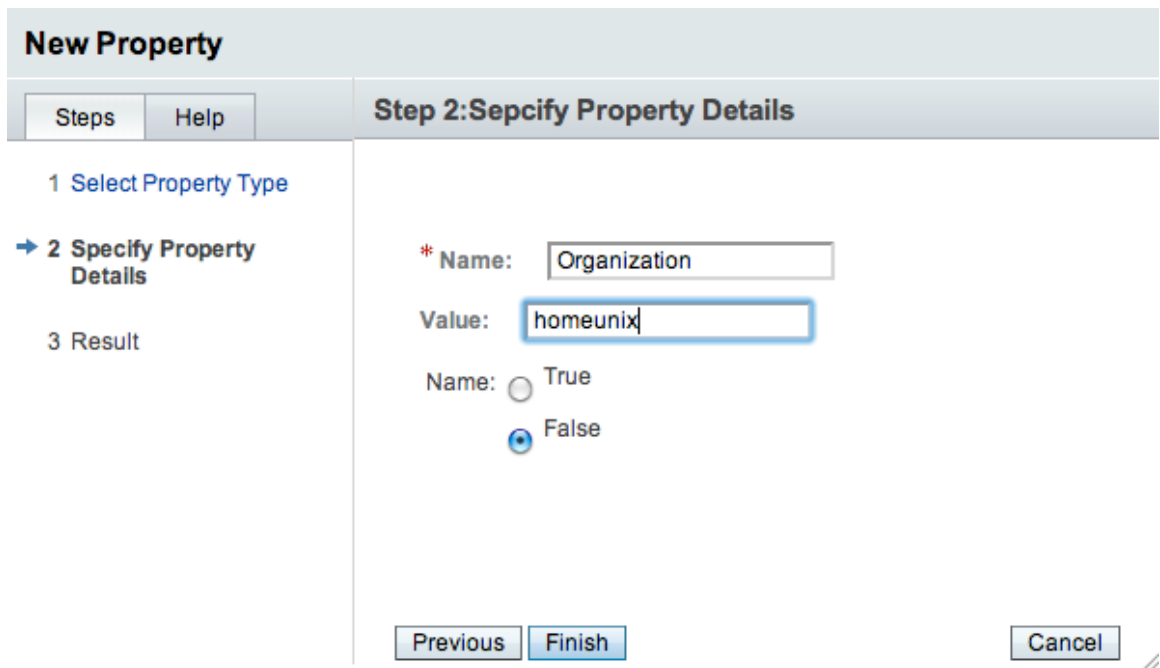
Property Path: myLoginJSP > customProperties

Properties (0)			
New Property... Remove Customization Save Reset 			
<input type="checkbox"/>	Name	Value	State
No items found.			

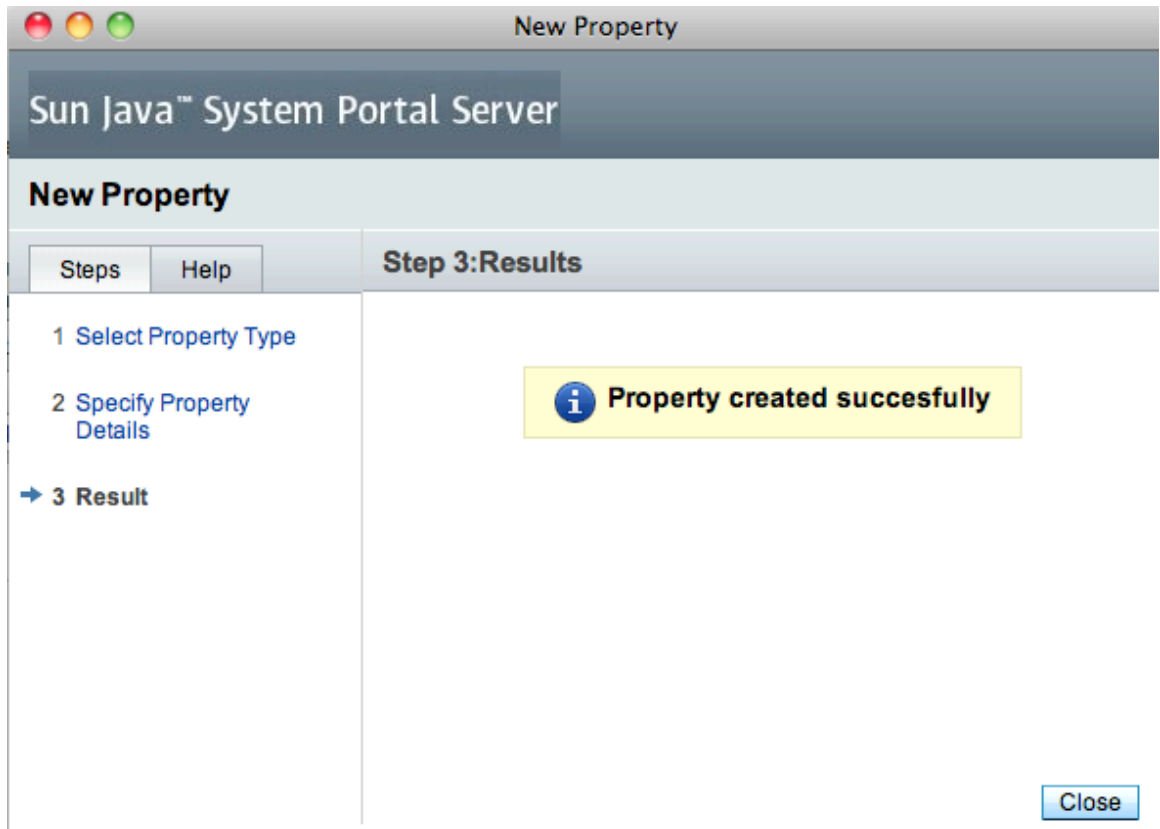
- 6) Create a New String property and click Next:



- 7) Set the Name to "Organization" and the Value to the Organization or Realm that belongs to the anonymous user. Then click the Finish button.




- 8) Click the Close button.



- 9) Repeat the process above and add a second custom property called "CookieDomain". Set this to the cookie domain for your portal.

### myLoginJSP

#### Edit Properties

To change client type and locale, click the Table Preferences button  below

Client Type: default  
Locale: default

Property Path: [myLoginJSP](#) > customProperties

Properties (2)				
Name	Value	Category	State	
<input type="checkbox"/> Organization	<input type="text" value="homeunix"/>	Basic	Customized	
<input type="checkbox"/> CookieDomain	<input type="text" value=".homeunix.net"/>	Basic	Customized	

Close

- 10) Close and Save your data.

- 11) Remove the existing loginProvider from the anonymous desktop.

## 3.2 Create the Portal Desktop Folder

Now that the Channel is configured the folder on the portal server hosts must be created. This must be done on all portal server hosts. Do the following:

- 1) Log on to the portal server host with an account with sufficient privileges to add the portal folder.
- 2) `cd /var/opt/SUNWportal/portals/{portalInstance}/desktop/{portalType}`
- 3) `cp -r ../default/JSPPProvider myLoginJSP`
- 4) `cd myLoginJSP`
- 5) place the loginJSP.jsp file found in **Appendix – loginJSP.jsp** in this folder. The file can be downloaded from <http://jeffnester.com/howtos/portal/v7/loginJSP.jsp.txt>

## 4 Testing

Go to <http://portal.sun.com/portal/dt>

If the channel is configured properly the portal show display the login channel.

## 5 Appendix – loginJSP.JSP

```
<%@page import="java.io.*" %>

<%@page import="java.util.*" %>
<%@page import="javax.servlet.http.*" %>

<%@page import="java.security.Principal" %>
<%@page import="javax.security.auth.*" %>
<%@page import="javax.security.auth.login.*" %>
<%@page import="java.net.URLEncoder" %>
<%@page import="javax.security.auth.callback.*" %>

<%@page import="com.sun.identity.authentication.AuthContext" %>
<%@page import="com.sun.identity.authentication.spi.AuthLoginException" %>
<%@page import="com.iplanet.sso.*" %>
<%@ taglib uri="/tld/jx.tld" prefix="jx" %>
<%@ taglib uri="/tld/desktop.tld" prefix="dt" %>
<%@ taglib uri="/tld/desktopProviderContext.tld" prefix="dtpc" %>
<%@ taglib uri="/tld/i18n.tld" prefix="i18n" %>
<!--
Author: Jeff Nester
Email: jeff.nester@sun.com
```

### Description:

The login provider on the Sun Portal Server 7.x anonymous desktop by default does not handle failure events. If an event like the user enters a bad password or the user's password has expired they are redirected to the Access Manager page that deals with the



error. This JSP provides a replacement for the default login mechanism for the anonymous desktop. This JSP handles all of the know errors inside the Channel.

Events Handled:

- 1) Authentication Failure due to a bad password or username
- 2) Password Retry Limit exeeded
- 3) Password Reset or Password Expired

```
--!>
<dt:obtainChannel channel="$JSPProvider">
  <dtpc:providerContext>
    <i18n:setBundle baseName="SampleJSP" var="samplejsp" />
    <!--
      The Organization is needed for the authentication to work. For now
      this is stored in the Description property for the channel.
    -->

    <dtpc:getCollectionProperty key="customProperties" id="cProp"/>
    <jx:declare id="cProp" type="java.util.HashMap" />

    <%
//
// The orgName must be set to the correct realm for your environment.
//

    int processed = 0;
    boolean looping = false;
    boolean processLogin = false;
    AuthContext lc = null;
    AuthContext.IndexType indexType;
    String indexName = null;
    String doWhat = request.getParameter("processLogin");
    String thisUser = null;
    String orgName = null;
    String sstokencookiedomain = null;

//
// get the orginazation and cookie domain from the customProperites on the
// desktop channel. There should be two properties defined:
//   Organization - i.e. homeunix
//   CookieDomain - i.e. .homeunix.net
//
    if ( cProp.size() != 0 ) {
      orgName = (String) cProp.get("Organization");
      sstokencookiedomain = (String) cProp.get("CookieDomain");
    }
    if ( ( orgName == null ) || ( sstokencookiedomain == null ) ) {
      doWhat = "customPropMissing";
    }

    if (doWhat == null) {
//
// First time through so display login page
//
      %>
```

```

<center>
  <form action="/portal/dt" method="post">
    <table>
      <tr>
        <td>
          Username:
        </td>
        <td>
          <input type="text" name="user"/>
        </td>
      </tr>
      <tr>
        <td>
          Password:
        </td>
        <td>
          <input type="password" name="passwd"/>
        </td>
      </tr>
      <tr>
        <td colspan="2" align="center">
          <input type="submit" value="Login"/>
        </td>
      </tr>
    </table>

    <input type="hidden" name="processLogin" value="login"/>
  </form>
</center>

<%           } else if (doWhat.equals("customPropMissing")) {
%>
<center>
  <div style="border:2px solid red;width:200px;height:40px;text-align:center;"><br>Custom Properties Not Defined!<br>
</div>
<br>
  Contact the System Administration<br><br>
  <a href="/portal/dt">Go back to Login</a>

</center>
<%           } else if (doWhat.equals("unknownStatus")) {
%>
<center>
  <div style="border:2px solid red;width:200px;height:40px;text-align:center;"><br>Unknown Error<br>
</div>
<br>
  Contact the System Administrator<br><br>
  <a href="/portal/dt">Go back to Login</a>

</center>
<%           } else if (doWhat.equals("retryExceeded")) {

```

```

%>
<center>
  <div style="border:2px solid red;width:200px;height:40px;text-
align:center;"><br>Exceeded Password Retry Limit<br>
  </div>
  <br>
  <a href="/portal/dt">Go back to Login</a>

</center>
<%           } else if (doWhat.equals("unknownStatus")) {
%>
<center>
  <div style="border:2px solid red;width:200px;height:40px;text-
align:center;"><br>Unknown Error<br>
  </div>
  <br>
  Contact the System Administrator<br><br>
  <a href="/portal/dt">Go back to Login</a>

</center>
<% } else if (doWhat.equals("authenticationFailed")) {
%>
<center>
  <form action="/portal/dt" method="post">
    <table>
      <tr>
        <td colspan="2">
          <div style="border:2px solid
red;width:200px;height:40px;text-align:center;"><br>
            <b><font color="red">Authentication
Failed</font></b><br>
          </div>
          <br>
        </td>
      </tr>
      <tr>
        <td>
          Username:
        </td>
        <td>
          <input type="text" name="user"/>
        </td>
      </tr>
      <tr>
        <td>
          Password:
        </td>
        <td>
          <input type="password" name="passwd"/>
        </td>
      </tr>
      <tr>
        <td colspan="2" align="center">
          <input type="submit" value="Login"/>
        </td>
      </tr>
    </table>
  </form>

```

```

        </td>
    </tr>
    <tr>
        <td colspan="2" align="center">
            <a href="/portal/dt">Go back to Login</a>
        </td>
    </tr>
</table>

    <input type="hidden" name="processLogin" value="login"/>
</form>
</center>
<% } else if (doWhat.equals("login")) {

//
// Process Login
//
        processed = 0;
        looping = false;
        lc = null;
        indexName = null;

//
// orgName is hard coded for now. Not sure how to get the realm that currently
connected to
//

        try {
            lc = new AuthContext(orgName);
            indexType = AuthContext.IndexType.MODULE_INSTANCE;
            indexName = new String("LDAP");
            lc.login(indexType, indexName);

        } catch (AuthLoginException le) {
            System.out.println("loginJSP --> AuthLoginException " +
le.getMessage());
            le.printStackTrace();
        }
    <%>
    <center>
        <div style="border:2px solid red;width:200px;height:60px;text-align:center;"><br>
            <font color="red">A Serious Error<br>has occurred</font>
        </div>
        <br>
        Contact the System Administrator<br><br>
        <a href="/portal/dt">Go back to Login</a>
    </center>
    <%
        break;
    }
//
// Process Callbacks
//
        Callback[] callbacks = null;

        // get information requested from module

```

```

while (lc.hasMoreRequirements()) {
    if (processed > 1) {
        System.out.println("loginJSP --> STOP!! You have been
here before");

        looping = true;

        break;
    }

    callbacks = lc.getRequirements();

    if (callbacks != null) {

        try {
            int i = 0;
            try {
                for (i = 0; i < callbacks.length; i++) {

                    if (callbacks[i] instanceof NameCallback) {

                        NameCallback nc = (NameCallback)
callbacks[i];
                        // I am making sure that this is the
correct prompt
                        // the xml has spaces before and after
                        if (nc.getPrompt().equals(" User Name:
")) {
                            if (request.getParameter("user") !=
null) {
                                nc.setName(request.getParameter("user"));
                                thisUser =
request.getParameter("user");

                                }
                            }
                        } else if (callbacks[i] instanceof
PasswordCallback) {
                            PasswordCallback pc =
(PasswordCallback) callbacks[i];
                            String input = null;
                            if (request.getParameter("passwd") !=
null) {
                                input =
request.getParameter("passwd");

                                }
                                char[] passwd = input.toCharArray();
                                pc.setPassword(passwd);
                            }
                        }
                    } catch (Exception e) {

```

```

        System.out.println("loginJSP --> Callback
Exception: " + e.getMessage());
        throw new
UnsupportedCallbackException(callbacks[i], "Callback exception: " + e);
    }
    lc.submitRequirements(callbacks);
    processed = processed + 1;
    } catch (Exception e) {
        System.out.println("loginJSP --> Missing Call
Backs");
    }
    <%>
    <center>
        <div style="border:2px solid red;width:200px;height:60px;text-
align:center;"><br>
            <font color="red">A Serious Error<br>has occurred</font>
        </div>
        <br>
        Contact the System Administrator<br><br>
        <a href="/portal/dt">Go back to Login</a>
    </center>
    <%
        return;
    }
    }
    if (!looping) {
        processLogin = true;
    } else {
        //
        // Password reset is the only valid reason to come through
again
        //
    }
    <%>
    <center>
        <form action="/portal/dt" method="post">
            <table>
                <tr>
                    <td colspan="2">
                        <div style="border:2px solid
red;width:200px;height:40px;text-align:center;"><br>
                            <b><font color="red">Password Expired</font></b><br>
                        </div>
                    <br>
                    </td>
                </tr>
                <tr>
                    <td>
                        Old Password:
                    </td>
                    <td>
                        <input type="password" name="passwd"/>
                    </td>
                </tr>
                <tr>

```

```

        <td>
            New Password:
        </td>
        <td>
            <input type="password" name="newPasswd" />
        </td>
    </tr>
    <tr>
        <td>
            Confirm Password:
        </td>
        <td>
            <input type="password" name="confirmPasswd" />
        </td>
    </tr>
    <tr>
        <td colspan="2" align="center">
            <input type="submit" value="Change Password" />
        </td>
    </tr>
    <tr>
        <td colspan="2" align="center">
            <a href="/portal/dt">Go back to Login</a>
        </td>
    </tr>
</table>

<input type="hidden" name="processLogin" value="changePassword" />
<input type="hidden" name="user" value="<%=thisUser%>" />

</form>
</center>

<%
    }
    } else if (doWhat.equals("changePassword")) {

//
// Process Login
//
        processed = 0;
        looping = false;
        lc = null;
        indexName = null;

//
// orgName is hard coded for now. Not sure how to get the realm that currently
connected to
//

        try {
            lc = new AuthContext(orgName);
            indexType = AuthContext.IndexType.MODULE_INSTANCE;
            indexName = new String("LDAP");
            lc.login(indexType, indexName);

```

```

        } catch (AuthLoginException le) {
            System.out.println("loginJSP --> AuthLoginException " +
le.getMessage());
            le.printStackTrace();
        }
    }
}

<center>
    <div style="border:2px solid red;width:200px;height:60px;text-align:center;"><br>
        <font color="red">A Serious Error<br>has occurred</font>
    </div>
    <br>
    Contact the System Administrator<br><br>
    <a href="/portal/dt">Go back to Login</a>
</center>
    <%
        break;
    }
//
// Process CallBacks
//
    Callback[] callbacks = null;

    // get information requested from module
    while (lc.hasMoreRequirements()) {
        if (processed > 3) {
            System.out.println("loginJSP --> STOP!! You have been
here before");

            looping = true;
            break;
        }

        callbacks = lc.getRequirements();

        if (callbacks != null) {

            try {
                int i = 0;
                try {
                    for (i = 0; i < callbacks.length; i++) {

                        if (callbacks[i] instanceof NameCallback) {

                            NameCallback nc = (NameCallback)
callbacks[i];
                            // I am making sure that this is the
correct prompt
                            // the xml has spaces before and after
                            if (nc.getPrompt().equals(" User Name:
")) {
                                if (request.getParameter("user") !=
null) {
                                    nc.setName(request.getParameter("user"));
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
}

```



```

request.getParameter("user");
    thisUser =
    }
} else if (callbacks[i] instanceof
PasswordCallback) {
    PasswordCallback pc =
    String thisPasswd = null;
    if (pc.getPrompt().contains("Old")) {
        if (request.getParameter("passwd")
!= null) {
            thisPasswd =
            }
        } else if
            if
                thisPasswd =
            }
        } else if
            if
                thisPasswd =
            }
        } else {
            if (request.getParameter("passwd")
!= null) {
                thisPasswd =
            }
        }
    }
    char[] passwd =
    pc.setPassword(passwd);
}
} catch (Exception e) {
    System.out.println("loginJSP - Callback
Exception: " + e.getMessage());
    throw new
UnsupportedCallbackException(callbacks[i], "Callback exception: " + e);
}
lc.submitRequirements(callbacks);
processed = processed + 1;

```



```

        <input type="password" name="newPasswd"/>
    </td>
</tr>
<tr>
    <td>
        Confirm Password:
    </td>
    <td>
        <input type="password" name="confirmPasswd"/>
    </td>
</tr>
<tr>
    <td colspan="2" align="center">
        <input type="submit" value="Change Password"/>
    </td>
</tr>
<tr>
    <td colspan="2" align="center">
        <a href="/portal/dt">Go back to Login</a>
    </td>
</tr>
</table>

```

```

<input type="hidden" name="processLogin" value="changePassword"/>
<input type="hidden" name="user" value="<%=thisUser%>"/>

```

```

</form>
</center>
<%

```

```

    }
}
if (processLogin) {

    if (lc.getStatus() == AuthContext.Status.SUCCESS) {
        SSOToken token = lc.getSSOToken();
        SSOTokenManager manager = SSOTokenManager.getInstance();
        manager.refreshSession(token);
        SSOTokenID tokenID = token.getTokenID();

        String redirectURL = new String("/portal/dt");

        String strToken = token.getTokenID().toString();
        String sstokencookieName = "iPlanetDirectoryPro";
        //
        //domain is hard coded for now
        //
        String sstokencookiePath = "/";
        Cookie sso = new Cookie(sstokencookieName,
URLLEncoder.encode(strToken, "UTF-8"));
        sso.setDomain(sstokencookiedomain);
        sso.setPath(sstokencookiePath);
        response.addCookie(sso);
        response.sendRedirect(redirectURL);

    } else if (lc.getStatus() == AuthContext.Status.FAILED) {

```

```

//
// 103 - Invalid Password
// 107 - is Authorization failed - could be because no such
user id
// 123 - Retry count exceeded
//
if ((lc.getErrorCode().equals("107") ||
lc.getErrorCode().equals("103"))) {
    %>

    <form name="authenticationFailed" method="POST" action="/portal/dt">
        <input type="text" name="test"/><br>
        <input type="hidden" name="processLogin" value="authenticationFailed"/>
    </form>
    <script type="text/javascript" language="JavaScript">
        document.authenticationFailed.submit();
    </script>
    <%
        } else if (lc.getErrorCode().equals("123")) {
    %>

    <form name="retryExceeded" method="POST" action="/portal/dt">
        <input type="text" name="test"/><br>
        <input type="hidden" name="processLogin" value="retryExceeded"/>
    </form>
    <script type="text/javascript" language="JavaScript">
        document.retryExceeded.submit();
    </script>
    <%
        }
    } else {
        System.out.println("loginJSP - Unknown Status");
    %>

    <form name="unknownStatus" method="POST" action="/portal/dt">
        <input type="text" name="test"/><br>
        <input type="hidden" name="processLogin" value="unknownStatus"/>
    </form>
    <script type="text/javascript" language="JavaScript">
        document.unknownStatus.submit();
    </script>
    <%
        }
    }
    %>
</dtpc:providerContext>
</dt:obtainChannel>

```