

# How to Enable Single Sign-on with Non-SAP Web Application



## Applies To:

SAP NetWeaver 2004s (EP7)

## Summary

This article describes how to establish Single Sign-on (SSO) between SAP Enterprise Portal and another non-SAP application. Portal can act as a gateway to access different application smoothly without logging in to the application every time. Seamless access can be provided in several ways – i) using user mapping ii) using single sign-on. If user mapping is implemented, each portal user has to provide corresponding user name/password of the specific application in some location in the portal (one-time activity). But if single sign-on is implemented, user does not have to bother about authentication. He/she has to log on to the portal only once. Then he/she will have seamless access to all the other application without being challenged for authentication details. Single sign-on can be easily implemented in SAP Enterprise Portal with other SAP applications like BW, R/3 etc, but this implementation is not straight forward for custom built application (developed in JSP or ASP.NET). This article provides step by step guide to solve the problem.

**Author:** Sudip Majumder

**Company:** TCS – Tata Consultancy Services

**Created on:** 19 March 2008

## Author Bio



Sudip Majumder is SAP certified - Development Consultant (SAP NetWeaver 2004 - Portal & KMC). He is working in TCS since 2004.

## Table of Contents

Problem Description .....	3
Solution Overview .....	3
Prerequisite.....	4
Solution Details .....	4
Creating a Portal Component .....	4
Sending SAP Logon Ticket to the Non-SAP Application .....	5
Deploy and Create iView.....	5
Decrypt the SAP Logon Ticket in Non-SAP Application .....	6
Personalizing the Application to Accept User Inputs .....	6
Conclusion .....	8
Related Content.....	8
Disclaimer and Liability Notice.....	9

## Problem Description

This article describes how to establish Single Sign-on (SSO) between SAP Enterprise Portal and another non-SAP application. Portal can act as a gateway to access different application smoothly without logging in to the application every time. Seamless access can be provided in several ways – i) using user mapping ii) using single sign-on. If user mapping is implemented, each portal user has to provide corresponding user name/password of the specific application in some location in the portal (one-time activity). But if single sign-on is implemented, user does not have to bother about authentication. He/she has to log on to the portal only once. Then he/she will have seamless access to all the other application without being challenged for authentication details. Single sign-on can be easily implemented in SAP Enterprise Portal with other SAP applications like BW, R/3 etc, but this implementation is not straight forward for custom built application (developed in JSP or ASP.NET). This article provides step by step guide to solve the problem.

## Solution Overview

Single Sign-on process flow is mentioned in the diagram below –

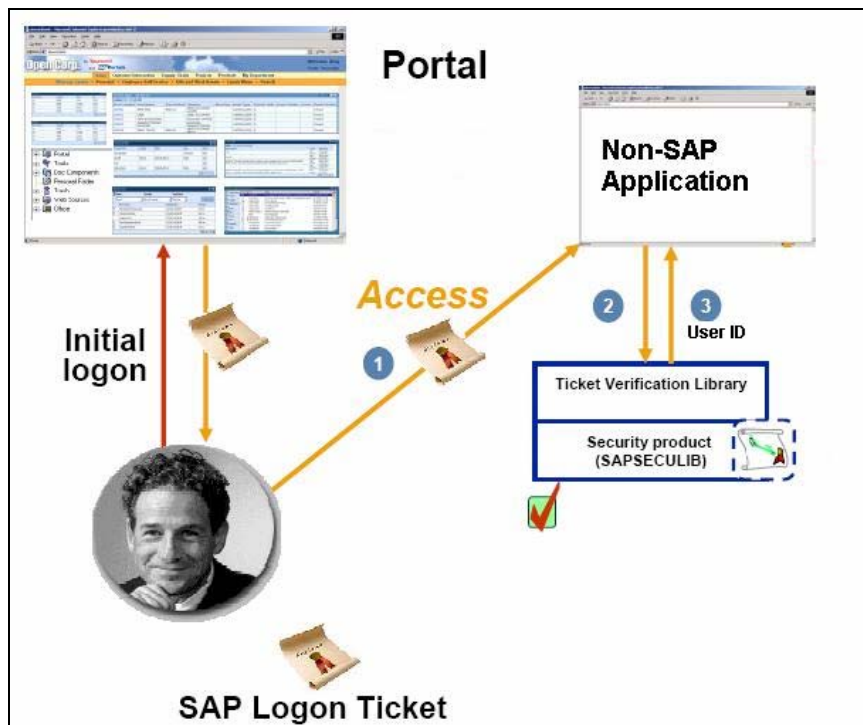


Figure 1 : Single Sign-on Process Flow

- User logs on to the portal.
- SAP Logon Ticket ("MYSAPSSO2" cookie) is sent to the user browser by the portal. This cookie is stored in the local PC of the user.
- This logon ticket contains the user ID in encrypted form.
- An application will be created in portal that will call the destination non-SAP application after attaching the logon ticket with the request.
- The destination non-SAP application will catch the cookie and decrypt it to obtain the user ID.
- To decrypt the cookie the destination application will need different dynamic link libraries (e.g. *sapsecu.dll* and *sapssoext.dll*) and the public key from portal.
- After obtaining the user ID, the non-SAP application screen will be shown for that user corresponding to the permission the user has in that application.

## Prerequisite

- SAP Enterprise Portal Server and the Destination Non-SAP application must lie in the same domain (e.g. if Portal URL is “*abc.company.com*” then the destination system URL should be like “*def.company.com*”).
- User IDs in the Non-SAP application must be same as that of portal (e.g. if there is a user in portal with the ID “abcd”, then the Non-SAP application must have a user having ID “abcd”)

## Solution Details

Please follow the below mentioned steps to accomplish the task.

### Creating a Portal Component

- Open NetWeaver Developer Studio (NWDS)
- Click New → Project → Portal Application. Provide a meaningful name for the project (e.g. *NonSAPApplicationIntegration*)
- Add a component to the project. Click New → Other → Portal Application → Create a new Portal Application Object → Select *NonSAPApplicationIntegration* → Portal Component → Abstract Portal Component.

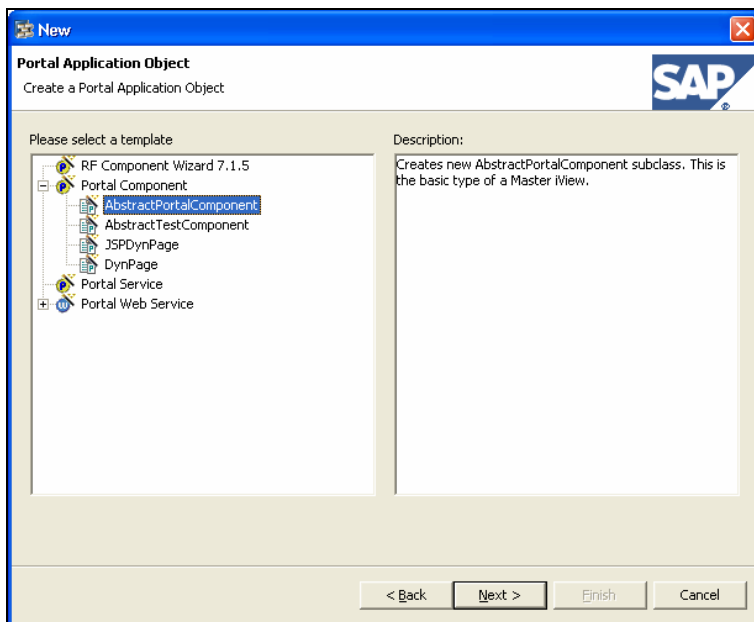


Figure 2 : Abstract Portal Component

- In the next screen enter the values as –
  - Name: *NonSAPApplicationIntegration*
  - Package Name: <Some meaningful name, e.g. *com.company.application.sso*>
- Click Finish
- Import the following “JAR” file to the project library –
  - servlet.jar*

## Sending SAP Logon Ticket to the Non-SAP Application

We will put our code inside the doContent() method in the portal component. Follow the steps as mentioned below –

- Obtain HTTP Request and HTTP Response.

```
HttpServletRequest req = request.getServletRequest();
HttpServletResponse resp = request.getServletResponse(true);
```

- Obtain all available cookies from the browser. From the available cookies, find out the SAP Logon Ticket (with cookie name "MYSAPSS02").

- ```
Cookie currentCookie = null;
Cookie cookiesArray[] = req.getCookies();
    int cookieArrayLength = cookiesArray.length;

    for(int i=0;i<cookieArrayLength;i++)
    {
        currentCookie = cookiesArray[i];

        currentCookieName = currentCookie.getName();

        //Get out of the loop when "MYSAPSS02" cookie is obtained
        if(currentCookieName.equalsIgnoreCase("MYSAPSS02"))
        {
            boolReceivedCookie = true;
            break;
        }
    }
```

- Add the SAP Logon Ticket in the HTTP Response.

```
resp.addCookie(currentCookie);
```

- Redirect the response to call the Non-SAP application with the added Logon Ticket.

```
String nonSAPApplicationURL = "<Application URL>";
// e.g. "http://MyNonSAPApplication/Integrated/Login.aspx"

resp.sendRedirect(nonSAPApplicationURL);
```

## Deploy and Create iView

To use this application follow the steps –

- Export the application to PAR file and deploy it to the portal.
- Create an iView with that PAR file

When you will preview the iView, the application will be executed and the non-SAP application will be called with the SAP Logon Ticket. The destination application has to catch the ticket and decrypt it to obtain the user ID from it.

## Decrypt the SAP Logon Ticket in Non-SAP Application

- Register the dynamic link libraries *sapsecu.dll* and *sapssoext.dll* in the server hosting the application.
- These libraries should be registered and kept in the path variable location (e.g. C:\WINDOWS\system32).
- Download the public key of the logon ticket (“*verify.pse*”) from the portal. Login to portal with system administrator privilege. Navigate to *System Administration* → *System Configuration* → *Keystore Administration* and download the “*verify.pse*” file. Keep this file inside the server hosting the non-SAP application. The path to the file will be referred in future.
- Modify the login page of your application to receive the cookie from the request object. Write code to decrypt the cookie in order to obtain the logon ID of the user who requested to open the application. Ready-made code is available in SDN to decrypt the cookie in Java and ASP.NET application. You may refer to the following link - <https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/ec82ec90-0201-0010-72bc-88ef150211ff>.
- After getting the logon ID of this user, retrieve data from the application database for authorization and open the application home page accordingly.

## Personalizing the Application to Accept User Inputs

To make this application robust, we need to provide a way such that users can customize the Non-SAP Application URL without modifying the code -

Follow the below mentioned steps to accomplish this task –

- Open NWDS and go to the project you have just created.
- In the project structure tree (left side of the screen), navigate to *dist* → *PORTAL-INF* → *portalapp.xml*.
- Click on to the Components Tab.
- Right-click component-profile and click Add component-profile property
- Click New and create property as mentioned below and click Finish.
  - Name : nonSAPApplicationURL
  - Value :”http://MyNonSAPApplication/Integrated/Login.aspx”
- Right-click on *nonSAPApplicationURL* and click Add sub-property.
- Click “Add Standard” and populate values as mentioned below -
  - *Personalization* : *dialog*
- The component-profile properties will look like this –

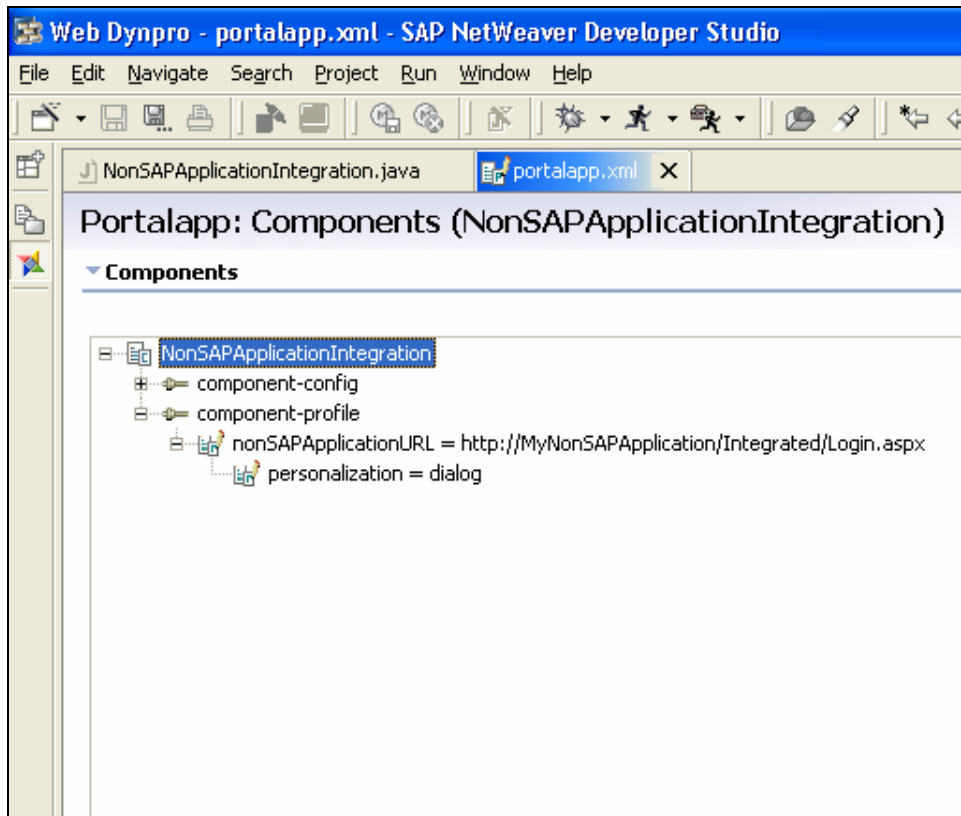


Figure 3 : Component-Profile Property

- Deploy and export the application in the portal.
- Open the iView object just created with the PAR file.
- You can see an editable options in the iView – *nonSAPApplicationURL* as shown below –

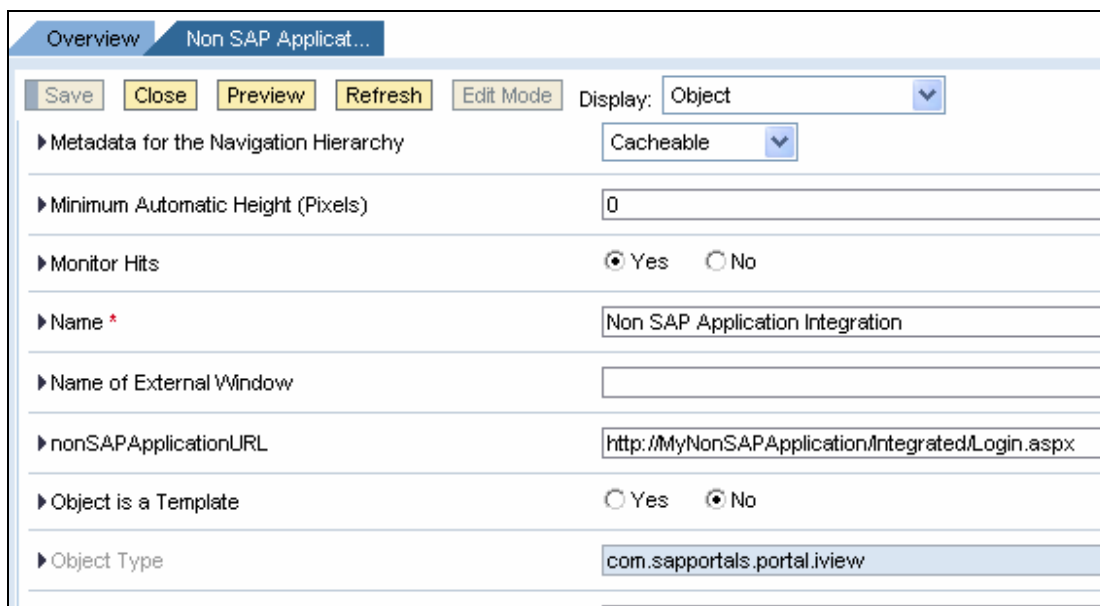


Figure 4 : Customizing the Application

- Modify the code such that the application gets the required information from the personalization setting. Update your code as follows –

```
IPortalComponentContext componentContext = request.getComponentContext();
    IPortalComponentProfile profile =
componentContext.getProfile();

    String appURL = profile.getProperty("nonSAPApplicationURL");
```

- *appURL* variable will now contain the URL of the application to be opened after integration. Based on this, you can apply the appropriate code as mentioned above.
- Now you do not have to open your code again and again if the application URL changes after some time. Just modify the value in the iView as shown above and the application will accommodate appropriate changes automatically.

## Conclusion

Now users will be able to access other non-SAP web application without multiple authentications. They have to provide their authentication details only once when he/she logs on to the portal (or when he/she logs on to the computer, if portal is integrated with windows authentication). If URL of the application changes then the application does not require any modification. These values can be simply changed in the iView properties and the application acts as per the user specified inputs. But this needs to be remembered that the above mentioned procedure is valid if portal and the destination application are in the same domain. Single sign-on is also possible with applications in different domains. But separate procedure has to be followed to do that.

## Related Content

[sdn.sap.com](http://sdn.sap.com)

[help.sap.com](http://help.sap.com)



## **Disclaimer and Liability Notice**

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.