

How to Populate Custom UME Properties from Excel File

Applies To:

SAP NetWeaver 2004s (EP7)

Summary

This article is about feeding user data automatically in the SAP Enterprise Portal. The client may have some excel file to hold all the user specific data (e.g. Expert Area). This data needs to be populated in the SAP Enterprise Portal, such that users can easily find the experts in different fields. Custom UME attributes can easily be created and tagged with the users. But the out-of-the-box features provided by SAP Enterprise Portal does not allow to import custom user attribute values from the excel file (available with the client).

Author: Mr. Sudip Majumder

Company: TCS – Tata Consultancy Services

Created on: 22 August 2007

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 Details	3
Transforming the Excel File	3
Adding Custom UME Attributes	3
Creating Portal Project and Abstract Portal Component	3
Code Details.....	4
PAR File Deployment and iView Creation	5
Conclusion	6
Related Content.....	7
Disclaimer and Liability Notice.....	8

Problem Description

This article is about feeding user data automatically in the SAP Enterprise Portal. The client may have some excel file to hold all the user specific data (e.g. Expert Area). This data needs to be populated in the SAP Enterprise Portal, such that users can easily find the experts in different fields. Custom UME attributes can easily be created and tagged with the users. But the out-of-the-box features provided by SAP Enterprise Portal does not allow to import custom user attribute values from the excel file (available with the client).

Solution Details

Please follow the below mentioned steps to accomplish the task –

Transforming the Excel File

We will not directly use the excel sheet that is provided by the client. We need to transform this file in CSV format such that we can easily feed it for our program. This excel sheet contains the user detail along with the custom user specific data. For simplicity assume only one custom attribute is provided i.e. ExpertArea.

The excel provided by the client looks like this –

User Id	User Name	E-Mail	Expert Area
c01234	James Law	j.law@abc.com	Java
c01235	Mathew Right	m.right@abc.com	SAP
c01236	Kim Shaw	k.shaw@abc.com	Oracle
c01237	Daren Morgan	d.morgan@abc.com	UNIX

User Id : Logon Id of the user in the portal.

User Name : Name of the user. This field is not required for this program, the user is already created earlier.

E-Mail : e-mail id of the user. This field is also not required for us in this program.

Expert Area : The area of expertise. We are going to populate the value corresponding to the mapped user.

We will now convert this excel file into a CSV file.

Please perform the following steps –

- Open the excel file with Microsoft Excel.
- Click File → Save As
- Select CSV from the dropdown and provide a name (*UME_Attributes_CSV.csv*).
- Keep the file in the portal server, e.g. we kept it in C:\CSV location.

Adding Custom UME Attributes

We are about to populate the custom user attribute value from the excel file. But this custom attribute needs to be created first. I am not providing the detail information for creating the custom UME attributes. Documents are already present in SDN in this topic. Please search with “How to Extend User Details” in SDN. You can also get the document from the link below -

<https://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/bce8dfbb-0d01-0010-15b0-97e20b8aed96>

Creating Portal Project and Abstract Portal Component

Please follow the steps to create a portal project and add an abstract portal component –

- Open NetWeaver Developer Studio.
- Click File → New → Other → Portal Application → Create a new Portal Application.
- Provide a meaningful name (*FileRead*) and location for the project.
- Add a portal component by clicking File → New → Other → Object → (Select Project) → Abstract Portal Component.
- Provide a meaningful name (*FileRead*).

Code Details

In the abstract portal component, we will modify the *doContent* method. Please follow the steps –

- Get the File Input Stream and Data Input Stream from the input file -

```
FileInputStream fis = new
FileInputStream("C:\\\\CSV\\\\UME_Attributes_CSV.csv");
DataInputStream din = new DataInputStream(fis);
```

- Read each line of the file by the function *readLine()*. The output of this line needs to be stored in a String variable.

```
String strLine = din.readLine();
```

- The full content of the line is now stored in the String *strLine*. This string contains all the necessary values separated by comma (','), We need to manipulate the string and extract each and every value from it. You can leave the first line; as in our example, the first line contains the header (User Id, User Name etc.).
- The second *din.readLine()* function call will give the string that contains the user information. Each call to the function *din.readLine()* will give the next line of text. So, call this function in a loop until the end of file is reached. When the end of file is reached, the function will return null.
- Now we will concentrate on the retrieved text. The first user value that is returned is "c01234,James Law,j.law@abc.com,Java". Use java String functions to differentiate this comma (','), separated values and put them into different strings. So, we obtained four strings –

```
string1 = "c01234"
string2 = "James Law"
string3 = "j.law@abc.com"
string4 = "Java"
```

- We will now update the UME attribute *ExpertArea* and populate its value as "Java" corresponding to the user authenticated by logon id "c01234". Refer to the following code details –

```
IUserFactory userFact = UMFactory.getUserFactory();
IUser user;
user = userFact.getUserByLogonID(string1);
IUserMaint userMaint =
(IUserMaint)userFact.getMutableUser(user.getUniqueID());
```

```
String strExpertArea[] = {string4};
userMaint.setAttribute("com.sap.security.core.usermanagement",
"ExpertArea", strExpertArea);
userMaint.commit();
```

- Follow the above mentioned steps until all the lines are reached.
- You may print some texts in between your code that may work as log, e.g.

```
response.write("Start of Program");
.....
response.write("UME Attributes are updated for User ID" + string1);
.....
response.write("End of Program");
```

PAR File Deployment and iView Creation

- Deploy the application *FileRead* using PAR file Export.
- Log on to the SAP Enterprise Portal where you have deployed the application.
- Now create an iView using Portal Component.
- Select your component (*FileRead*).
- Create the iView and see the preview.
- The program now populates the custom UME attributes from the csv file placed in the specified location mentioned above. You can also view the logs (that you entered previously) displayed in the iView.

Conclusion

Now we are able to upload custom user attributes easily in portal UME database from the excel file containing the main user information. You can apply some extra caution in the code to validate the user id, obtained from the file. You can do it easily by implementing a try-catch block when you are about to get the IUser object. i.e.

```
try
{
    user = userFact.getUserByLogonID(string1);
}
catch(Exception e)
{
    response.write("Invalid User Id : " + string1);
}
```

Related Content

sdn.sap.com

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.