

Access External Database and Display through HTMLB Charts at SAP NetWeaver Portal



Applies to:

SAP EP 7.0 SP 11, SAP NWDS 2004s

Microsoft Windows XP Professional, Version 2002, Service Pack 2

For more information, visit the [Portal and Collaboration homepage](#).

Summary

This application helps us to deal with DSN-less connectivity at SAP NetWeaver Portal for accessing data from an External Database, MS Excel Work Book present at the server. This summarizes the implementation of HSSF architecture, HTMLB objects and Vector Chart Model. The cause and resolution for the issues faced was also summarized.

Author: Kasha Shaila

Company: Satyam Computers Services Limited

Created on: 22 December 2008

Author Bio



Shaila Kasha working as an SAP Net weaver Portal Consultant for Satyam Computers-Hyderabad since 2006.

Table of Contents

Problem Description	3
Solution Details	3
Process Flow.....	4
JDBC Architecture	4
Solution Details.....	5
Prerequisites	5
Procedure.....	5
Steps.....	5
Tricks and Tips	8
MS Excel Access	8
Using POI-HSSF API	8
Implementation of HTMLB	8
Intro.....	8
Vector Chart Model:	8
Troubleshoot.....	9
Issues at using DSN-less connections.....	9
Summarizing Wiki Contents:.....	9
Scenario 1:.....	9
Scenario 2:.....	9
Important Note	12
Obtain Result Set.....	12
Security Permissions.....	12
Related Content.....	13
Disclaimer and Liability Notice.....	14

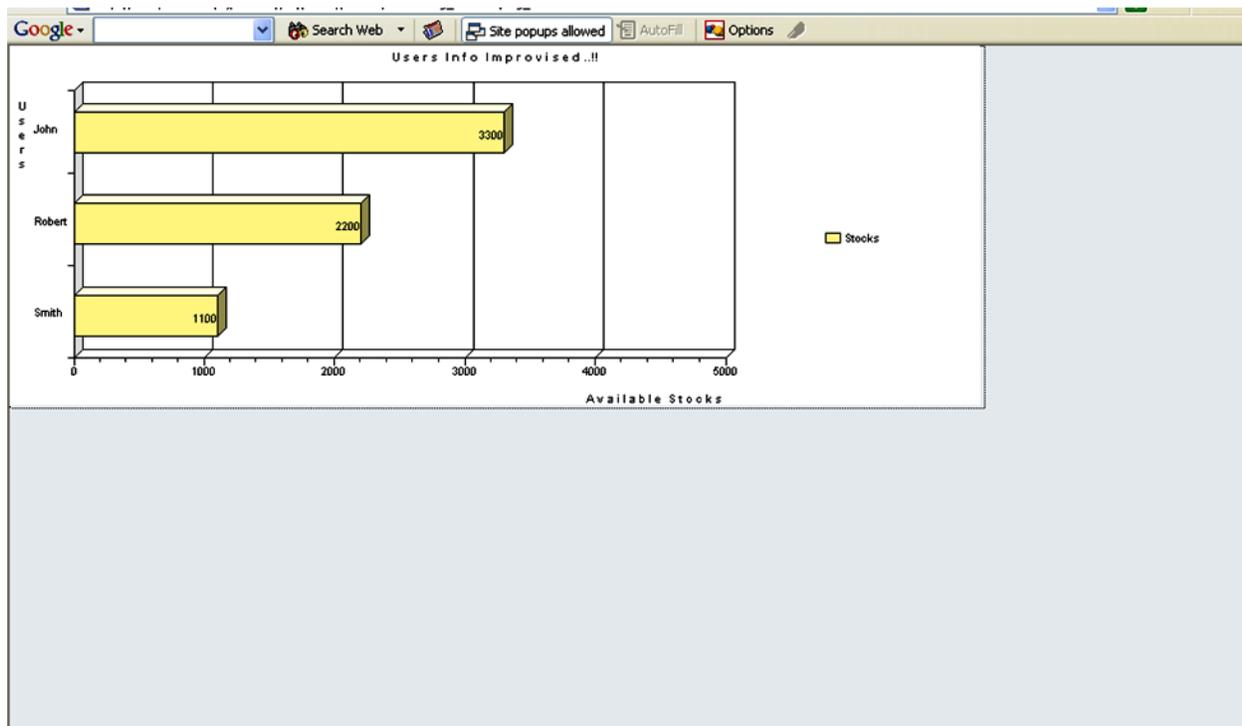
Problem Description

The situation arises when we need to store huge amounts of data for Portal use. It becomes unmanageable to store these huge amounts of data. In our present scenario we use Database with manageable size. Hence we are storing this at a Server folder. Now we are facing with the issue of accessing this data and trying to modify the same using HTMLB Charts. As I was implementing the methods for accessing an External Database from SAP Enterprise Portal, I was held with many unforeseen issue for establishing the DSN-less connectivity. So I thought it would be a good option to blog these down so that it might be useful for all of us who are trying to implement such scenarios and use of Chart Models at SAP Portal.

Solution Details

As we know that portal shares the same database and schema installed for the SAP Web Application Server (SAP WebAS). In this schema, it stores unprocessed information about objects and properties, including meta-data relating to lists, profiles of property files and their attributes, and other data. Interfacing with the database is a set of portal services in the PCD that access the stored information and other data needed by the portal at runtime. But for our requirement we need to access an External Database so I have used JDBC-ODBC drivers for accessing files at the Server and then obtain the data for modifying using SQL statements. Finally HTMLB Chart Model for displaying the data in a more readable format.

Screen 1: This is the output required which shows the sample info(Simplest Example) about users present at our External database.



Note: Security permissions at the folder are most vital pre-requisites.

Process Flow

JDBC Architecture

The Java Database Connectivity (JDBC) API is the industry standard for database-independent connectivity between the programming language and a wide range of databases – SQL databases and other tabular data sources, such as spreadsheets or flat files. The JDBC API provides a call-level API for SQL-based database access.

JDBC technology allows you to use the programming language to exploit the capabilities for applications that require access to enterprise data. With a JDBC technology-enabled driver, you can connect all corporate data even in a heterogeneous environment

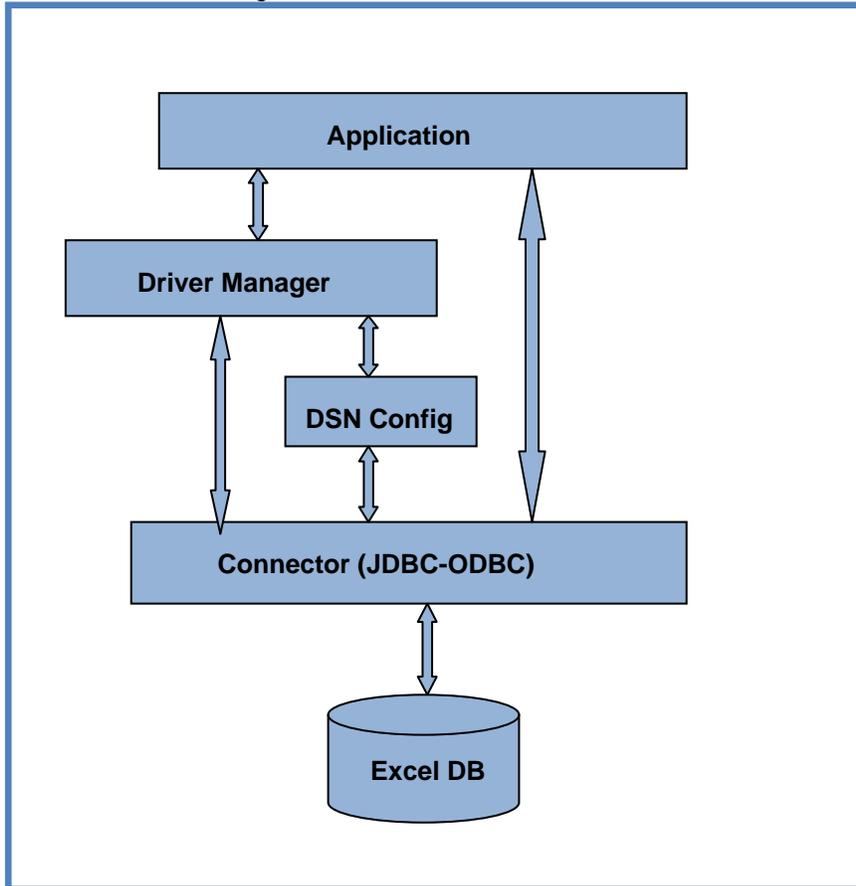


Figure 1: Connection Architecture

The implementation for achieving this was done at DYN page where we have the provision of accessing methods at ease. The methods are `doInitialization`, `doProcessAfterInput`, `doProcessBeforeOutput`. First and foremost is the establishment of connectivity for the MS Excel file.

We implement code for getting the driver and connection string. Once we get the apt connection we proceed with the accessing of data using SQL statement. Here we test for data existence and now we start with using of HTMLB charts. A chart displays data that are relevant for the user in a graphical representation so that the characteristics of the data and their relations are easy to capture for the user.

Solution Details

Prerequisites

- SAP NWDS and the component for accessing External Database are implemented at the same server
- “Everyone” user has ‘full control’ permissions for that folder at Server where our data was stored.

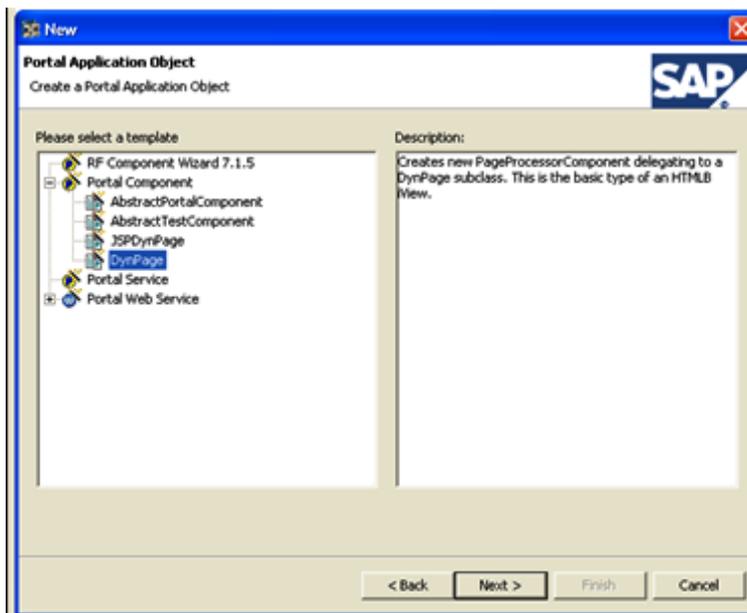
Procedure

Steps

Please follow these at SAP NWDS.

We are implementing this at the SAP NWDS 7.0. Please follow the below procedure in a stepwise manner for achieving our aim.

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



- In the next screen enter the values as –
 - Name: Charts
 - Package Name: <Some meaningful name, e.g. com.xxx.application.charts>

- Click Finish
- Include the below code in the in doProcessBeforeOutput() of the Charts.

```
String Connection_str=null;
String excel_path="D:/FOLDER/FILE.xls";
Statement SQL_stmt=null;
Connection excel_conn=null;
ResultSet excel_result=null;
IPortalComponentRequest request= (IPortalComponentRequest) this.getRequest ();
IPortalComponentResponse response= (IPortalComponentResponse) this.getResponse ();

try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
// TO GET THE DRIVER
excel_conn=DriverManager.getConnection("jdbc:odbc:Driver={Microsoft Excel Driver
(*.xls)};DBQ="+excel_path+";DriverID=790;READONLY=false","", "");
//TO GET EXCEL CONNECTION connection=
DriverManager.getConnection(conn_str,<id>,<pwd>);
SQL_stmt=excel_conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONC
UR_UPDATABLE);
//TO GET THE SQL STATEMENT TO ACCESS THE EXCEL
excel_result=SQL_stmt.executeQuery("Select * from [SHEET_NAME$]");
//SQL STATEMENT

Form myForm = this.getForm(); // get the form from DynPage
Chart mychart=new Chart();
VectorChartModel chartmodel= new VectorChartModel();

while (excel_result.next())
{
//THE EXCEL CONSISTS DATA coz result set has next elements in it

String Row1=excel_result.getString(1);
String Row2=excel_result.getString(2);
String Row3=excel_result.getString(3);

mychart.setVisible(true);
mychart.setDisplayObjectValues(true);
mychart.setTitleCategories("Users");
mychart.setTitleValues("A v a i l a b l e S t o c k s");
mychart.setTitle("U s e r s I n f o I m p r o v i s e d . . ! !");
mychart.setChartType(ChartType.BARS_3D);
mychart.setLegendPosition(ChartLegendPosition.EAST);
mychart.setColorOrder(ChartColorOrder.STRAIGHT);
//mychart.setTooltip ("Stocks Info");
mychart.setAxisMaxVal(5000);
mychart.setAxisMinVal(0);
mychart.setHeight("300");
mychart.setWidth("800");

chartmodel.addItem("Stocks",Row2,Row3,"yellow","Extn","Available Stocks for
Users");
```

```
}

/* THE API FOR METHODS IN VECTOR CHART MODEL

public void addItem(String groupid,
                   String x,
                   String y,
                   String color,
                   String extension)adds an item to the list of choices.

Parameters:
groupid - the item's group name
x - the item's category name
y - the item's value
color - the item's user defined color
extension - the item's extension data for HREF and ALternate text

*/

mychart.setModel(chartmodel);
myForm.addComponent(mychart);

}
catch(Exception E)
{
E.printStackTrace();
}
}
```

Tricks and Tips

MS Excel Access

Using POI-HSSF API

Suppose we should not to expose the whole data provided in our data base to users then we have to create and Excel sheet taking the data from our MS Excel DB. For this purpose we use JAVA APIs which helps in creating an excel file through code.

HSSF (Horrible Spread Sheet Format API) provides ways to read spreadsheets create, modify, read and write XLS spreadsheets. HSSF is the POI Project's pure Java implementation of the Excel '97(-2007) file format. The few implementations are to create Business Plan, Calendar, Timesheet and Calculator.

The implementation is this has been provided in this blog by Mr. Prakash Singh.

<https://www.sdn.sap.com/irj/scn/weblogs?blog=/pub/wlg/1390>

This is for EP 6.0 version which will create an Excel file from JAVA API. Now this file can be added with data and then used for modifying and displayed through charts.

Implementation of HTMLB

Intro

HTMLB (HTML-Business for Java) provides a full set of easy-to-use Web controls. HTMLB allows a design-oriented page layout. It is designed to overcome typical servlet problems. HTMLB provides the technological infrastructure for easy Customer Branding also.

Form is basically the wrapping paper of your page and essential for the data transfer from the web client to the web browser and for the event handling. Controls in the form must have unique control names. Controls are placed in a form, which are GUI elements that build the application.

A chart displays data that are relevant for the user in a graphical representation so that the characteristics of the data and their relations are easy to capture for the user. Typical functionality, which charts offer, are:

- Switch between chart view and table view
- Zooming and panning (nor available for the chart control)
- Drill-down

To use the functionality of Switching between chart and table the below is one of the best links I found. This uses Visual Composer for this task.

<https://www.sdn.sap.com/irj/scn/weblogs?blog=/pub/wlg/6910>

Vector Chart Model:

The below link of SAP Help documentation has the required methods for using the chart model.

<http://help.sap.com/javadocs/nw04s/current/hb/com/sapportals/htmlb/VectorChartModel.html>

These methods help us with the implementation of the code. Here in our present implementation we tried to show a data to users in chart format i.e. 3D-BAR chart. We tried to show the Available stocks and the Users list at the charts.

My Database has Only 3 columns namely, Access Number, Users and Stocks. I tired to show the users against their available stocks. Now if we are have huge amounts of data, we need to create our own excel file (hssf) with required data and show it across our charts. Once our excel is created we can either use Style sheets for all the fields available at the file using the below code

After we follow the blog we will be able to create excel sheet with values which are added in random. If we want to typically use the data provided in our MS Excel database then use the below code which is part of a switch case and includes all the cases which form the columns of our newly created Output File System.
HSSFCell cell = null;

case 1:

```
HSSFCellStyle style1=wb.createCellStyle();
style1.setFillForegroundColor(new HSSFColor.GREY_80_PERCENT().getIndex());
style1.setFillBackgroundColor(new HSSFColor.VIOLET().getIndex());
style1.setFillPattern(HSSFCellStyle.SOLID_FOREGROUND);
cell =row.createCell((short)1);
cell.setCellStyle(style1);
cell.setCellValue(resultset.getString(i));
//GET THE VALUE FROM OUR EXCEL DB INTO OUTPUT FILE STREAM
out = new FileOutputStream("path");
wb.write(out);break;
```

// REPEAT THIS FOR ALL THE COLUMNS AVAILABLE

Hence we will obtain our excel sheet with style patterns included. Output will be in a more sober form.

Troubleshoot

Issues at using DSN-less connections

My Wiki

Please check the below wiki which I wrote for helping us in solving issues which we face during External Database Connectivity

<https://www.sdn.sap.com/irj/scn/wiki?path=/display/EP/Using+DSN-less+Connectivity+from+Portal>

Summarizing Wiki Contents:

As i was implementing the methods for accessing an External Database from SAP Enterprise Portal, i was held with many unforeseen issue for establishing the connectivity. Though i added the needed jar files at the java build path these exceptions arised. So i thought it would be a good option to blog these down so that it might be useful for all of us who are trying to implement such scenarios.

Introduction: DSN-less connectivity means using a connection string that does not use a data source name (DSN) to connect to an external data source. Similarly manner we use DNS-less connectivity for an External SQL server Database and in Microsoft Excel. DSN-less connections resolve the issue of maintaining data sources on several computers. All of the necessary information is stored in the workbook itself.

Scenario 1:

We will connect to SQL server which is in the same domain. Same Domain means both servers exists in same domain for e.g. If portal is in <http://<hostname>.company.com/irj/portal> then the external database also should be in <http://<hostname>.company.com>. We will be using the JDBC SQL driver for connecting to the External Database.

Scenario 2:

We will connect to Microsoft Excel DB file at server which is also at same domain. We will use JDBC-ODBC drivers for this connectivity.

These error are seen at the logs and traces at SAP NetWeaver Administrator. The following is the path to access 'logs and traces'- <http://<HOSTNAME>:<PORTNUM>/index.html>. Login to Portal with Admin

credentials, then at the Tab System Management->Monitoring now at Detailed Navigation->Logs and Traces-> Show view (Default Trace).

Issues

Exception 1: java.sql.SQLException: [Microsoft][SQLServer 2000 Driver for JDBC]Error establishing socket.

Cause: There are many common reasons why this error can occur.

Resolution: Check if the server which you are trying to access is a Clustered environment server or stand alone server. The server name/IP is incorrect. To test basic connectivity to your server, use the **ping** command from a command prompt. This will verify that you have network connectivity to that server, but it will not confirm if the SQL Server can accept connections.

The server name might be correct, but the server is not running. You can verify whether or not your server is running from the SQL Server Network Utility on the server, or from another client utility such as SQL Query Analyzer.

The server name is correct and it is running, but the server does not have TCP/IP enabled. Check for the open port which is available for connectivity. Use that port number at the connection string. Also we need to ensure that the port number which we are using for the connection string is apt for TCP/IP. If the problem is intermittent, it might be due to network problems or due to server problems (for example, the server is occasionally too busy to accept a connection).

Exception 2: java.lang.ClassNotFoundException: com.microsoft.jdbc.sqlserver.SQLServer1Driver

-----ClassLoader name:

[com.sapportals.portal.prt.util.ApplicationClassLoader@10acd87] Parent loader name:

[com.sapportals.portal.prt.util.ApplicationClassLoader@136fcbc] References: not registered! Resources:

D:\usr\sap\PE7\JC01\j2ee\cluster\server0\apps\sap.com\irj\servlet_jsp\irj\root\WEB-INF\portal\portalapps\trigger_extdatabase\private\lib\<APPLICATION>.jar

Cause: If driver itself was incorrectly given this particular issue arises.

Resolution: In the error itself, the incorrect driver will be mentioned which will give us a scope to correct it

Exception 3: java.sql.SQLException: [Microsoft][SQLServer 2000 Driver for JDBC][SQLServer]Login failed for user 'USER'.

Cause: If the server login credentials are incorrect. Login failed for that user means either the user doesn't have permissions to access it or the user which was provided is incorrect.

Resolution: Verify the User. Try to login SQL Query Analyzer with the same user. This will prove if we were correct or not.

Exception 4: java.sql.SQLException: [Microsoft][SQLServer 2000 Driver for JDBC][SQLServer]Line 1: Incorrect syntax near '<columnname>'.

Cause: If the SQL query string is incorrect. Syntax error refers to error at SQL statements.

Resolution: Check for the SQL statements. If in case of dilemma run the same at SQL Query Analyzer and check for apt statements.

These are few of the problems which we face during accessing SQL server Tables from SAP Enterprise Portal though we have added the needed jars like **msbase.jar, msutil.jar, mssqlserver.jar** . Use the below code snippet for proper connectivity.

```

Statement stmt = null;
ResultSet rs = null;
Connection con=null;
String COLUMN1,COLUMN2=null;

try
{

Class.forName("com.microsoft.jdbc.sqlserver.SQLServerDriver");
//response.write("driver loaded");
con=DriverManager.getConnection("jdbc:microsoft:sqlserver://HOSTNAME:1PORTNUM;DatabaseName=<NAME>","<ID>","<PWD>");

//response.write("got connected ");
stmt = con.createStatement();
rs = stmt.executeQuery("Select COLUMN1 from TABLENAME where COLUMN2= ");
}
catch(Exception ex)
{
ex.printStackTrace();
}

```

For the **Scenario 2**, Even at Accessing Excel file we might face issues though we have written the code with no errors.

Exception 5: java.sql.SQLException: [Microsoft][ODBC Excel Driver]General error Unable to open registry key 'Temporary (volatile) Jet DSN for process 0x840 Thread 0xdac DBC 0x99cc81c Excel'.

Cause: The account that is being used to access the page does not have access to the registry key i.e. the user has no permissions or full control at the file.

Resolution: This can happen right from the start, because proper permissions were never applied. However, it can also happen when a compact and repair operation against the MDB file resets permissions. So, if our database is getting corrupted often enough, we might find that compacting and repairing the database file also leaves us with this error. At the folder level give full controls to the USER whose credentials we are using to try accessing that file.

Exception 6: java.sql.SQLException: [Microsoft][ODBC Driver Manager] Data source name not found and no default driver specified

Cause: The Data Source name i.e. DB name that is being used to access was not correctly provided. And also the default Driver was not mentioned correctly.

Resolution: Make sure that the provided Database name is apt as it is case sensitive. Use the default id which is present fro the MS excel file.

The following is the code snippet can be used for accessing excel file from Portal using JDBC-ODBC drivers.

```

String Connection_str=null;
String excel_path="<PATH>.xls";
Statement stmnt=null;
Connection excel_conn=null;
ResultSet excel_result=null;

try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

```

```
Connection_str="jdbc:odbc:Driver={Microsoft Excel
Driver(*.xls)};DBQ="excel_path";DriverID=22;READONLY=false";
excel_conn=DriverManager.getConnection("Connection_str,"<ID>","<PWD>");
stmt=excel_conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_UPDAT
ABLE);
}
catch(Exception E)
{
E.printStackTrace();
}
```

Important Note

Obtain Result Set

One vital point to be noted here is that when trying to execute SQL statements here use the Sheet name appended by *'\$'.*

For example:

```
ResultSet excel_result =stmt.executeQuery("select * from [SHEET NAME$]");
```

Hope this might be useful for all who are using these drivers.

The following are the other related issues at MS Excel Access

Though we established the connectivity for accessing the MS Excel DB from portal we will face issues at the output. These are mainly because of Security issues at using the file at the server.

Security Permissions

We should be provided with the folder level access to the files at the server for establishing proper connectivity. "Full Control" on the file is of utmost importance. Data contents cannot be viewed resulting in Unable to open registry key error.

Also there is a chance that data file might get corrupted which will result in access denied for that file. So we need to be careful with data provided and stored in our database.

Related Content

<https://www.sdn.sap.com/irj/scn/wiki?path=/display/EP/Using+DSN-less+Connectivity+from+Portal>

<https://www.sdn.sap.com/irj/scn/weblogs?blog=/pub/wlg/1390>

http://help.sap.com/saphelp_nw70/helpdata/EN/4b/9b0e41a346ef6fe1000000a1550b0/frameset.htm

For more information, visit the [Portal and Collaboration homepage](#).

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.