



PeopleSoft Development Session

Judi Doolittle
Sandia National Laboratories

Speaker Qualifications

- Judi Doolittle, MA, OCP
- Author:
 - “PeopleSoft Developer's Guide for PeopleTools & PeopleCode”
- IOUG Board Member:
 - Executive Vice President
 - Director of RUGs
- PeopleSoft Team Lead:
 - Pension Module
- Developer
 - Benefits Open Enrollment
 - Service Bridging

Agenda

- XML
 - Major Components of XML
 - SOAP
 - XML Publisher for PeopleSoft
 - Publisher Security
 - Flat File Layout
 - Tags
 - Basic Setup
 - Report Categories
 - Data Source Creation
 - XML Application Packages
 - Register the Data Source
 - XML Files in SQL Dev
 - Putting it Together

Agenda Continued

- Tips and Tricks
 - Helpful Links
 - Email from Application Engine
 - Open Enrollment Email
 - PL/SQL
 - PL/SQL in SQR
 - SQR Template
 - SQC Example
 - UTIL_FILE
 - How to find bad SubPages
 - Finding Hidden Folders
 - Where is email value stored?

Agenda Continued

- Tips and Tricks Continued
 - How to find Message Catalogs by date
 - Find what access a Role gives a User
 - Find where the Component is in PeopleSoft Portal and Permission List
 - Find where the Page is in the PeopleSoft Portal System
 - Secrets of PSAUTHITEM
 - Security Overview of primary table relations
 - XLat changes in 8.9
 - Implementation of Roth 401(k)

Agenda Continued

- Component Interface Quick Reference
 - PeopleCode Example
 - Second PeopleCode Example

XML Basics

- Basic file: 12345, John Smith, 0987
 - Three pieces are EMPLID, NAME, and DEPTID
 - Pieces of information in the flat file coma delimited

<Employees>

<Employees>

<EMPLID>12345</EMPLID>

<NAME>John Smith</NAME>

<DEPTID>0987</DEPTID>

</Employees>

</Employees>

XML Basics Continued

- `<>` Markup Elements
- Outside brackets is Data
- Each piece of Data has a starting and ending tag
 - No limit to tags you can
 - Create
 - Or Rename
 - Example of renaming NAME to ENAME

`<Employees>`

`<Employees>`

`<EMPLID>12345</EMPLID>`

`<ENAME>John Smith</ENAME>`

`<DEPTID>0987</DEPTID>`

`</Employees>`

`</Employees>`

Major Components of XML:

- Root Element--Encloses all the Information
- Start Tag--XML elements have a start and end tag
- End Tag--XML elements have a start and end tag
- XML Element--The combination of a start and end tag are collectively a XML Element
- Data--Information between start and end tags

Major Components of XML:

- XML Declaration--the general characteristics of the document
- DTD Document Type Declaration--the structure of the document; internal declarations local to the XML document
- DTD Internal Subset--DTD contains references to other DTD
- XML Information Set--Document content

SOAP

- Simple Object Access Protocol
- SOAP applications can be developed in separate unrelated systems
- These separate applications exchange information
- Standards: <http://www.w3.org/TR/soap/>
- Good Training Site:
<http://www.soapuser.com/basics3.html>

SOAP Example

Host: www.widget.com

Content-Type: text/xml: charset="utf-8"

Content-Length: 560

SOAPAction: "http://www.widgets.com/assemblysearch"

<SOAP-ENV:Envelope—SOAP message's encoding style

xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelop/"

<SOAP-ENV:Body>

<m:GetAssembly xmlns:m="http://test.widgets.com">

<assembly_docs>K2</assembly_docs>

<model_number>widget_2</model_number>

</m:GetAssembly>

</SOAP-ENV:Body>

</SOAP-ENV:Envelope>

SOAP Response

<SOAP-ENV:Envelope

 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"

 SOAP-

 ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">

 <SOAP-ENV:Body>

 <m:GetAssemblyResponse xmlns :m="http://test.widgets.com">

 <AssemblyApprove>John Smith</AssemblyApprove>

 <m:GetAssemblyResponse>

 </SOAP>-ENV Body>

</SOAP-ENV:Envelope>

SOAP Elements

- Complex Elements
 - <definitions>
 - Related Services
 - <types>
 - Allows for specification of low-level data-typing
 - <message>
 - Individual pieces of communication
 - <porttype>
 - Grouping Element
 - <binding>
 - Link between physical and logical model
 - <soap:operation>
 - Transmission method i.e. SMTP
 - <Service>
 - Physical location of the end-point communication

XMLP

- Java based reporting platform
 - XML Publisher
 - Or XMLP
- Template Formats
 - HTML
 - Excel
 - PDF
 - RTF

XML Publisher Security

- Three Parts to XMLP Security
 - Report Definitions
 - Report Running
 - Report Viewing
- Query Defined Reports
 - Underlying Query Security Governs Report
 - XML Does Not Provide This Level of Security
- Security is Governed by:
 - Query Security
 - Application Security
 - And Or Process Scheduler Security

XML Publisher Security

- Queries: Query Security is the Deciding Security
- Data Extraction: Component Security Determines Access
- Processes: Process Security is the Key

Flat File Layout

- Each piece of information is separated by a coma
- Example:

12345, John Smith, 0987

<Employees>

<Employees>

<EMPLID>12345</EMPLID>

<NAME>John Smith</NAME>

<DEPTID>0987</DEPTID>

</Employees>

</Employees>

Tags

- No Limit to the Number of Tags You Can
 - Use
 - Create
 - Or Rename
- Example you want to rename NAME to ENAME

<Employees>

<Employees>

<EMPLID>12345</EMPLID>

<ENAME>John Smith</ENAME>

<DEPTID>0987</DEPTID>

</Employees>

</Employees>

Basic Setup

- Settings are Defined in the xdo.cfg File
- Example:

```
<config version="1.0.0" xmlns="http://xmlns.oracle.com/oxp/config/">
  <properties>
    <!-- System level properties -->
    <property name="xslt-xdoparser">true</property>
    <property name="xslt-scalable">true</property>
    <property name="system-temp-dir">CACHE</property>
    <property name="system-cachepage-size">50</property>
    <!-- PDF compression -->
    <property name="pdf-compression">true</property>
    <!-- PDF Security -->
    <!-- <property name="pdf-security">true</property> -->
    <!-- <property name="pdf-open-password">user</property> -->
    <!-- <property name="pdf-permissions-password">owner</property> -->
  </properties>
</config>
```

Basic Setup

- `<property name="system-temp-dir">CACHE</property>`
this setting is the cache of the application server
- for example on my machine that is set up for very simple location of `\\PS\CACHE`
- More Complex Example:
`[PS_HOME]/appserv/[DOMAIN_NAME]/CACHE`
- can also specify the application server or process scheduler domain in the `xdo.cfg` you need only update the JavaVM

Basic Setup

- psappsrv.cfg before change:

;To pass java options to the jvm loaded via JNI, uncomment JavaVM Options=
;and set it equal to the list of parameters you would like to use.

;The options need to be a blank space separated list of valid java options.

;Consult your JRE vendor for valid java options.

;For example, JavaVM Options=-Xmx256m -Xms128m

JavaVM Options=-Xrs -Dxdo.ConfigFile=../xdo.cfg

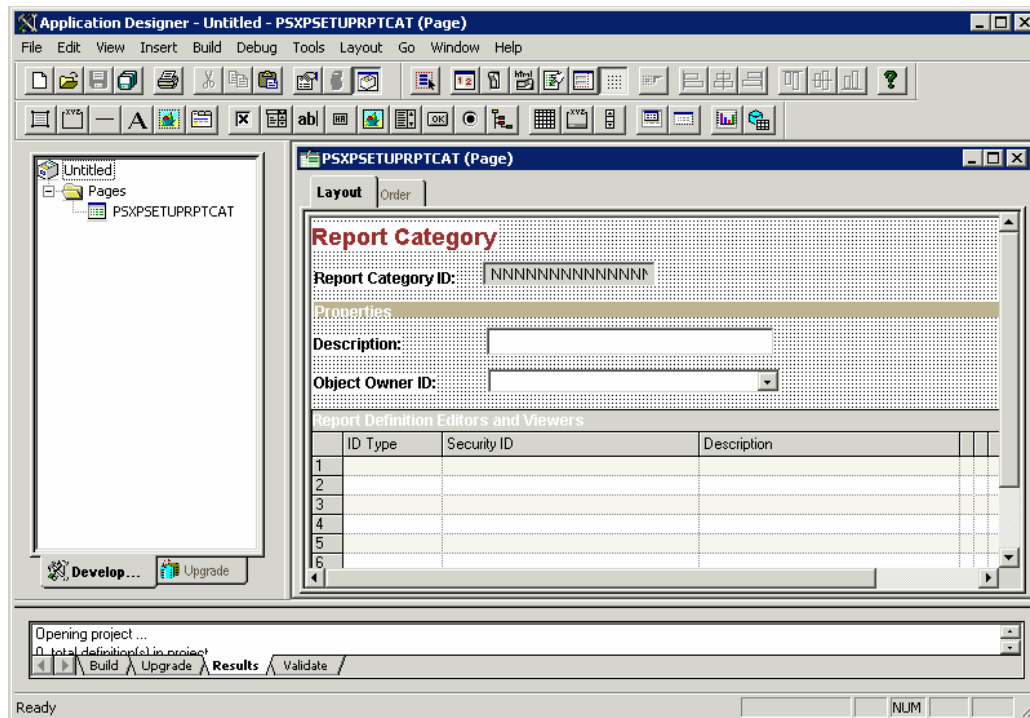
- New Setting:

JavaVM Options=-Xrs -Dxdo.ConfigFile=xdo.cfg

- Changes will not take effect until you restart the Application Server

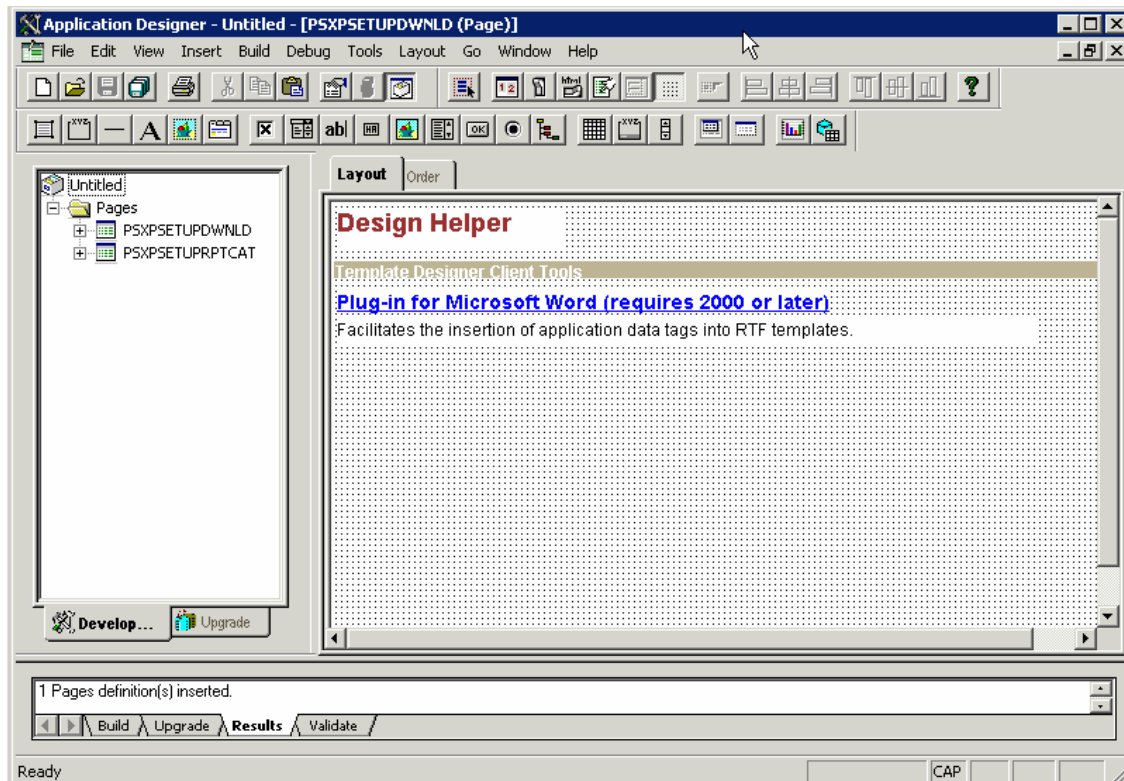
Pages Used in Setup

- Two Pages Used in Set-up of XML Publisher in PeopleSoft
- PSXPSETUPRPTCAT -- the Report Category Page



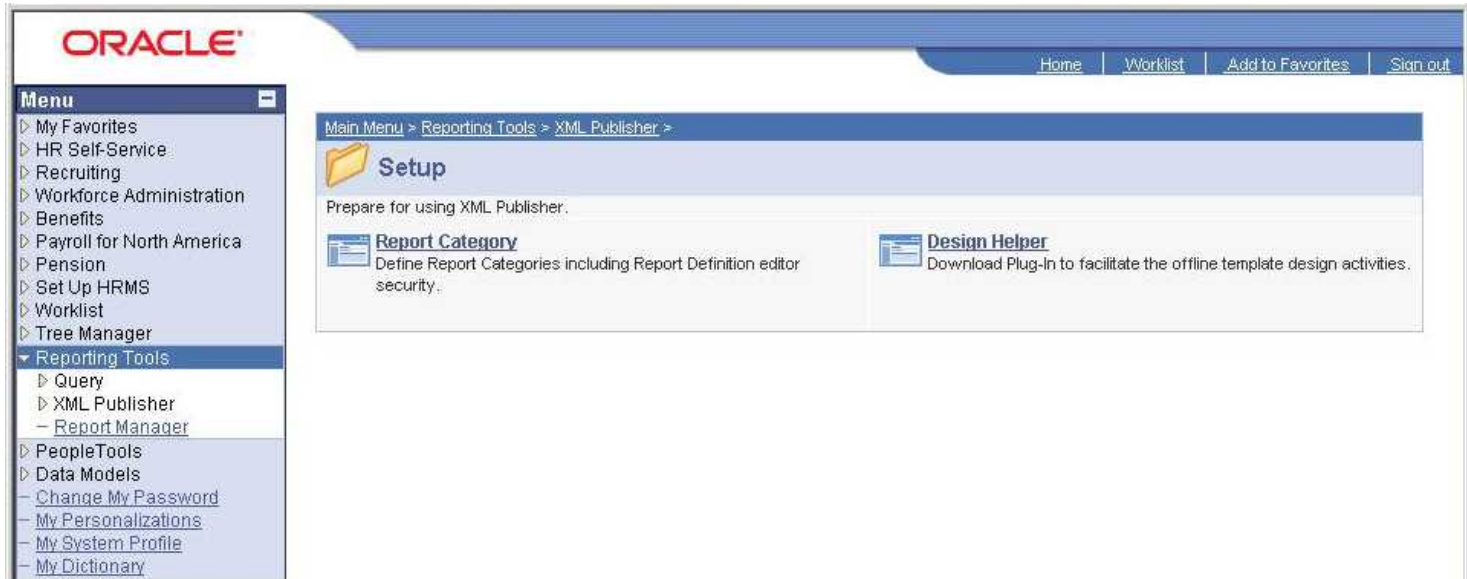
Pages Used in Setup Continued

- Second Page in Set-up of XML Publisher in PeopleSoft
- PSXPSETUPDWNLNLD -- the Design Helper Page



Report Categories Continued

- Set-up Folder
 - Contains:
 - Report Category
 - Design Helpers



Report Categories

- Report Definitions Required
- All Templates Stored in Content Library Including Sub-Templates
- Set Row Level Security to the level of the Component



The screenshot shows the Oracle XML Publisher web interface. On the left is a navigation menu with the following items: My Favorites, HR Self-Service, Recruiting, Workforce Administration, Benefits, Payroll for North America, Pension, Set Up HRMS, Worklist, Tree Manager, Reporting Tools (expanded), Query, XML Publisher, Report Manager, PeopleTools, Data Models, Change My Password, My Personalizations, My System Profile, and My Dictionary. The main content area is titled "XML Publisher" and contains several components: Data Source (Register schema and sample data files for XML Publisher data sources), Report Definition (Define reports by associating Data Sources with layout Template files), Content Library (Create and maintain reusable Sub-Template definitions), Translations (Register Translation XLIFF files for Report RTF templates and Content Library RTF sub-templates), Query Report Viewer (Run Query-based XMLP reports real-time. View in a window), Query Report Scheduler (Schedule Query-based XMLP reports. Use to run burst reports), XMLP Report Search (Locate XML Publisher report output using enhanced search criteria), and Setup (Prepare for using XML Publisher). The Setup section includes links for Report Category and Design Helper. The top of the page features the Oracle logo and navigation links: Home, Worklist, Add to Favorites, and Sign out.

Report Categories Continued

- Report Category in Payroll with Two PeopleSoft Roles Defined

Report Category

Report Category ID: TC_TEST

Properties

Description: XML Test

*Object Owner ID: Payroll

Report Definition Editors and Viewers

View All First 1-2 of 2 Last

*ID Type	*Security ID	Description	Read Only
Role	Resource Manager	[System] Resource Manager	☐
Role	Benefits Administrator	[WF] Benefits Administrator	☒

Save Notify

Add Update/Display

Report Categories Continued

- Key Information in Creating Report Category
 - Report Category ID
 - Description
 - Object Owner ID
 - ID Type
 - Security ID
 - Description
 - Read Only

Report Categories Continued

- Report Category

Definition	Template	Output	Security
Report Name: PYW2C07S_EE			
Data Source			
Data Source Type: XML File			
Data Source ID: PYW2C07			
Data Source Description: W2C PDF Form for 2007			
Report Properties			
Report Description: W2c Employee SS Form 2007			
*Report Status: Active			
*Report Category ID: ALLUSER ALL PeopleSoft User			
Object Owner ID: Payroll			
Template Type: ☒ PDF ☐ RTF ☐ eText ☐ XSL			
Registered Date/Time: 08/09/2007 3:34:28PM Registered By: PS			
Updated Date/Time: 11/13/2007 12:59:31PM Updated By: PPLSOFT			
Download: Data Schema Sample Data			
Return to Search Previous in List Next in List Add Update/Display Include History Correct History			
Save			

Data Source Creation

- Two Main Pieces to XML Reports
 - PeopleSoft Developer usually Does the Data Piece
 - Functional Business Team Member does Template
- Different Types of Data Sources:
 - PeopleSoft Queries
 - Rowsets
 - XML Document Objects
 - XML Files
 - XSD Files

XML Data Source

- RowSet
- Example taken from website <http://peoplesofttipster.com/2007/05/24/xml-publisher-within-peoplesoft/>

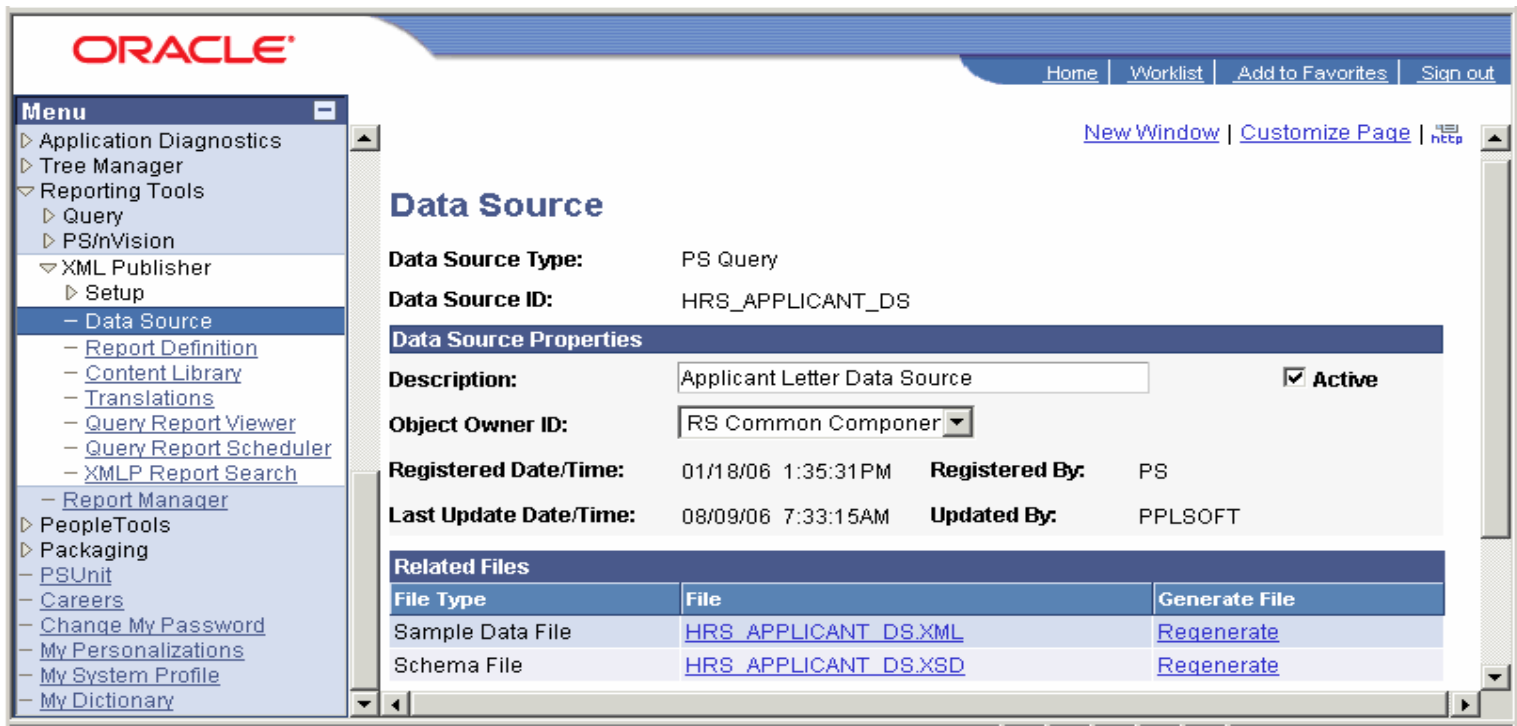
```
/*Populating the RowSet */
import PSXP_XMLGEN:*;
/*create rowsets */
&rs_compensation CreateRowSet(Record.COMPENSATION);
&rs_job CreateRowSet(Record.JOB);
/*Fill Parent */
&rs_job.Fill("where EMPLID like 'K0G00%'");
/*Fill Child */
For &i = 1 to &rs_Job.ActiveRowCount
    &row = &rs_Job.GetRow(&i);
    &rs = &row.GetRowSet(Scroll.COMPENSATION);
    &rs.Fill("WHERE EMPLID = :1", &row.JOB.EMPLID.Value;
End-For;
/*Output Files */
/*Create XSD */
&rds = create psxp_xmlgen:RowSetDS();
&myschema = &rds.GetXSDSchema(&rs_Job);
&f1 = GetFile("C:\temp\rpt01.xsd", "W", %FilePath_Absolute;
&f1.WriteLine(&myschema);
&f1.Close();
/*Create Sample XML File */
&myXMLFile = &rds.GetXMLData(&rs_Job, "c:\temp\rpt01.xsd");
&f2 = GetFile("c:\temp\rpt01.xml", "W", %FilePath_Absolute);
&f2.WriteLine(&myXMLFile);
&f2.Close();
/*Code taken from a presentation by Duncan Davies */
```

XML Application Packages

- XML Publisher Classes
 - import PSXP_RPTDEFNMANAGER:*;
 - import PSXP_REPORTMGR:*;
 - import PSXP_XMLGEN:*;
 - import PSXP_ENGINE:*;

Register the Data Source

- PeopleSoft Query
- Here we will only be looking at populating a Data Source from a Query
- Data Source Page
 - <Reporting Tools<XML Publisher<Setup<Data Source



ORACLE

Home | Worklist | Add to Favorites | Sign out

[New Window](#) | [Customize Page](#) | [http](#)

Menu

- Application Diagnostics
- Tree Manager
- Reporting Tools
 - Query
 - PS/nVision
 - XML Publisher
 - Setup
 - Data Source**
 - Report Definition
 - Content Library
 - Translations
 - Query Report Viewer
 - Query Report Scheduler
 - XMLP Report Search
 - Report Manager
- PeopleTools
- Packaging
- PSUnit
- Careers
- Change My Password
- My Personalizations
- My System Profile
- My Dictionary

Data Source

Data Source Type: PS Query

Data Source ID: HRS_APPLICANT_DS

Data Source Properties

Description: Applicant Letter Data Source ☒ Active

Object Owner ID: RS Common Component

Registered Date/Time: 01/18/06 1:35:31PM **Registered By:** PS

Last Update Date/Time: 08/09/06 7:33:15AM **Updated By:** PPLSOFT

Related Files

File Type	File	Generate File
Sample Data File	HRS_APPLICANT_DS.XML	Regenerate
Schema File	HRS_APPLICANT_DS.XSD	Regenerate

Register the Data Source Continued

- First Setting is the Data Source Type
 - Which Can Be:
 - PSQuery
 - Rowset
 - XML Doc
 - XML File
- Next Setting is the Data Source ID
 - When working with PSQuery
 - PS Pulls the Data Source ID from the Query Manager
 - Which is why it is important to have your Query Created before you register the Data Source
- Next is Description
- Object Owner is Optional
 - This is the:
 - Product
 - Feature or
 - Application that Owns the Data Source

Register the Data Source Continued

- Registered Date/Time
 - Read Only
 - Maintained by the System
 - Stores the Date and Time of the Last Update
- Active
 - Sets a Data Source to be Actively Chosen when Creating the Report Definition
 - If InActive--You will not see it when you try to Create a Report
- Registered By
 - UserID
 - Person who First Registered the Data Source
 - Updated by the System to Note Last Person to Update the Field

Register the Data Source Continued

- Create a Report Definition with the Query

ORACLE

Home | Worklist | Add to favorites | Sign out

[New Window](#) | [Customize Page](#) | [http](#)

Menu

- ▷ Student Financials
- ▷ Academic Advisement
- ▷ Contributor Relations
- ▷ Set Up HRMS
- ▷ Set Up SACR
- ▷ Enterprise Components
- ▷ Worklist
- ▷ Application Diagnostics
- ▷ Tree Manager
- ▷ Reporting Tools
 - ▷ Query
 - ▷ PS/InVision
- ▷ XML Publisher
 - ▷ Setup
 - [Data Source](#)
 - [Report Definition](#)
 - [Content Library](#)
 - [Translations](#)
 - [Query Report Viewer](#)
 - [Query Report Scheduler](#)
 - [XMLP Report Search](#)
 - [Report Manager](#)
- ▷ PeopleTools
- ▷ Packaging
 - [PSUnit](#)
 - [Careers](#)
 - [Change My Password](#)
 - [My Personalizations](#)
 - [My System Profile](#)
 - [My Dictionary](#)

Definition | Template | Output | Security | Bursting

Report Name: HRS_INT_LET

Data Source

Data Source Type: PS Query

Data Source ID: HRS_INTERVIEW_DS

Data Source Description: Interview Letter Data Source

Report Properties

Report Description: Interview Letters

***Report Status:** Active

***Report Category ID:** ALLUSER All PeopleSoft User

Object Owner ID:

Template Type: ☐ PDF ☒ RTF ☐ eText ☐ XSL

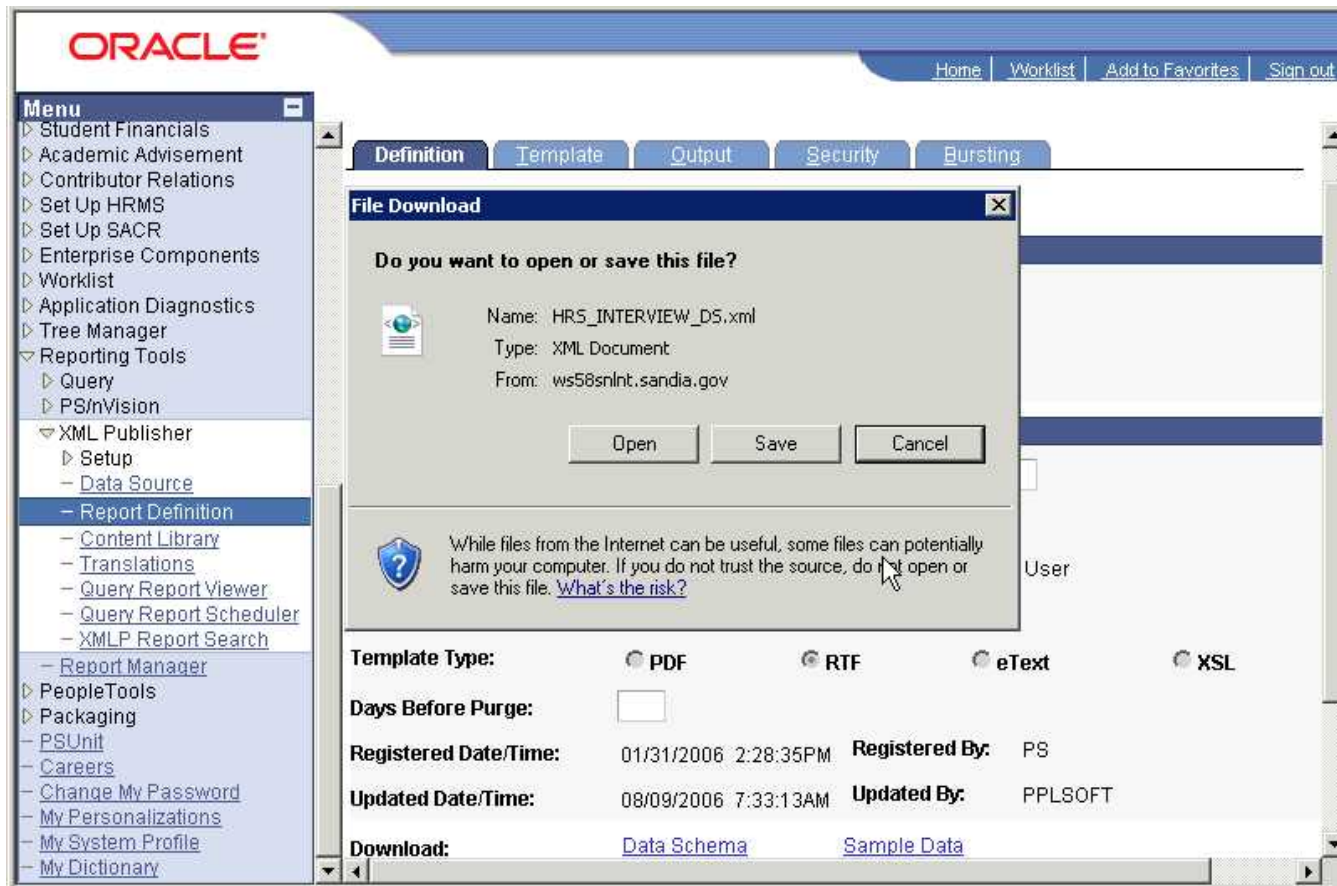
Days Before Purge:

Registered Date/Time: 01/31/2006 2:28:35PM **Registered By:** PS

Updated Date/Time: 08/09/2006 7:33:13AM **Updated By:** PPLSOFT

Register the Data Source Continued

- Create a Report Definition with the Query



XML Files in SQL Dev

- In the Highlighted Rows Right Click Export Results as XML File
- Example it Creates:

```
<?xml version="1.0" ?>
```

```
<ROWDATA>
```

```
<ROW>
```

```
<C0>1</C0>
```

```
<FIELDNAME>ABSENT_AFTER</FIELDNAME>
```

```
<FIELDVALUE>N</FIELDVALUE>
```

```
<EFFDT>1/1/1900</EFFDT>
```

```
<EFF_STATUS>A</EFF_STATUS>
```

```
<XLATLONGNAME>No</XLATLONGNAME>
```

```
<XLATSHORTNAME>No</XLATSHORTNAME>
```

```
<LASTUPDDTTM>9/5/1997</LASTUPDDTTM>
```

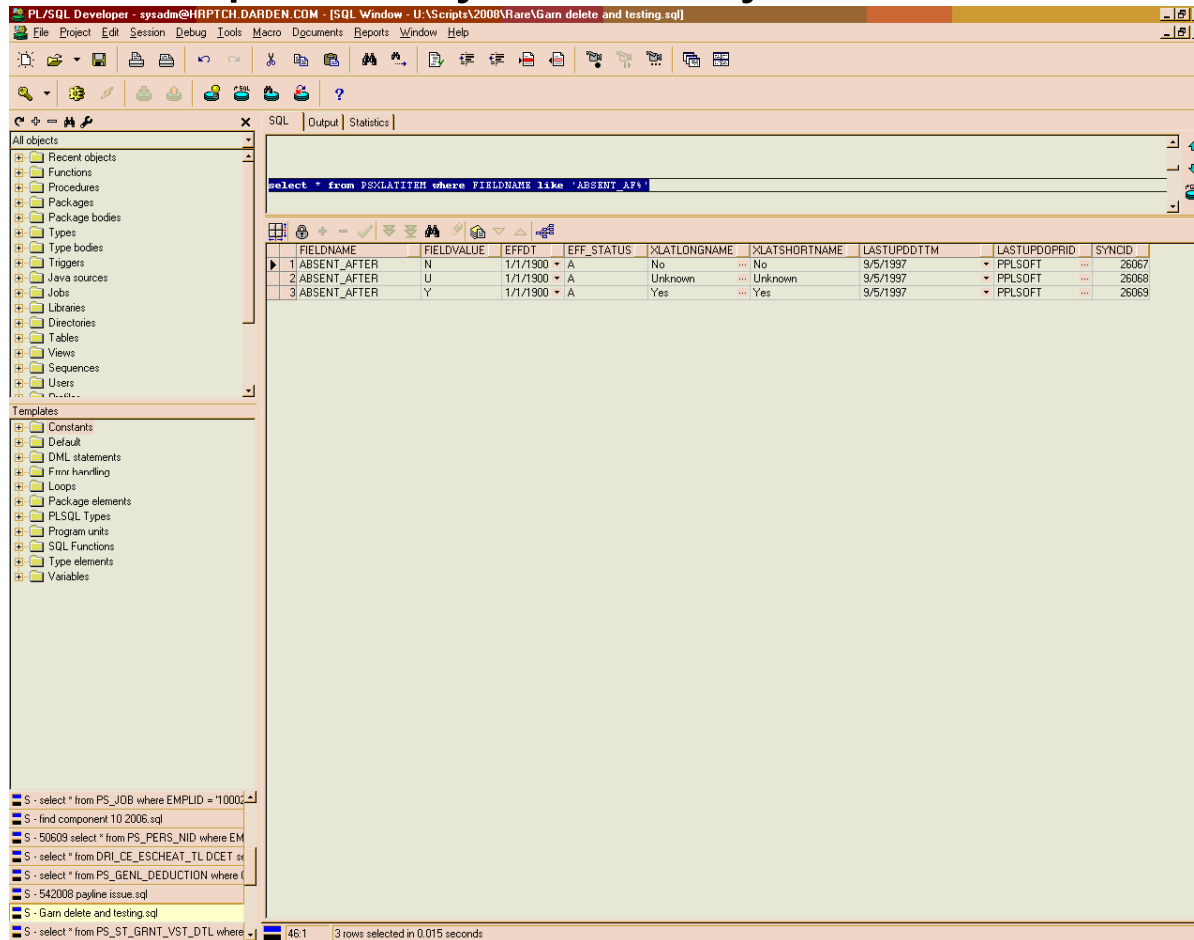
```
<LASTUPDOPRID>PPLSOFT</LASTUPDOPRID>
```

```
<SYNCID>26067</SYNCID>
```

```
</ROW> ....
```

XML Files in SQL Dev Continued

- First Step Make your Query



The screenshot shows the PL/SQL Developer interface. The main window displays the following SQL query:

```
select * from PSXLATTIH where FIELDNAME like 'ABSENT_AFI'
```

The results are displayed in a table with the following columns: FIELDNAME, FIELDVALUE, EFFDT, EFF_STATUS, XLATLONGNAME, XLATSHORTNAME, LASTUPDDTTM, LASTUPDDPRID, and SYNCID. The table contains three rows of data:

	FIELDNAME	FIELDVALUE	EFFDT	EFF_STATUS	XLATLONGNAME	XLATSHORTNAME	LASTUPDDTTM	LASTUPDDPRID	SYNCID
1	ABSENT_AFTER	N	1/1/1900	A	No	No	9/5/1997	PPLSOFT	26067
2	ABSENT_AFTER	U	1/1/1900	A	Unknown	Unknown	9/5/1997	PPLSOFT	26068
3	ABSENT_AFTER	Y	1/1/1900	A	Yes	Yes	9/5/1997	PPLSOFT	26069

The status bar at the bottom indicates: 46:1 | 3 rows selected in 0.015 seconds.

Example--Putting it Together

Records

Query

Expressions

Prompts

Fields

Criteria

Having

View SQL

Run

Query Name:

SL_XML_VISIT_FORM

Description:

Visit Forms

View field properties, or use field as criteria in query statement.

Reorder / Sort

Fields										Customize	Find	View All	First	1-35 of 35	Last
Col	Record	Fieldname	Format	Ord	XLAT	Agg	Heading Text	Add Criteria	Edit	Delete					
1	A	SL_SNLID - Sandia Issued ID	Char11				SNL_ID		Edit	-					
2		substr(A.SL_SSN,6,4)	Char11				SSN		Edit	-					
3	A	SL_LAST_NAME ', ' A.SL_FIRST_NAME ' ' substr(A.SL_MIDDLE_NAME,1,1)	Char30				A.SL_LAST_NAME ', ' A.SL_FI		Edit	-					
4		to_char(to_date(A.SL_BIRTHDATE, 'YYYY-MM-DD'), 'MM/DD/YYYY')	Char10				to_char (A.SL_BIRTHDATE, 'MM/DD		Edit	-					
5		to_char(to_date(2, 'YYYY-MM-DD'), 'MM/DD/YYYY')	Char10				START_DATE		Edit	-					
6		to_char(to_date(3, 'YYYY-MM-DD'), 'MM/DD/YYYY')	Char10				END_DATE		Edit	-					
7		'TO'	Char2				'TO'		Edit	-					
8		to_char(to_date(9, 'YYYY-MM-DD'), 'MM/DD/YYYY')	Char10				CURR_DATE		Edit	-					
9		'Electronically approved by ' B.SL_NAME_FORMAL	Char50				'Electronically approved by '		Edit	-					
10		'	Char1				'		Edit	-					
11	B	SL_SNLID - Sandia Issued ID	Char11				SNL_ID		Edit	-					
12	B	SL_NAME_FORMAL - Formal Name	Char108				Formal Name		Edit	-					
13	B	SL_DEPTNAME_ABBRV - Department Number	Char10				Dept Number		Edit	-					
14	B	SL_MAILSTOP - Mail Stop	Char4				sl_mailstop		Edit	-					
15	B	SL_WORK_PHONE - Work Phone #	Char24				Work Phone		Edit	-					
16	B	SL_FAX - Fax Number	Char24				Fax		Edit	-					
17	:	:4	Char30				LOCATION		Edit	-					
18	:	:5	Char50				PURPOSE		Edit	-					
19	:	:6	Char50				MEETING_ACCESS		Edit	-					
20	:	:7	Char30				SIGMA		Edit	-					
21	:	:8	Char20				SITE		Edit	-					
22	:	:10	Char3				COUNTRY		Edit	-					
23	:	:11	Char55				ADDRESS		Edit	-					
24	:	:12	Char12				COMP_PHONE		Edit	-					

Example--Putting it Together

◀
Prompts
Fields
Criteria
Having
View SQL
Run

Query Name: SL_XML_VISIT_FORM
 Description: Visit Forms

Add Prompt

Prompts List		
Prompt	Edit	Delete
:1 = EMPLID - ID	Edit	
:2 = START_DATE - Start Date	Edit	
:3 = END_DATE - End Date	Edit	

Query Continued

Records

Query

Expressions

Prompts

Fields

Criteria

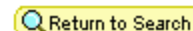
Having

View SQL

Run

Query Name: SL_XML_VISIT_FORM**Description:** Visit Forms**Query SQL:**

```
SELECT DISTINCT A.SL_SNLID, substr(A.SL_SSN,6,4), A.SL_LAST_NAME || ' ' || A.SL_FIRST_NAME || ' ' || substr(
A.SL_MIDDLE_NAME,1,1), to_char(to_date( TO_CHAR(A.SL_BIRTHDATE,'YYYY-MM-DD'), 'YYYY-MM-DD'), 'MM/DD/YYYY'), to_char(to_date
('2','YYYY-MM-DD'), 'MM/DD/YYYY'), to_char(to_date('3','YYYY-MM-DD'), 'MM/DD/YYYY'), TO', to_char(to_date('9','YYYY-MM-
DD'), 'MM/DD/YYYY'), 'Electronically approved by ' || B.SL_NAME_FORMAL, ' ', B.SL_SNLID, B.SL_NAME_FORMAL,
B.SL_DEPTNAME_ABBRV, B.SL_MAILSTOP, B.SL_WORK_PHONE,
B.SL_FAX, '4', '5', '6', '7', '8', '10', '11', '12', '13', '15', '16', '17', '18', '19', '20', '21', '22', '23', '24'
FROM SL_EP_PER_OWW A, SL_EP_PERSON B
WHERE A.SL_SNLID = :1
AND B.SL_SNLID = :14
```

A small icon of a floppy disk with a green arrow pointing to it, indicating a save action.[Save As](#)[New Query](#)[Preferences](#)[Properties](#)[New Union](#)A small icon of a magnifying glass with a blue circle around it, indicating a search action.

Query Continued

17 :4	Char30	LOCATION		Edit	-
18 :5	Char50	PURPOSE		Edit	-
19 :6	Char50	MEETING_ACCESS		Edit	-
20 :7	Char30	SIGMA		Edit	-
21 :8	Char20	SITE		Edit	-
22 :10	Char3	COUNTRY		Edit	-
23 :11	Char55	ADDRESS		Edit	-
24 :12	Char12	COMP_PHONE		Edit	-
25 :13	Char50	SL_AFFILIATION		Edit	-
26 :15	Char1	T_BADGE_YES		Edit	-
27 :16	Char1	T_BADGE_NO		Edit	-
28 :17	Char256	COMMENTS		Edit	-
29 :18	Char11	CO_HOST_EMPLID		Edit	-
30 :19	Char12	CO_HOST_PHONE		Edit	-
31 :20	Char12	CO_HOST_FAX		Edit	-
32 :21	Char50	CO_HOST_NAME		Edit	-
33 :22	Char10	MGR_SL_SNLID		Edit	-
34 :23	Char50	MGR_NAME		Edit	-
35 :24	Char12	MGR_PHONE		Edit	-

XML Data for Query

```
<?xml version="1.0" ?>
- <query numRows="10" queryname="SL_XML_VISIT_FORM" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="">
- <row rownumber="1">
  <EMPLID>EMPLID samp</EMPLID>
  <SSN>SSN sample</SSN>
  <FULL_NAME>FULL_NAME sample data</FULL_NAME>
  <SL_BIRTHDATE>SL_BIRTHDA</SL_BIRTHDATE>
  <START_DATE>START_DATE</START_DATE>
  <END_DATE>END_DATE s</END_DATE>
  <TO>TO</TO>
  <CURR_DATE>CURR_DATE</CURR_DATE>
  <SIGNITURE>SIGNITURE sample data</SIGNITURE>
  <SLASH>S</SLASH>
  <HOST_SL_SNLID>HOST_SL_SNL</HOST_SL_SNLID>
  <HOST_NAME>HOST_NAME sample data</HOST_NAME>
  <HOST_ORG>HOST_ORG s</HOST_ORG>
  <HOST_MAILSTOP>HOST</HOST_MAILSTOP>
  <HOST_PHONE>HOST_PHONE sample data</HOST_PHONE>
  <HOST_FAX>HOST_FAX sample data</HOST_FAX>
```

XML Data for Query Continued

```
<LOCATION>LOCATION sample data</LOCATION> --:4
<PURPOSE>PURPOSE sample data</PURPOSE> --:5
<MEETING_ACCESS>MEETING_ACCESS sample data</MEETING_ACCESS>--:6
<SIGMA>SIGMA sample data</SIGMA> --:7
<SITE>SITE sample data</SITE> --:8
<COUNTRY>COU</COUNTRY> --:10
<ADDRESS>ADDRESS sample data</ADDRESS>--:11
<COMP_PHONE>COMP_PHONE s</COMP_PHONE> --:12
<SL_AFFILIATION>SL_AFFILIATION sample data</SL_AFFILIATION> --:13
<T_BADGE_YES>T</T_BADGE_YES> --:15
<T_BADGE_NO>T</T_BADGE_NO> --:16
<COMMENTS>COMMENTS sample data</COMMENTS>--:17
<CO_HOST_EMPLID>CO_HOST_EMP</CO_HOST_EMPLID>--:18
<CO_HOST_PHONE>CO_HOST_PHON</CO_HOST_PHONE> --:19
<CO_HOST_FAX>CO_HOST_FAX</CO_HOST_FAX> --:20
<CO_HOST_NAME>CO_HOST_NAME sample data</CO_HOST_NAME>--:22
<MGR_SL_SNLID>MGR_SL_SNL</MGR_SL_SNLID> --:22
<MGR_NAME>MGR_NAME sample data</MGR_NAME> --:23
<MGR_PHONE>MGR_PHONE sa</MGR_PHONE> --:24
</row>
```

Report Definition

Definition**Template****Output****Security****Report Name:** SL_CLS_FORM

Data Source

Data Source Type: PS Query**Data Source ID:** SL_XML_VISIT_FORM**Data Source Description:** Query to populate Visit Forms

Report Properties

Report Description: Classified Visit Form**Report Status:** Active**Report Category ID:** ALLUSER All PeopleSoft User**Object Owner ID:** PeopleTools**Template Type:** ☒ PDF ☐ RTF ☐ eText ☐ XSL**Registered Date/Time:** 05/08/2008 11:53:50AM **Registered By:** bsando**Updated Date/Time:** 07/30/2008 2:41:56PM **Updated By:** bsando**Download:** [Data Schema](#) [Sample Data](#)[Return to Search](#)[Previous in List](#)[Next in List](#)[Add](#)[Update/Display](#)[Include History](#)[Correct History](#)

Report Definition Template

Definition
Template
Output
Security

Report Name: SL_CLS_FORM

Template
Find | View All
First 1 of 1 Last

Template ID: SL_CLS_FORM_1
☒ **Default Template**

Description: Classified Visit Form

Language Code: English
Channel: Web Posting

Template Files
Find | View All
First 1 of 1 Last

Effective Date: 07/30/2008

Status: Active

Template File: 2730cva.pdf
Upload
Download
Preview

Map File:
Generate
Upload
Download

Return to Search
Previous in List
Next in List
Add
Update/Display
Include History
Correct History

Report Definition Output

Definition **Template** **Output** **Security**

Report Name: SL_CLS_FORM

General

☐ PDF report output may be edited

Runtime Output Format Options

Format Type	Enabled	Default
PDF	☒	☒

Output Location

*Location: Any

[Return to Search](#) [Previous in List](#) [Next in List](#) [Add](#) [Update/Display](#) [Include History](#) [Correct History](#)

[Save](#)

Report Definition Security

[Definition](#)
[Template](#)
[Output](#)
[Security](#)

Report Name: SL_CLS_FORM

Report Viewers

☒ Allow viewer ID assignment at report runtime

Report Viewers			View All	First	1 of 1	Last
ID Type	Distribution ID	Description				
Role	SL_EMPLOYEE	[WF] Employee				

[Return to Search](#)
[Previous in List](#)
[Next in List](#)
[Add](#)
[Update/Display](#)
[Include History](#)
[Correct History](#)

[Save](#)



- ▷ My Favorites
- ▷ HR Self-Service
- ▷ Manager Self Service
- ▷ Enterprise Person
- ▷ Sandia Directory
- ▷ Recruiting
- ▷ Workforce Administration
- ▷ Benefits
- ▷ Compensation
- ▷ Time and Labor
- ▷ Payroll for North America
- ▷ Workforce Development
- ▷ Organizational Development
- ▷ Enterprise Learning
- ▷ Workforce Monitoring
- ▷ Pension
- ▷ Templates
- ▷ Set Up HRMS
- ▷ Enterprise Components
- ▷ Worklist
- ▷ Application Diagnostics
- ▷ Tree Manager
- ▽ Reporting Tools
 - ▷ Query
 - ▷ PS/nVision
- ▽ XML Publisher
 - ▷ Setup
 - [Data Source](#)
 - [Report Definition](#)
 - [Content Library](#)
 - [Translations](#)
 - [Query Report Viewer](#)
 - [Query Report Scheduler](#)
 - [XMLP Report Search](#)
 - [Report Manager](#)
- ▷ PeopleTools
- ▷ Data Models
 - [Careers](#)
 - [Change My Password](#)
 - [My Personalizations](#)
 - [My System Profile](#)
 - [My Dictionary](#)

ID:	
Start Date:	31
End Date:	31
Location of Visit:	
Purpose of Visit:	
Meeting Access:	
SIGMA:	
Site:	
Current Date:	31
Citizen Country:	
Address:	
Phone:	
Affiliation:	
HOST_SNL_ID:	
T_Badge_Yes:	☐
T_BADGE_NO:	☐
Comments:	
CO_HOST_EMPLID:	
CO_HOST_PHONE:	
CO_HOST_FAX:	
CO_HOST_NAME:	
MGR_SL_SNLID:	
MGR_NAME:	
MGR_PHONE:	

Badge Request Instructions

PeopleCode to Generate Report

```
&sRptDefn = "SL_CLS_FORM";  
&sTemplateId = "SL CLS FORM 1";  
/* detect system directory separator */  
&sDirSep = GetDirSeparator();  
  
/* create process directory */  
CreateDirectory("XMLP", %FilePath_Relative);  
  
/* get report definition object */  
&oRptDefn = create PSXP_RPTDEFNMANAGER:ReportDefn(&sRptDefn);  
&oRptDefn.Get();  
  
/* output directory */  
&RptOutputDir = GetEnv("PS_SERVDIR") | &sDirSep | "files" | &sDirSep | "XMLP" | &sDirSep | UuidGen();  
&sOutputDir = &RptOutputDir | &sDirSep | "RptInst";  
&sDataDir = &RptOutputDir | &sDirSep | "Data";  
CreateDirectory(&sOutputDir, %FilePath_Absolute);  
CreateDirectory(&sDataDir, %FilePath_Absolute);  
  
&oRptDefn.OutDestination = &RptOutputDir;  
/* MessageBox(0, "", 0, 0, " report output dir " | &RptOutputDir);*/  
/* Output format */  
&sOutPutFormat = "PDF";  
  
/*fill query runtime prompt record */  
&rcdQryPrompts = &oRptDefn.GetPSQueryPromptRecord();  
&rcdQryPrompts.GetField(Field.EMPLID).Value = &emplid;
```

PeopleCode Cont.

```
/*generate report*/
&oRptDefn.ProcessReport(&sTemplateId, %Language_User, &AsOfDate, &sOutPutFormat);
&sFileext = GetFileExtension(&sOutPutFormat);

/*publish report */

CommitWork();

&sRptDir = &RptOutputDir;

&sReportFile = &sRptDefn | "." | &sOutPutFormat;
&sRptFilePath = &sRptDir | &sDirSep | "RptInst" | &sDirSep | &sReportFile;

&oRptDefn.DisplayOutput();

/*&ret = SendMail(0, "nygarci@sandia.gov", "nygarci@sandia.gov", "nygarci@sandia.gov", "TEST", "TEST", &sRptFilePath, "TEST", "", "",
"TEST", "", "");*/

/* cleanup */
DeleteLocalFile(&sOutputfile, %FilePath_Absolute);
```

Defining PDF Fields

Additional information: COMMENTS		Visit Key # _____ (For Badge Office Use Only)	
Responsible Manager	MGR_SL_SNLID SSN or Employee ID	MGR_NAME Name	Phone No. MGR_PHONE
SNL Host	HOST_SL_SNLID 1. SSN or Employee ID	HOST_NAME Name <i>(Must be Q or L cleared)</i>	Phone No. HOST_PHONE 2.
	SIGNATURE Chief Protection Officer Signature	Org. HOST_ORG 4.	Fax No. HOST_FAX 5.

VISIT DATES: Begin: 2008-02-20 End: 2008-02-20

BADGE DATES: Begin: 2008-02-20 End: 2008-02-20

PROGRAM/MEETING NAME/PURPOSE: LOCATION sample data JUSTIFY sample data

MEETING ACCESS LEVEL: _____
(e.g., SRD/CNWDI, SRD/NO CNWDI, SNSI, AREA, etc.)

SIGMAs (IF APPLICABLE): _____
(i.e., 1, 2, 3)

HOST NAME: SL_NAME_FORMAL sampl CO-HOST NAME: _____
ORG./MAIL STOP: SL M TELEPHONE: _____
TELEPHONE NO.: SL WORK PHONE sample FAX NO.: _____
FAX NO.: SL FAX sample data

VISITOR LIST

Social Security No.	Full Name (Last, First, Middle Initial)	Company Name
1. <u>SSN sample</u>	<u>SL_NAME_FORMAL sample</u>	<u>SL_AFFILIATION sample data</u>

Helpful PeopleSoft Websites

- Jim's PeopleSoft Journal: <http://jjmpsj.blogspot.com/>
- The PeopleSoft DBA Blog: <http://blog.psftdba.com/>
- PeopleSoft Support and Tips:
<http://peoplesoftexperts.blogspot.com/>
- Driven by PeopleSoft motivated by Microsoft:
<http://psguyblog.blogspot.com/index.html>
- Manalang: <http://manalang.wordpress.com/2006/02/01/adding-live-search-to-peoplesoft-enterprise/>
- Grey Sparling: <http://blog.greysparling.com/>

Email from Application Engine

Local string &mail_text1, &to_emailid, &cc_name_list
, &email_subject, &mail_text2, &Mail_text3, &dbname;

&dbname = %DbName;

&RET = Char(13);

&mail_text2 = TC_JCEARN_AET.HR_NP_NOTE_TEXT;

&Mail_text3 = TC_JCEARN_AET.TC_PEN_NOTE_TEXT;

If &dbname = "Production" Then

 &to_emailid = "wg-pro@your_company.com";

Else

 &to_emailid = "wg-test@your_company.com";

End-If;

&email_subject = &dbname | " " | "Sample";

&mail_text1 = "This email is a system generated email, please do not reply to this email." | &RET;

&mail_text1 = &mail_text1 | &RET | "The following are ...: " | &RET;

&mail_text1 = &mail_text1 | &RET | "Header1: ";

&mail_text1 = &mail_text1 | &RET | "If the ..." | &RET | &mail_text2 | &RET;

&mail_text1 = &mail_text1 | &RET | "Header2: ";

&mail_text1 = &mail_text1 | &RET | "If the ..." | &RET | &Mail_text3 | &RET;

SendMail(0, &to_emailid, &cc_name_list, "", &email_subject, &mail_text1);

Email Added to Open Enrollment

- Added an Email
- Provides Confirmation of Choices with Cost
- Workflow Event not called
- Work around
- Survey

Open Enrollment Email Continued

- Two Submit Buttons

Submit

Click **Submit** to continue. You have one more step!



Important: Your enrollment will not be complete until you

- 1) click the **Submit** button on this page *and***
- 2) click the **Submit** button on the next page to authorize deductions**

Benefits Enrollment

Submit Benefit Choices

Judi Doolittle

You have almost completed your enrollment. If you have no further changes, click **Submit** at the bottom of this page to finalize your benefit choices.

You may store your choices on each page and return to the Enrollment Summary as many times as you like until the enrollment deadline. After the deadline, you will not be able to make any further benefit changes until the next Open Enrollment period or until you have a qualifying mid-year change event.

Authorize Election Deductions

By submitting your benefit choices you are authorizing the company to take deductions from your paycheck to pay for your benefit costs. You are also authorizing the Benefits Department to send necessary personal information to your selected providers to initiate and support your coverage.

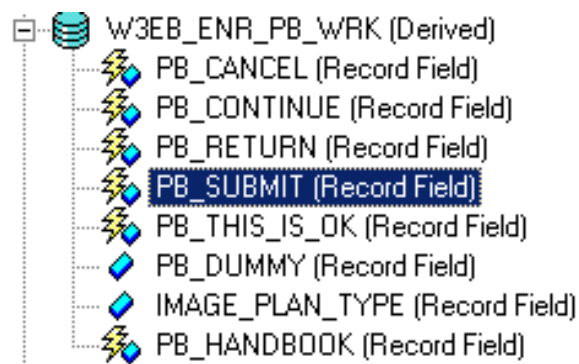
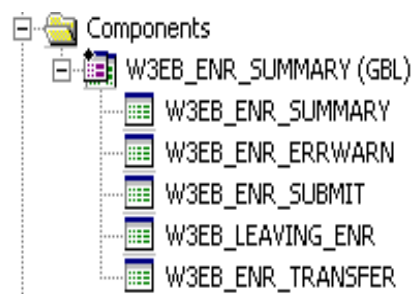
Submit

Click **Submit** to send your choices to the Benefits Department.

Cancel

Click **Cancel** if you are not ready to submit your choices and wish to return to the Enrollment Summary.

Where to place the Email?



Code

```

PB_SUBMIT (field)
FieldChange

import PT_MCF_MAIL:*;

Declare Function StartSubmit PeopleCode FUNCLIB_W3EBENR.SUMMARY FieldFormula;

Declare Function Get_Person_Name PeopleCode FUNCLIB_NAME.PERSON_NAME FieldFormula;

local Rowset <FILEROWSET, <RS_W3EB_ENR_SM_WRK, <level0, <level1;
local string <fromaddress, <to1list, <cc1list, <bc1list, <subject, <emplid, <name, <dbname, <ret, <indt, <stab;
local number <i1;
local string <plan_type_descr, <new_covrg, <before_tax, <after_tax, <event_class, <nbsp, <as_of;
local string <html_table, <html_begin, <html_text, <html_end, <email_id, <bounce;
local PT_MCF_MAIL:MCFOutboundEmail <email = create PT_MCF_MAIL:MCFOutboundEmail();

rem MessageBox(0, "", 0, 0, " %1", "before if");

If %Page = Page.W3EB_ENR_SUBMIT Then

    rem WinMessage("Inside IF " | %Page, 0);
    /*JAD */
    <html_table = "<table border = '2'>";
    <html_text = "<HTML> <Body>";
    <nbsp = " ";
    rem W3EB_ENR_SM_WRK.EVT_CLS_DESCR.Value;
    rem WinMessage("Event Class:" | W3EB_ENR_SM_WRK.EVT_CLS_DESCR.Value, 0);

    /*<level0 = GetLevel0(); don't need because we are on the component record--parent */
    <FILEROWSET = GetRowset(Scroll.W3EB_ENR_SUM_VW);
    <html_begin = <html_text | "<A title=http://www-irrn.sandia.gov/hr/benefits.htm href='http://www-irrn.sandia.gov/hr/benefits.htm'>";
    <html_table = <html_table | "<tr bgcolor=#99ccff>" | "<th>Benefit Plan</th>" | "<th>Description</th>" | "<th> Before Tax</th>"
    For <i1 = 1 To <FILEROWSET.ActiveRowCount
        <plan_type_descr = <FILEROWSET(<i1).W3EB_ENR_SM_WRK.PLAN_TYPE_DESCR.Value;
        <new_covrg = <FILEROWSET(<i1).W3EB_ENR_SM_WRK.NEW_COVRG.Value;
        <before_tax = "<$" | <FILEROWSET(<i1).W3EB_ENR_SM_WRK.BEFORE_TAX.Value;
        <after_tax = "<$" | <FILEROWSET(<i1).W3EB_ENR_SM_WRK.AFTER_TAX.Value;
        <html_table = <html_table | "<tr>" | "<td><align = 'left'>" | <plan_type_descr | "</td>";
        <html_table = <html_table | "<td><align = 'left'>" | <new_covrg | "</td>";
        <html_table = <html_table | "<td><align = 'right'>" | <before_tax | "<nbsp;" | "</td>";
        <html_table = <html_table | "<td><align = 'right'><colspan = '5'>" | <after_tax | "<nbsp;" | "</td></tr>";
    End-For;
    <html_table = <html_table | "</table>";

    /*<level1 = getlevel1();
    <filerowset2 = getrowset(scroll.w3eb_enr_sum_vw); */
    /*WinMessage("PLAN_TYPE_DESCR " | <FILEROWSET(1).W3EB_ENR_SM_WRK.PLAN_TYPE_DESCR.Value, 0); */

    <emplid = %EmployeeId;
    <name = Get_Person_Name(<emplid, "", "", "");
    <emplid = %EmployeeId;

```

Code Continued

```

PB_SUBMIT (field) FieldChange

&emplid = %EmployeeId;
&name = Get_Person_Name(&emplid, "", "", "");
&emplid = %EmployeeId;

/*-- Create an email object by setting individual parameters ---*/

&dbname = %DbName;
Evaluate &dbname
When = "PSPRI"
    &fromaddress = "hris_pspri@sandia.gov";
    &tolist = %OperatorId | "@sandia.gov";
    &cclist = "ebenefits@sandia.gov";
    &bounce = "ebenefits@sandia.gov";
    &subject = "Benefits Elections for " | &name | " (ID:" | &emplid | ")";
    Break;
When-Other
    /* enter test operids to prevent email going to user*/
    If %OperatorId = "dbaker" Or
        %OperatorId = "dyzquia" Or
        %OperatorId = "labunti" Or
        %OperatorId = "isanche" Or
        %OperatorId = "wafaust" Or
        %OperatorId = "jjcarna" Or
        %OperatorId = "mlabate" Or
        %OperatorId = "jabusta" Or
        %OperatorId = "jlhill" Or
        %OperatorId = "jaashba" Or
        %OperatorId = "rahudso" Or
        %OperatorId = "deotts" Or
        %OperatorId = "mlcurry" Then
        &tolist = "pkeega@sandia.gov";
    Else
        &tolist = %OperatorId | "@sandia.gov";
    End-If;
    &fromaddress = "hris_psdv2@sandia.gov";
    &cclist = %OperatorId | "@sandia.gov";
    &cclist = "ebenefits@sandia.gov";
    &bounce = "ebenefits@sandia.gov";
    &subject = "Benefits Elections for " | &name | " (ID:" | &emplid | ") | " TESTING: " | &dbname;
    Break;
End-Evaluate;
/*
    When = "PSQAL"
    &fromaddress = "hris_psqal@sandia.gov";
    &tolist = %OperatorId | "@sandia.gov";
    &cclist = "ebenefits@sandia.gov";
    &bounce = "ebenefits@sandia.gov";
*/

```

Code Continued

```

PB_SUBMIT (field)
FieldChange

<subject = "Benefits Elections for " | &name | " (ID:" | &emplid | ")" | " TESTING: " | &dbname;
Break;*/

<email.From = &fromaddress;
<email.Recipients = &tolist;
<email.BCC = &ccolist;
<email.Subject = &subject;
<email.BounceTo = &bounce;
<as_of = DateTimeToLocalizedString(%Datetime, "M/d/yyyy h:mm a");

If W3EB_ENR_SM_WRK.EVT_CLS_DESCR.Value = "Hire / ReHire Self-Service" Then
    <html_text = <html_begin | &name | "<br><p>" | "Your New Hire elections (below) have been saved as of " | &as_of | ".";
    <html_text = <html_text | "<p>" | "<p>" | " The dollar amounts shown will be your pay-check deductions.";
    <html_end = <html_end | "<p>" | "<p>" | " Your elections will be finalized overnight. If you need to modify your elections after t
    <html_end = <html_end | "<p>" | "<p>" | "For interactive self-help customer service on the web, click on: <A title=http://
    <html_end = <html_end | "<p>" | " PLEASE DO NOT REPLY - This message was sent from an unattended mailbox.";
    <html_end = <html_end | "</HTML> </Body>";
Else
    <html_text = <html_begin | &name | "<br><p>" | "Your Open Enrollment elections (below) have been saved as of " | &as_of |
    /* <html_text = <html_text | "<br><p>" | " Open enrollment Elections: Employees have until #date_variable at 5:00 pm(
    <html_text = <html_text | "<p>" | "<p>" | " The dollar amounts shown will be your pay-check deductions.";

    <html_end = <html_end | "<p>" | "You may change your elections in ";
    <html_end = <html_end | "<A href='https://ws3lsmint-new.sandia.gov/psp/psprlux/EMPLOYEE/HRMS/c/W3EB_MENU.W3EB_ENR_SELECT.GE
    <html_end = <html_end | " until 5:00 PM MST on the last day of Open Enrollment. ";
    <html_end = <html_end | "<p>" | "<p>" | "For interactive self-help customer service on the web, click on: <A title=http://
    <html_end = <html_end | "<p>" | "Your feedback is important to us! Please complete this short <A href='https://sareprod.s
    <html_end = <html_end | "<p>" | " PLEASE DO NOT REPLY - This message was sent from an unattended mailbox.";
    <html_end = <html_end | "</HTML> </Body>";
End-If;
<email.Text = <html_text | <html_table | <html_end;
<email.ContentType = "text/html";

End-If;

If %Page = Page.W3EB_ENR_SUMMARY Then
    StartSubmit();
Else
    DoSaveNow();
    Local integer <resp = <email.Send();
    Local boolean <done;

    Evaluate <resp
    When %ObEmail_Delivered
    /* every thing ok */

```

Code Continued

```

PB_SUBMIT (field) FieldChange

End-If;

If %Page = Page.W3EB_ENR_SUMMARY Then
    StartSubmit();
Else
    DoSaveNow();
    Local integer <resp = <email.Send();
    Local boolean <done;

    Evaluate <resp
    When %ObEmail_Delivered
        /* every thing ok */
        <done = True;
        Break;

    When %ObEmail_NotDelivered
        /*-- Check <email.InvalidAddresses, <email.ValidSentAddresses
        and <email.ValidUnsentAddresses */
        <done = False;
        Break;

    When %ObEmail_PartiallyDelivered
        /* Check <email.InvalidAddresses, <email.ValidSentAddresses
        and <email.ValidUnsentAddresses; */
        <done = True;
        Break;

    When %ObEmail_FailedBeforeSending
        /* Get the Message Set Number, message number;
        Or just get the formatted messages from <email.ErrorDescription,
        <email.ErrorDetails;*/
        <done = False;
        Break;

    End-Evaluate;
    /* SNL BEN SCR845 07/14/2006 DGB */
    /*-----
    * During testing PS had trouble going to the next screen
    * the following Forces the transfer here.
    *-----*/

    Transfer( False, MenuName.W3EB_MENU, BarName.ENROLLMENT, ItemName.SELECT, Panel.W3EB_ENR_SELECT, "L");
    /*END SCR845 */
End-If;

```



D.Duck:

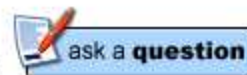
Your Open Enrollment elections (below) have been saved as of 10/20/2008 7:42 AM.

The dollar amounts shown will be your pay-check deductions.

Benefit Plan	Description	Before Tax	After Tax
Medical	UHC Standard PPO:Empl+Child	\$11.50	\$
Dental	Dental Care Plan:Empl + One	\$7.50	\$
Vision	Vision Care Plan:Empl + One	\$0.00	\$
Vol Grp Accident Plan 1A	Voluntary Group Accident 1A: \$125,000	\$	\$1.56
Vol Grp Accident Plan 1B	Voluntary Group Accident 1B: \$100,000	\$	\$1.98
Vol Grp Accident Plan 2	Voluntary Group Accident 2: \$75,000	\$	\$0.17
Long-Term Disability	Long-term Disability + 20%: 70.00% of Salary	\$	\$10.65
FSA Health Care	FSA - Health Care: \$5,000.00	\$192.31	\$
FSA Day Care	Waive	\$0.00	\$
Vacation Buy	Vacation Buy: 44.00 Hours	\$49.67	\$

You may change your elections in [Benefits Enrollment](#) until 5:00 PM MST on the last day of Open Enrollment.

For interactive self-help customer service on the web, click on:



Your feedback is important to us! Please complete this short [Survey](#)

PLEASE DO NOT REPLY - This message was sent from an unattended mailbox.

Embedding PL/SQL in your Solutions

- Packages
 - See the code on the following four slides—for custom PS Package to log who, and errors for PL/SQL programs in PeopleSoft

```
CREATE OR REPLACE PACKAGE ps_utl_pkg
AS
PROCEDURE log_new_line;
PROCEDURE intialize_log ( p_pgm_name IN VARCHAR2, p_log_file_output IN BOOLEAN);
PROCEDURE write_log (p_log_entry IN VARCHAR2);
PROCEDURE write_error (p_error_entry IN VARCHAR2);
END ps_utl_pkg;
/
CREATE OR REPLACE PACKAGE BODY ps_utl_pkg
AS
/*****
NAME:    ps_utl_pkg
*****/
gv_output_type  VARCHAR2 (25)          := 'LOG';
gv_pgm_name     sl_plsql_log.programname%TYPE;
PROCEDURE log_new_line
AS
BEGIN
    write_log ( ' ');
END;
```

```
PROCEDURE intialize_log (p_pgm_name          IN  VARCHAR2, p_log_file_output
                        IN  BOOLEAN)
AS
BEGIN
gv_pgm_name := p_pgm_name;
  IF p_log_file_output
  THEN
    gv_output_type := 'LOG';
  ELSE
    DBMS_OUTPUT.ENABLE (100000);
    gv_output_type := ' ';
  END IF;
log_new_line;
  write_log ( 'Log started for '
            || gv_pgm_name
            || ' at '
            || TO_CHAR (SYSDATE, 'YYYY-MM-DD HH24:MI:SS')
            );
log_new_line;
END;
```

```
PROCEDURE close_log ( p_pgm_name IN  VARCHAR2 )
AS
BEGIN
    log_new_line;
    write_log ( 'End of log for ' || gv_pgm_name || ' at ' || TO_CHAR (SYSDATE,
        'YYYY-MM-DD HH24:MI:SS'));
    log_new_line;
END;

PROCEDURE write_log (p_log_entry IN VARCHAR2)
AS
BEGIN
IF gv_output_type = 'LOG'
    THEN
        INSERT INTO sl_plsql_msg_temp
            VALUES (SYSDATE, p_log_entry);
    ELSE
        DBMS_OUTPUT.put_line (p_log_entry);
    END IF;
END;
```

```
PROCEDURE write_error (p_error_entry IN VARCHAR2)
AS
  PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
  IF gv_output_type = 'LOG'
  THEN
    INSERT INTO sl_plsql_msg_temp
      VALUES (SYSDATE, p_error_entry);
    COMMIT;
  ELSE
    DBMS_OUTPUT.put_line (p_error_entry);
  END IF;
END;
END ps_util_pkg;
/
```

Executing PL/SQL

- Invoke PL/SQL anonymous blocks straight from SQL*PLUS
- Run PL/SQL anonymous blocks by embedding the inside other programs
- Stored procedure or function you can call it in the following ways:
 - SQL*PLUS or iSQL*Plus
 - Oracle Development tools (Oracle Forms Developer)
 - Another procedure

Executing PL/SQL Example

- *Anonymous Block*
@scriptname.sql
- *From SQL*Plus*
EXECUTE execution_one
- *Stored Procedure*
execution_one;
- *From Another Procedure*
CREATE OR REPLACE PROCEDURE emp_leave
(p_id IN emp.emp_id%TYPE)
IS
BEGIN
DELETE FROM emp
WHERE emp_id = p_id;
log_execution;
END
leave_emp;

SQR

- See code on the following three slides
- Note: double ;; is required to embed in PL/SQL in SQR

Example embedded anonymous block::

```
BEGIN-SQL on-error=Db_Error
```

```
DECLARE
```

```
V_ERR_CDE PS_UU_RPT_LDGR_ERR.UU_ERR_CODE%TYPE;;
```

```
V_ERR_MSG PS_UU_RPT_LDGR_ERR.UU_ERR_MSG%TYPE;;
```

```
V_BLOCK_NAME
```

```
PS_UU_RPT_LDGR_ERR.UU_BLOCK_NAME%TYPE;;
```

```
BEGIN
```

```
DBMS_STATS.GATHER_TABLE_STATS('FSPROD','PS_UU_RPT_LE  
DGER', NULL, NULL, FALSE, 'FOR ALL COLUMNS SIZE 1', NULL,  
'DEFAULT', TRUE, NULL, NULL, NULL);;
```

```
EXCEPTION
```

```
  WHEN OTHERS THEN
```

```
    V_ERR_MSG := SUBSTR(SQLERRM,1,100);;
```

```
    V_ERR_CDE := SQLCODE;;
```

```
    V_BLOCK_NAME := 'ANALYZE STATS ';;
```

```
    INSERT INTO PS_UU_RPT_LDGR_ERR(UU_BLOCK_NAME
                                   ,UU_ERR_CODE
                                   ,UU_ERR_MSG
                                   ,UU_DATE)
```

```
    VALUES (V_BLOCK_NAME
             ,V_ERR_CDE
             ,V_ERR_MSG
             ,SYSDATE);;
```

```
END;;
```

```
END-SQL
```

Example calling a stored procedure:

BEGIN-SQL

BEGIN EXTCOPY.GW_DOWN_GET_STATUS_W_NUMS

 (\$project-num,
 \$task-num,
 \$v_ei_date,
 \$v_expend_type,
 \$v_non_labor_resource,
 #v_emplid,
 #v_quantity,
 \$status,
 \$status_desc);;

END;;

END-SQL

SQR Template

- See code on the following nine slides

```
!*****
```

```
#include 'setenv.sqc' !Set environment
```

```
!*****
```

```
!create a temporary table. if exists, then skip
```

```
!*****
```

```
begin-procedure intialize-plsql-output
```

```
! create a temporary table to be used pl/sql for logging
```

```
begin-sql on-error=skip-error
```

```
create global temporary table sysadm.sl_plsql_msg_temp (sl_orig_timestamp date,  
message_descr long varchar)
```

```
on commit preserve rows
```

```
end-sql
```

```
! do grants on temp table
```

```
begin-sql on-error=skip-error
```

```
grant select, insert on sl_plsql_msg_temp to psuser, psdeveloper, workflow, payroll
```

```
end-sql
```

```
end-procedure intialize-plsql-output
```

```
!*****
```

```
!*****
```

```
!SQL to extract and display output
```

```
!*****
```

```
begin-procedure display-output
```

```
  let $sql-statement = 'select sl_plsql_msg_temp'
```

```
begin-select on-error=sql-error
```

```
a.message_descr
```

```
  print &a.message_descr (+1,3,100)
```

```
from sysadm.sl_plsql_msg_temp a
```

```
order by sl_orig_timestamp
```

```
end-select
```

```
end-procedure display-output
```

```
!*****
```

```
!error processing for pl/sql errors
```

```
!*****
```

```
begin-procedure plsql-error
```

```
  display 'PL/SQL Error'
```

```
  display ' '
```

```
  display 'SQL Status = '      noline
```

```
  display #sql-status 99999    noline
```

```
  display ', SQL Error = '     noline
```

```
  display $sql-error
```

```
  display ' '
```

```
  display ' - SQL Statement = ' noline
```

```
  display $sql-statement
```

```
  display ' '
```

```
  ! print log
```

```
  do display-output
```

```
  stop
```

```
end-procedure plsql-error
```

```
!*****
```

```
!procedure to skip sql errors
```

```
!*****
```

```
begin-procedure skip-error
```

```
!See Paper for Full Code
```

```
!*****
```

```
! Program Section
```

```
!*****
```

```
begin-program
```

```
do Init-Report
```

```
display 'DEBUG: calling intialize-plsql-output'
```

```
do intialize-plsql-output
```

```
do Process-Main
```

```
do End-Processing
```

```
end-program
```

```
!*****
```

```
! Procedure Init-Report
```

```
!*****
```

```
!Standard Init-Report See Paper for Full Code
```

```
begin-procedure Init-Report
```

```
end-procedure Init-Report
```

```
!*****
! Section Heading
!*****
!Standard Heading
!*****
! Process-Main
!*****
!Standard Process-Main
!*****
! Invoke stored procedure created by DBA's.
!*****
begin-procedure process-ld-pa
  let $sql-statement = 'Proc: PAY_FILE_AUDIT_PKG.AUDIT_LD_PA;'
BEGIN-SQL ON-ERROR=plsql-error
BEGIN
  SYSADM.PAY_FILE_AUDIT_PKG.AUDIT_LD_PA;;
END;;
END-SQL
end-procedure process-ld-pa
```

```
!*****
! display log output
!*****
begin-procedure select-log
  do display-output
end-procedure select-log
!*****
begin-procedure END-PROCESSING
!*****
  do reset
  do get-current-datetime
  display 'End of Run'
  display $AsOfNow
  do Stdapi-Term
  display 'Successful Completion'
end-procedure END-PROCESSING
!*****
!Included SQCs
!*****
!Standard includes only exception is the PLSQL.SQC see paper for full code
#include 'plsql.sqc'
```

SQC Example

- See code on the following four slides

```
!*****
```

```
! Name:      PLSQL.SQC
```

```
!*****
```

```
#include 'setenv.sqc' !Set environment
```

```
!*****
```

```
!create a temporary table. if exists, then skip
```

```
!*****
```

```
begin-procedure intialize-plsql-output
```

```
! create a temporary table to be used pl/sql for logging
```

```
begin-sql on-error=skip-error
```

```
create global temporary table sysadm.sl_plsql_msg_temp (sl_orig_timestamp date,  
  message_descr long varchar)
```

```
  on commit preserve rows
```

```
end-sql
```

```
! do grants on temp table
```

```
begin-sql on-error=skip-error
```

```
grant select, insert on sl_plsql_msg_temp to psuser, psdeveloper, workflow, payroll
end-sql
end-procedure intialize-plsql-output
!*****
!SQL to extract and display output
!*****
begin-procedure display-output
    let $sql-statement = 'select sl_plsql_msg_temp'
begin-select on-error=sql-error
a.message_descr
    print &a.message_descr (+1,3,100)
from sysadm.sl_plsql_msg_temp a
order by sl_orig_timestamp
end-select
end-procedure display-output
```

```
!*****
```

```
!error processing for pl/sql errors
```

```
!*****
```

```
begin-procedure plsql-error
```

```
  display 'PL/SQL Error'
```

```
  display ' '
```

```
  display 'SQL Status = '      noline
```

```
  display '#sql-status 99999'  noline
```

```
  display ', SQL Error = '     noline
```

```
display $sql-error
```

```
display ' '
```

```
display ' - SQL Statement = ' noline
```

```
display $sql-statement
```

```
display ' '
```

```
! print log
```

```
do display-output
```

```
stop
```

```
end-procedure plsql-error
```

```
!*****
```

```
!procedure to skip sql errors
```

```
!*****
```

```
begin-procedure skip-error
```

```
! skip error and do nothing
```

```
end-procedure skip-error
```

PL/SQL in PeopleSoft Upgrades

- See Code for Creation of History Table

History Table

History table Creation

```
CREATE TABLE SOURCE_HIST AS SELECT SYSDATE CHANGE_DATE,  
    USER_SOURCE.* FROM USER_SOURCE WHERE 1=2;
```

Trigger to Store and Maintain History

```
CREATE OR REPLACE TRIGGER CHANGE_HIST AFTER CREATE ON SCHEMA BEGIN  
    IF DICTIONARY_OBJ_TYPE in ('PROCEDURE', 'FUNCTION', 'PACKAGE',  
    'PACKAGE BODY', 'TYPE')  
    THEN  
    INSERT INTO SOURCE_HIST SELECT sysdate, user_source.* FROM USER_SOURCE  
  
    WHERE TYPE = DICTIONARY_OBJ_TYPE  
    AND NAME = DICTIONARY_OBJ_NAME;  
    END IF;  
    EXCEPTION WHEN OTHERS THEN RAISE_APPLICATION_ERROR(-20000,  
    SQLERRM); END; Metalink Note:258690.1
```

UTIL_FILE Example

DECLARE

```
v_file utl_file.file_type;  
v_desc varchar2(4000);  
v_host varchar2(40);
```

```
cursor c1 is select  
name,  
text  
from user_source  
order by line;  
r1 c1%rowtype;
```

```
BEGIN  
select host_name  
into v_host  
from v$instance;
```

```
v_file := utl_file.fopen('c:\oracle\admin\orcl\utl_file', 'plsql_exp.utl', 'W');
utl_file.put_line(v_file, '/*');
utl_file.put_line(v_file, ' +-----+');
utl_file.put_line(v_file, ' UPDATED: ' || to_char(sysdate, 'MM-DD-YYYY
HH12:MIam'));
utl_file.put_line(v_file, ' SCHEMA: ' || lower(user));
utl_file.put_line(v_file, ' DB SERVER: ' || lower(v_host));
utl_file.put_line(v_file, ' +-----+');
utl_file.put_line(v_file, '*/');
```

```
open c1;
  loop
    fetch c1 into r1;
    exit when c1%notfound;
    utl_file.new_line(v_file,2);
    v_desc := rtrim(rtrim(r1.name,chr(10) ));
    utl_file.put_line(v_file, 'CREATE OR REPLACE PROCEDURE ' || v_desc);
    utl_file.put_line(v_file, r1.text);
    utl_file.put_line(v_file, '/');
  end loop;
close c1;
utl_file.fclose(v_file);
END;
/
```

How to find bad subpages

---Sql to find the subpages on the page

```
SELECT *  
FROM PSPNLDEFN  
WHERE PNLNAME LIKE 'TC%'  
AND EXISTS (SELECT 'x' FROM PSPNLFIELD WHERE  
             SUBPNLNAME = :SUBPNLNAME)
```

--- example 'PERS_SRCH_SBP'

---fields on page

```
SELECT * FROM PSPNLFIELD  
WHERE SUBPNLNAME = :SUBPNLNAME
```

Hidden folders and content references

If you do not see a folder or content Reference (Menu Item) in Left hand side navigation, (Applicable to 8.4x) then you must check to see if the folder or content reference is not marked as hidden. Other reasons could be security. Here is a SQL to find out all the objects that are hidden.

To find all the folders which are hidden from Portal Navigation.

```
SELECT *  
from PSPRSMSYSATTRVL  
where portal_name = 'EMPLOYEE'  
and PORTAL_ATTR_NAM =  
    'PORTAL_HIDE_FROM_NAV' and portal_Reftype = 'F'
```

Hidden Folders Continued

```
SELECT
a.portal_objname,b
.portal_label
,b.portal_prntobjname
,c.portal_label
  from PSPRSMSYSATTRVL a,psprsmdefn b,psprsmdefn c
where a.portal_name = 'EMPLOYEE'
  and a.PORTAL_ATTR_NAM = 'PORTAL_HIDE_FROM_NAV'
  and b.portal_Reftype = 'F' and a.portal_name = b.portal_name
  and a.portal_objname = b.portal_objname
  and b.portal_name = c.portal_name
  and b.portal_prntobjname = c.portal_objname
```

Hidden Folders Continued

To find all the content references which are hidden from Portal Navigation.

```
SELECT *  
from PSPRSMSYSATTRVL  
where portal_name = 'EMPLOYEE'  
and PORTAL_ATTR_NAM =  
    'PORTAL_HIDE_FROM_NAV' and portal_Reftype = 'C'
```

Hidden Folders Continued

```
SELECT
a.portal_objname
,b.portal_label
,b.portal_prntobjname
,c.portal_label
  FROM PSPRSMSYSATTRVL a,psprsmdefn b,psprsmdefn c
 WHERE a.portal_name = 'EMPLOYEE'
       AND a.PORTAL_ATTR_NAM = 'PORTAL_HIDE_FROM_NAV'
       AND b.portal_Reftype = 'C'
 AND a.portal_name = b.portal_name
       AND a.portal_objname = b.portal_objname
       AND b.portal_name = c.portal_name
       AND b.portal_prntobjname = c.portal_objname
```

Hidden Folders Continued

Replace EMPLOYEE with Your portal name.

Some of the Other names are:

CUSTOMER

DEMOSITE

EMPLOYEE

MOBILE

PORTAL

PS_SITETEMPLATE

SUPPLIER

Portal Content Reference/folder Attributes are stored in:

PSPRSMSYSATTR

PSPRSMSYSATTRVL

Where is the email value stored?

- *Remember if you are working with emails, workflow, and Oprid tables there are two tables the email is stored*
- *Most times it uses PSOPRDEFN but sometimes it uses PS_ROLEXLATOPR*
- *What is in PSOPRDEFN should reflect in PS_ROLEXLATOPR*
- *So if you write a script to change one you need to write a script to change the other*

```
SELECT *  
FROM PSOPRDEFN  
WHERE OPRID = 'smith'
```

```
SELECT *  
FROM PS_ROLEXLATOPR  
WHERE OPRID = 'smith'
```

Email Stored Continued

- *There is one more table that needs to change as well*
- *Which is the email that was triggering for workflow for the manager in Job Opening*

```
SELECT *  
FROM PSUSEREMAIL  
WHERE OPRID = 'smith'
```

Find a process scheduler for a process

- *How to find if a process has a process Scheduler*
 - *What page it has attached , and type*
 - *Who runs the process even people who are gone*
 - *Who ran what process*
- there a process scheduler for a SQR or process app engine*
--also descr plus Page

```
SELECT A.PRCSTYPE, A.PRCNAME, PNLGRPNAME,  
       DESCRLONG  
FROM PS_PRCSEFN A, PS_PRCSEFNPNL B  
WHERE A.PRCSTYPE = B.PRCSTYPE  
AND A.PRCNAME = B.PRCNAME  
AND B.PRCNAME = 'SHRS104U'
```

Process Scheduler Continued

--what user ran what process

```
SELECT *  
FROM PS_PMN_PRCSLIST  
WHERE RUNCNTLID = 'jsmith'
```

```
SELECT *  
FROM PS_PMN_PRCSLIST  
WHERE RUNCNTLID = :RUNCNTLID
```

Process Scheduler Continued

--who ran this process

```
SELECT *  
FROM PS_PMN_PRCSLIST  
WHERE PRCSNAME = 'SHRS104U'
```

```
SELECT *  
FROM PS_PMN_PRCSLIST  
WHERE PRCSNAME = :PRCSNAME
```

Process Scheduler Continued

--who and process

```
SELECT *  
FROM PS_PMN_PRCSLIST  
WHERE RUNCNTLID = :RUNCNTLID  
AND PRCSNAME = :PRCSNAME
```

```
SELECT *  
FROM PS_PMN_PRCSLIST  
WHERE RUNCNTLID = :RUNCNTLID  
AND PRCSNAME = :PRCSNAME
```

How to find a Message Catalog by Date

- *Message Catalog are stored in a new record PSMSGCATDEFN the old record was PS_MESSAGE_CATALOG*

--8.3 old should not exist

```
SELECT *  
FROM PS_MESSAGE_CATALOG
```

--8.9 exist

```
SELECT *  
FROM PSMSGCATDEFN  
WHERE MESSAGE_SET_NBR = '20302'
```

Message Catalog by Date Continued

--This SQL will help you determine what message catalogs have been changed in 8.9 or added.

```
SELECT *  
FROM PSMSGCATDEFN  
WHERE TO_DATE(SUBSTR(LAST_UPDATE_DTTM,1  
    ,10)) '01-Apr-06'
```

Message Catalog by Date Continued

*-- for ones in your message catalog range for eRecruit in house we use
20301 to 20310*

```
SELECT *  
FROM PSMSGCATDEFN  
WHERE TO_DATE(SUBSTR(LAST_UPDATE_DTTM,1  
    ,10)) '01-Apr-06'  
AND (MESSAGE_SET_NBR > 20300  
AND MESSAGE_SET_NBR < 20304)
```

Message Catalog by Date Continued

*--metadata record for project Items Item OBJECTTYPE = 25 is for
message catalogs*

--if you know your project name and know the amount above the values

```
SELECT *  
FROM PSPROJECTITEM  
WHERE PROJECTNAME = 'TEST'  
AND OBJECTTYPE = 25
```

Message Catalog by Date Continued

-- The 2 sql values should match if I created 6 Message catalogs there should be in your project

```
SELECT COUNT(*)  
FROM PSMSGCATDEFN  
WHERE TO_DATE(SUBSTR(LAST_UPDATE_DTTM,1,10)) '01-  
Apr-06'  
AND (MESSAGE_SET_NBR > 20300  
AND MESSAGE_SET_NBR < 20304)
```

```
SELECT COUNT(*)  
FROM PSPROJECTITEM  
WHERE PROJECTNAME = 'TEST'  
AND OBJECTTYPE = 25
```

Find what Access a Role Gives a User

--the role is TC_EMPLOYEE in this case

--how to tell what role as access to what folder/pages in the portal

---here is the only role that jlstest has and he can or should see all of this

```
SELECT PORTAL_PERMNAME AS
```

```
Permission_list,PORTAL_URI_SEG1 AS Menu ,
```

```
PORTAL_URI_SEG2 AS
```

```
component,'https://www.testco.com/psp/test/EMPLOYEE/HRMS/' ||
```

```
'c/' || PORTAL_URI_SEG1 || '.' || PORTAL_URI_SEG2 ||'.'
```

```
PORTAL_URI_SEG3 AS Address_string
```

Find what access a Role gives a User

```
FROM PSPRSMPerm A, PSPRSMDEFN B WHERE
  A.PORTAL_NAME = B.PORTAL_NAME
AND A.PORTAL_REFTYPE = B.PORTAL_REFTYPE
  AND A.PORTAL_OBJNAME = B.PORTAL_OBJNAME

AND PORTAL_PERMNAME = (SELECT CLASSID
  FROM PSROLECLASS WHERE Rolename =
  'SL_EMPLOYEE' AND A.PORTAL_PERMNAME =
  CLASSID )
```

Find what access a Role gives a User

--sort by menu and no blank menus

```
SELECT PORTAL_PERMNAME AS  
  Permission_list, PORTAL_URI_SEG1 AS Menu ,  
  PORTAL_URI_SEG2 AS  
  component, 'https://www.test.com/psp/test/EMPLOYEE/H  
RMS/' || 'c/' || PORTAL_URI_SEG1 || '.' ||  
  PORTAL_URI_SEG2 || '.' || PORTAL_URI_SEG3 AS  
  Address_string
```

Find what access a Role gives a User

```
FROM PSPRSMPerm A, PSPRSMDEFN B
WHERE A.PORTAL_NAME = B.PORTAL_NAME AND
      A.PORTAL_REFTYPE = B.PORTAL_REFTYPE AND
      A.PORTAL_OBJNAME = B.PORTAL_OBJNAME AND
      PORTAL_PERMNAME =
(SELECT CLASSID
FROM PSROLECLASS
WHERE Rolename = 'SL_EMPLOYEE'
AND A.PORTAL_PERMNAME = CLASSID )
AND PORTAL_URI_SEG1 <> ''
ORDER BY PORTAL_URI_SEG1
```

Find where the Component is in PeopleSoft Portal and Permission List

-- 1 permission and Address String by component

```
SELECT PORTAL_PERMNAME AS
  Permission_list, https://www.test.com/psp/test/EMPLOYEE/HRMS/ ||
  'c/' || PORTAL_URI_SEG1 || '.' || PORTAL_URI_SEG2 || '.' ||
  PORTAL_URI_SEG3
AS Address_string
FROM PSPRSMPerm A, PSPRSMDEFN B
WHERE A.PORTAL_NAME = B.PORTAL_NAME
AND A.PORTAL_REFTYPE = B.PORTAL_REFTYPE      AND
  A.PORTAL_OBJNAME = B.PORTAL_OBJNAME        AND
  B.PORTAL_URI_SEG2 = 'HRS_CE'
```

Find where the Component is in PeopleSoft Portal and Permission List

-- 2 permission, menu, component and Address String by component

```
SELECT PORTAL_PERMNAME AS Permission_list,  
       PORTAL_URI_SEG1 AS Menu , PORTAL_URI_SEG2 AS  
       component, 'https://www.test.com/psp/test/EMPLOYEE/HRMS/' || 'c/'  
       || PORTAL_URI_SEG1 || '.' || PORTAL_URI_SEG2 || '.' ||  
       PORTAL_URI_SEG3 AS Address_string  
FROM PSRSMPerm A, PSRSMDEFN B  
WHERE A.PORTAL_NAME = B.PORTAL_NAME  
AND A.PORTAL_REFTYPE = B.PORTAL_REFTYPE AND  
     A.PORTAL_OBJNAME = B.PORTAL_OBJNAME AND  
     B.PORTAL_URI_SEG2 = 'HRS_CE'
```

Find where the Component is in PeopleSoft Portal and Permission List

-- 3 permission, menu, component and Address String by Label name in Portal name

```
SELECT PORTAL_PERMNAME AS Permission_list,  
       PORTAL_URI_SEG1 AS Menu , PORTAL_URI_SEG2 AS  
       component, 'https://www.test.com/pspr3/EMPLOYEE/HRMS/' ||  
       'c/' || PORTAL_URI_SEG1 || '.' || PORTAL_URI_SEG2 || '.' ||  
       PORTAL_URI_SEG3 AS Address_string  
FROM PSRSMPerm A, PSRSMDEFN B  
WHERE A.PORTAL_NAME = B.PORTAL_NAME  
AND A.PORTAL_REFTYPE = B.PORTAL_REFTYPE      AND  
     A.PORTAL_OBJNAME = B.PORTAL_OBJNAME      AND  
     PORTAL_LABEL = 'Careers'
```

Find where the Page is in the PeopleSoft Portal System

How to find where the page is in PeopleSoft? (portal system)

The data was gathered from the portal information, then you narrow the fields down to one PORTAL_URLTEXT. Use the component name to retrieve the Rows for gathering the address

```
SELECT *  
FROM PSPRSMPerm A, PSPRSMDEFN B  
WHERE A.PORTAL_NAME = B.PORTAL_NAME  
AND A.PORTAL_REFTYPE = B.PORTAL_REFTYPE  
AND A.PORTAL_OBJNAME = B.PORTAL_OBJNAME  
AND B.PORTAL_URI_SEG2 = 'HR_SSTEXT_LANG'
```

Find where the Page is in the PeopleSoft Portal System

Run this SQL too...

```
SELECT 'https://www.test.com/psp/pspr3/EMPLOYEE/HRMS/' || 'c/' ||  
       PORTAL_URI_SEG1 || '.' || PORTAL_URI_SEG2 || '.' ||  
       PORTAL_URI_SEG3 AS Address_string  
FROM PSPRSMPerm A, PSPRSMDEFN B  
WHERE A.PORTAL_NAME = B.PORTAL_NAME  
AND A.PORTAL_REFTYPE = B.PORTAL_REFTYPE  
AND A.PORTAL_OBJNAME = B.PORTAL_OBJNAME  
AND B.PORTAL_URI_SEG2 = 'HR_SSTEXT_LANG'
```

Secrets of PSAUTHITEM

- Excerpts taken from an article written by Bret Martin
April, 2007
- PSAUTHITEM is one of PeopleSoft's core security tables
- The primary purpose is to track which pages and authorized actions that each permission list is assigned
- It also tracks web library (iScript) permissions

Secrets of PSAUTHITEM

columns in PSAUTHITEM:

CLASSID

MENUNAME

BARNAME

BARITEMNAME

PNLITEMNAME

DISPLAYONLY

AUTHORIZEDACTIONS

Secrets of PSAUTHITEM

Query that shows everything you need to set up on-line security

Component is the PNLGRPNAME field.

```
select a.MENUNAME, b.PNLGRPNAME,  
       a.PNLITEMNAME, a.BARNAME, a.BARITEMNAME,  
       a.AUTHORIZEDACTIONS  
from PSAUTHITEM a, PSMENUITEM b  
   where a.menuname = b.menuname  
and    a.barname = b.barname  
and    a.baritemname = b.itemname  
and    a.classid = 'ALLPAGES'
```

Secrets of PSAUTHITEM

Authorized Actions as a 4-bit field with each bit representing an action. Here's what the bit positions mean:

Correction		Update/DisplayAll		Update/Display		Add	
1		1		1		1	=
2^3	+	2^2	+	2^1	+	2^0	=
8	+	4	+	2	+	1	= 15

So to grant a permission list full access to a page, this value will be equal to 15

Secrets of PSAUTHITEM

AUTHORIZED ACTIONS return something a little more friendly

```
SELECT classid, menuname, barname, baritemname, pnitemname,  
       displayonly,  
       case when bitand(authorizedactions,1) > 0 then 'Y' else 'N' END  
       ADDACTION,  
       case when bitand(authorizedactions,2) > 0 then 'Y' else 'N' END  
       UPDATEACTION,  
       case when bitand(authorizedactions,4) > 0 then 'Y' else 'N' END  
       UPDATEALLACTION,  
       case when bitand(authorizedactions,8) > 0 then 'Y' else 'N' END  
       CORRECTIONACTION,  
       case when authorizedactions > 15 then 'Y' else 'N' END SPECIAL  
FROM psauthitem  
where classid = 'ALLPAGES'
```

Security Overview of primary table relations

- *Here is are SQL statements that join the relevant tables to determine who can access what and what can be accessed by whom.*

/ all User IDs */*

```
SELECT OPRID  
FROM PSOPRDEFN  
ORDER BY OPRID
```

/ all role names */*

```
SELECT ROLENAM  
FROM PSROLEDEFN  
ORDER BY ROLENAM
```

Security Overview of primary table relations

/ all permission lists (class names) */*

```
SELECT CLASSID  
FROM PSCLASSDEFN  
ORDER BY CLASSID
```

/ users --> roles */*

```
SELECT ROLEUSER, ROLENAME  
FROM PSROLEUSER  
ORDER BY ROLEUSER, ROLENAME
```

Security Overview of primary table relations

/ users --> permission lists (class names) */*

```
SELECT DISTINCT UR.ROLEUSER, RP.CLASSID  
FROM PSROLEUSER UR, PSROLECLASS RP  
WHERE UR.ROLENAME = RP.ROLENAME  
ORDER BY UR.ROLEUSER, RP.CLASSID
```

/ users --> components */*

```
SELECT DISTINCT UR.ROLEUSER, PC.MENUNAME, PC.BARNAME,  
PC.BARITEMNAME, PC.PNLITEMNAME  
FROM PSROLEUSER UR, PSROLECLASS RP, PSAUTHITEM PC  
WHERE UR.ROLENAME = RP.ROLENAME  
AND RP.CLASSID = PC.CLASSID  
ORDER BY UR.ROLEUSER, PC.MENUNAME, PC.BARNAME,  
PC.BARITEMNAME, PC.PNLITEMNAME
```

Security Overview of primary table relations

/ roles --> permission lists (class names) */*

```
SELECT ROLENAM, CLASSID  
FROM PSROLECLASS  
ORDER BY ROLENAM, CLASSID
```

/ roles --> components */*

```
SELECT DISTINCT RP.ROLENAM, PC.MENUNAM  
, PC.BARNAM, PC.BARITEMNAM, PC.PNLITEMNAM  
FROM PSROLECLASS RP, PSAUTHITEM PC  
WHERE RP.CLASSID = PC.CLASSID
```

XLat changes in 8.9

The xlat table is now it was xlattable

Peoplesoft delivers a view below no PS_ in front of the record

```
SELECT * FROM PSXLATItem
```

```
SELECT * FROM PSXLATdefn
```

```
SELECT * FROM XLATtable_VW
```

Implementation of Roth 401(k)

- Relates only to people that have not installed up to bundle 10 in version 8.9 HRMS
- Extract COBOL programs:
 - PSCDEDT3.CBL
 - PSPDEDLD.CBL
 - PSPDCLIM.CBL
 - PSPDCSAV.CBL
 - PSCLMTTB.CBL
- Then re-compile and link all COBOLs in your library
- Run Data Mover Script:
 - PSDEDLD.DMS

Limit Type Translate Values

Use this page to maintain translate values

Field Name: **LIMIT_TYPE** Length: **4**

Maintain Translate Values

[Customize](#) |
 [Find](#) |
 [View All](#) |
 
 First  1-13 of 13  Last

General
Last Updated


	<u>Value</u>	<u>Effective Date</u>		<u>Status</u>		<u>Long Name</u>	<u>Short Name</u>		
1	401	01/01/1900 		Active 		401(a) Limit	401(a)		
2	401G	01/01/1900 		Active 		401(a) Limit Grandfather Plan	401(a) gfp		
3	402	01/01/1900 		Active 		402(g) Limit	402(g)		
4	403	01/01/1900 		Active 		403(b) Limit	403(b)		
5	415	01/01/1900 		Active 		415(c) Limit	415(c)		
6	415Z	01/01/1900 		Active 		415(c) Limit for 403(b) plans	415z		
7	457	01/01/1900 		Active 		457 Limit	457		
8	CH01	01/01/1900 		Active 		User Defined Per Check Limit 1	Per Check1		
9	CH02	01/01/1900 		Active 		User Defined Per Check Limit 2	Per Check2		
10	CH03	01/01/1900 		Active 		User Defined Per Check Limit 3	Per Check3		
11	CH04	01/01/1900 		Active 		User Defined Per Check Limit 4	Per Check4		
12	CH05	01/01/1900 		Active 		User Defined Per Check Limit 5	Per Check5		
13	USER	01/01/1900 		Active 		User Defined Limit	User Def		

REASON_NOT_TAKEN Translate Values

Use this page to maintain translate values

Field Name: REASON_NOT_TAKEN Length: 1

Maintain Translate Values						
General		Last Updated	Customize Find View All			
			First 1-22 of 22 Last			
*Value	Effective Date	Status	*Long Name		*Short Name	
1 A	01/01/1900	Active	Annual Limit Exceeded		Annual	
2 B	01/01/1900	Active	Annual Limit Exceeded Rollover		Ann R/O	
3 D	01/01/1900	Active	402(g) Limit Exceeded		402 Limit	
4 E	01/01/1900	Active	402(g) Limit Exceeded Rollover		402 R/O	
5 F	01/01/1900	Active	457 Limit Exceeded		457 Limit	
6 G	01/01/1900	Active	457 Limit Exceeded Rollover		457 R/O	
7 H	01/01/1900	Active	HCE Limit Exceeded		HCE Limit	
8 I	01/01/1900	Active	HCE Limit Exceeded Rollover		HCE R/O	
9 K	01/01/1900	Active	415 Limit Exceeded Rollover		415 R/O	
10 L	01/01/1900	Active	415 Limit Exceeded		415 Limit	
11 M	01/01/1900	Active	Maximim Deduction Met		Maximum	
12 N	01/01/1900	Active	Not Enough Net Pay		Net	
13 O	01/01/1900	Active	Per Check Limit Exceeded		CheckLimit	
14 P	01/01/1900	Active	Per Check Limit Exceeded R/O		Check R/O	
15 R	01/01/1900	Active	Deductn Max Exceeded Rollover		Max R/O	
16 S	01/01/1900	Active	401(a) Limit Exceeded		401 Limit	
17 T	01/01/1900	Active	401(a) Limit Exceeded Rollover		401 R/O	
18 U	01/01/1900	Active	403(b) Limit Exceeded		403 Limit	
19 V	01/01/1900	Active	403(b) Limit Exceeded Rollover		403 R/O	
20 W	01/01/1900	Active	User Defined Limit Exceeded		User Limit	
21 X	01/01/1900	Active	User Defined Lmt Exceeded R/O		Usr Def RO	
22 Z	01/01/1900	Active	Multiple Reasons		Multiple	

Sample Limit Table Setup

Limit Table

Limit Type: User Defined Per Check Limit 1

Employee Earnings Limits
[Find](#) | [View All](#)
First 1 of 1 Last

'Effective Date: 01/18/2008
 Rollover Allowed ☐

'Description: Combine 401k/Roth check limit
 Short Description: CheckLimit

Special Accumulator: 401
 Max Percent of Accumulator: 4.000

Maximum Yearly Deduction:

▼ Deductions Subject To Limit
[Customize](#) | [Find](#)
First 1-7 of 7 Last

'Plan Type	Benefit Plan	General Deduction Code	'Ded Class
401(k)	SIP	SIP	Nontaxable
401(k)	SIP	SIP	PTax Ben
401(k)	SSP	SSP	Nontaxable
401(k)	SSP	SSP	PTax Ben
401(k)	SSPM	SSP Max	Nontaxable
401(k)	SSPM	SSP Max	PTax Ben
401 Roth	SIPR	SIP Roth	Nontaxable

▶ Deductions Which Add to Spcl Accum

▶ Deductions Which Subtract From Spcl Accum

Quick Reference to CI

Declare Function CheckErrorCodes PeopleCode FUNCLIB_SL_STAF.EMPLID FieldFormula;

Local string &oprid;

Local ApiObject &delete_user_profile;

Local boolean &error;

/* Get values from the AET record */

&oprid = SL_EP_ROLES_AET.OPRID;

/*Get Session*/

&Session = %Session;

%Session.Connect(1, "EXISTING", "", "", 0);

/* Call CI that will delete the user profile information */

&delete_user_profile = %Session.GetCompIntfc(CompIntfc.DELETE_USER_PROFILE);

&delete_user_profile.InteractiveMode = False;

&delete_user_profile.oprID = &oprid;

&error = False;

/* If error with CI send message to log */

If &delete_user_profile = Null Then

 CheckErrorCodes(&Session, &oprid);

 &error = True;

 MessageBox(0, " ", 0, 0, " profile empty" | &oprid);

/* Application Specific Error Processing */

Quick Reference to CI Continued

Else

```
/* Set Component Interface Standard Properties */
```

```
&delete_user_profile.InteractiveMode = True;
```

```
&delete_user_profile.GetHistoryItems = False;
```

```
&delete_user_profile.EditHistoryItems = False;
```

```
&delete_user_profile.GetDummyRows = False;
```

```
If &delete_user_profile.Get() Then
```

```
  If Not &delete_user_profile.save() Then
```

```
    CheckErrorCodes(&Session, &oprid);
```

```
    &error = True;
```

```
    MessageBox(0, " ", 0, 0, " failed for oprid " | &oprid);
```

```
    rem Error (MsgGet(20200, 15, ""));
```

```
  Else
```

```
    MessageBox(0, " ", 0, 0, " successful for oprid " | &oprid);
```

```
    rem MessageBox(64, MsgGetText(20200, 16, "Success"), 20200, 16, "JOB Data  
    Success.");
```

```
  End-If;
```

```
End-If;
```

```
End-If;
```

```
&delete_user_profile.Cancel();
```

Quick Reference to CI Continued

Declare Function CheckErrorCodes PeopleCode FUNCLIB_SL_STAF.EMPLID FieldFormula;
Declare Function GetRandomPassword PeopleCode FUNCLIB_CS.CREATE_USERID
FieldFormula;

Local string &oprid, &sl_oprid, &sl_snid, &emailid, &sl_email_addr, &sl_rel_sname,
&opralias_type;

Local string &outpassword, &outSymbolicId, &outPrimary, &outData, &outProcess,
&outCurrency, &tmp, &prioremail;

Local ApiObject &user_profile;

Local ApiObject &Session;

Local ApiObject &EmailCol;

Local ApiObject &EmailItm;

Local ApiObject &IDItm;

Local ApiObject &IDCol;

Local boolean &error;

/* Get values from the AET record */

&oprid = SL_EP_ROLES_AET.OPRID;

&sl_oprid = SL_EP_ROLES_AET.SL_OPERID;

&sl_snid = SL_EP_ROLES_AET.SL_SNLID;

&emailid = SL_EP_ROLES_AET.EMAILID;

&sl_email_addr = SL_EP_ROLES_AET.SL_EMAIL_ADDR;

&sl_rel_sname = SL_EP_ROLES_AET.SL_REL_SNAME;

&opralias_type = SL_EP_ROLES_AET.OPRALIASTYPE;

Quick Reference to CI Continued

```
If &sl_rel_sname <> "EMP" Then  
    &sl_rel_sname = "NEM";  
End-If;
```

```
If &sl_operid <> &oprid Or  
    &sl_email_addr <> &emailid Or  
    &sl_rel_sname <> &oprasiatype Then
```

```
/* assign values */  
&outSymbolicId = "SYSADM1";  
&outPrimary = "PPALL";  
&outData = " ";  
&outProcess = "XCSYREPT";  
&outCurrency = "USD";
```

Quick Reference to CI Continued

```
/*Get Session*/
```

```
&Session = %Session;
```

```
%Session.Connect(1, "EXISTING", "", "", 0);
```

```
/* Call CI that will add or change user profile information */
```

```
&user_profile = %Session.GetCompIntfc(CompIntfc.USER_PROFILE);
```

```
&user_profile.InteractiveMode = False;
```

```
&error = False;
```

```
If &user_profile = Null Then
```

```
    CheckErrorCodes(&Session, &oprid);
```

```
    &error = True;
```

```
    MessageBox(0, " ", 0, 0, " oprid does not exist" | &oprid);
```

```
/* Application Specific Error Processing */
```

```
Else
```

```
/* Set Component Interface Standard Properties */
```

```
&user_profile.InteractiveMode = True;
```

```
&user_profile.GetHistoryItems = False;
```

```
&user_profile.EditHistoryItems = False;
```

```
&user_profile.GetDummyRows = False;
```

Quick Reference to CI Continued

```
If RTrim(&oprid) = "" Then
```

```
    MessageBox(0, " ", 0, 0, " oprid does not exist create new " | &sl_oprid | " emplid " |  
    &sl_snld);
```

```
    /* ***** oprid does not exist create new *****/
```

```
    /* create a new userid for this emplid */
```

```
    &user_profile.userID = &sl_oprid;
```

```
    /* create a random password */
```

```
    &outpassword = GetRandomPassword();
```

Quick Reference to CI Continued

If &user_profile.CREATE() Then

```
&user_profile.emplid = &sl_snid;
```

```
/* set security codes */
```

```
&user_profile.SymbolicID = &outSymbolicID;
```

```
&user_profile.setpassword(&outpassword, &outpassword);
```

```
&user_profile.PrimaryPermissionList = &outPrimary;
```

```
&user_profile.RowSecurityPermissionList = &outData;
```

```
&user_profile.ProcessProfilePermissionList = &outProcess;
```

```
&user_profile.CurrencyCode = &outCurrency;
```

```
MessageBox(0, " ", 0, 0, "in create email address " | &sl_email_addr);
```

```
If RTrim(&sl_email_addr) = "" Then
```

```
    &sl_email_addr = " ";
```

```
    &user_profile.emailuser = "N";
```

```
    &user_profile.emailaddresses.DeleteItem(1);
```

Quick Reference to CI Continued

Else

```
&user_profile.emailuser = "Y";
&user_profile.emailid = &sl_email_addr;
/* Get Email Collection Field Properties -- */
&EmailCol = &user_profile.emailaddresses;
&EmailItm = &EmailCol.InsertItem(1);
&EmailItm.emailtype = "BUS";
&EmailItm.emailaddress = &sl_email_addr;
&EmailItm.primaryEmail = "Y";
End-If;

&user_profile.AccountLocked = 0;

/* Set values for ID Types collection */
/* Get ID types Collection Field Properties -- */
&IDCol = &user_profile.IDTypes;
&IDItm = &IDCol.InsertItem(1);
&IDItm.IDtype = RTrim(&sl_rel_sname);
&IDItm.OPRALIASVALUE = &sl_snid;
&IDItm.EMPLID = &sl_snid;
```

Quick Reference to CI Continued

```
/* Set values for ID Types Attributes collection */
```

```
    &IDItm.Attributes.Item(1).AttributeValue = &sl_snld;
```

```
/* invoke SetUserDescription method */
```

```
&tmp = &user_profile.SetUserDescription();
```

```
If Not &user_profile.save() Then
```

```
    CheckErrorCodes(&Session, &oprid);
```

```
    &error = True;
```

```
    MessageBox(0, " ", 0, 0, " Could not create User ID" | &sl_oprid);
```

```
Else
```

```
    MessageBox(0, " ", 0, 0, " successful for oprid " | &sl_oprid);
```

```
End-If;
```

```
End-If;
```

Quick Reference to CI Continued

Else

```
/* get existing userprofile and update with changes */
&user_profile.userID = &oprid;
If &user_profile.Get() Then

    &user_profile.emplid = &sl_snid;
    &user_profile.userID = &sl_oprid;
    MessageBox(0, " ", 0, 0, "in change email address " | &sl_email_addr | " " | &emailid);
    /* Get Email Collection Field Properties -- */
    If RTrim(&sl_email_addr) = "" Then
        &sl_email_addr = " ";
        &user_profile.emailuser = "N";
        &user_profile.emailaddresses.DeleteItem(1);
```

Else

```
    If RTrim(&emailid) = "" Then
        /* Insert email information -- */
        &user_profile.emailuser = "Y";
        &user_profile.emailid = &sl_email_addr;
        &EmailCol = &user_profile.emailaddresses;
        &EmailItm = &EmailCol.InsertItem(1);
        &EmailItm.emailtype = "BUS";
        &EmailItm.emailaddress = &sl_email_addr;
        &EmailItm.primaryEmail = "Y";
```

Else

Quick Reference to CI Continued

```
/* update current information */
    &user_profile.emailuser = "Y";
    &user_profile.emailid = &sl_email_addr;
    &EmailCol = &user_profile.emailaddresses;
    &EmailItm = &EmailCol.item(1);
    &EmailItm.emailaddress = &sl_email_addr;
End-If;
End-If;

/* Set values for ID Types collection */
/* Get ID types Collection Field Properties -- */
&IDCol = &user_profile.IDTypes;
&IDItm = &IDCol.item(1);
&IDItm.IDtype = RTrim(&sl_rel_sname);
&IDItm.OPRALIASVALUE = &sl_snid;
&IDItm.EMPLID = &sl_snid;
```

Quick Reference to CI Continued

```
/* Set values for ID Types Attributes collection */  
    &IDItm.Attributes.Item(1).AttributeValue = &sl_snid;
```

```
/* invoke SetUserDescription method */  
    &tmp = &user_profile.SetUserDescription();
```

```
/* Set values for Roles collection */  
/* Delete roles that are dynamic */  
    &rolecollect = &user_profile.Roles;
```

Quick Reference to CI Continued

```
If Not &user_profile.save() Then
```

```
    CheckErrorCodes(&Session, &oprid);
```

```
    &error = True;
```

```
    MessageBox(0, " ", 0, 0, " failed for oprid " | &oprid);
```

```
Else
```

```
    MessageBox(0, " ", 0, 0, " successful for oprid " | &oprid);
```

```
End-If;
```

```
End-If;
```

```
End-If;
```

```
End-If;
```

```
&user_profile.Cancel();
```

```
If RTrim(&oprid) <> RTrim(&sl_oprid) And
```

```
    RTrim(&oprid) <> "" Then
```

```
/* Call CI that will delete the user profile information */
```

```
&delete_user_profile = %Session.GetComplntfc(Complntfc.DELETE_USER_PROFILE);
```

```
&delete_user_profile.InteractiveMode = False;
```

```
&delete_user_profile.oprID = &oprid;
```

```
&error = False;
```

Quick Reference to CI Continued

```
/* If error with CI send message to log */  
If &delete_user_profile = Null Then  
    CheckErrorCodes(&Session, &oprid);  
    &error = True;  
    MessageBox(0, " ", 0, 0, " profile empty" | &oprid);  
/* Application Specific Error Processing */  
Else  
    MessageBox(0, " ", 0, 0, "in delete profile " | &oprid);  
/* Set Component Interface Standard Properties */  
&delete_user_profile.InteractiveMode = True;  
&delete_user_profile.GetHistoryItems = False;  
&delete_user_profile.EditHistoryItems = False;  
&delete_user_profile.GetDummyRows = False;
```

Quick Reference to CI Continued

```
If &delete_user_profile.Get() Then
    If Not &delete_user_profile.save() Then
        CheckErrorCodes(&Session, &oprid);
        &error = True;
        MessageBox(0, " ", 0, 0, " failed for oprid " | &oprid);
        rem Error (MsgGet(20200, 15, ""));
    Else
        MessageBox(0, " ", 0, 0, " successful for oprid " | &oprid);
        rem MessageBox(64, MsgGetText(20200, 16, "Success"), 20200, 16, "JOB Data
Success.");
    End-If;
End-If;
End-If;

&delete_user_profile.Cancel();

End-If;

End-If;
```

Second CI Example

```
Declare Function CheckErrorCodes PeopleCode FUNCLIB_SL_STAF.EMPLID FieldFormula;
```

```
Local ApiObject &Session;
```

```
Local ApiObject &SL_CI_JOB_DATA;
```

```
Local ApiObject &JOBCol;
```

```
Local ApiObject &JOBItm;
```

```
Local ApiObject &COMPENSATIONCol;
```

```
Local ApiObject &COMPENSATIONItm;
```

```
Local boolean &Error, &success, &Rows_returned;
```

```
Local number &baseamt, &currcomp, &newrate, &effseq, &stdhrs, &mult;
```

```
Local string &emplid, &fullparttime;
```

```
Local string &ACTION, &ACT_REASON, &flsa;
```

```
Local date &effdt, &jobeffdt;
```

```
&emplid = SL_CHCR_LD_AET.EMPLID;
```

```
&effdt = SL_CHCR_LD_AET.EFFDT;
```

```
&baseamt = SL_CHCR_LD_AET.SL_PRPINCRT;
```

```
&ACTION = "PAY";
```

```
&ACT_REASON = "RVW";
```

```
&effseq = 0;
```

```
rem MessageBox(0, "", 0, 0, "emplid " | &emplid);
```

```
&Error = False;
```

```
&Session = %Session;
```

```
&SL_CI_JOB_DATA = &Session.GetCompIntfc(CompIntfc.CI_JOB_DATA);
```

Second CI Example Continued

```
If &SL_CI_JOB_DATA = Null Then
    CheckErrorCodes(&Session, &emplid);
    &Error = True;
    /* Application Specific Error Processing */
Else
    /* Set Component Interface Standard Properties */
    &SL_CI_JOB_DATA.InteractiveMode = True;
    &SL_CI_JOB_DATA.GetHistoryItems = False;
    &SL_CI_JOB_DATA.EditHistoryItems = False;
    &SL_CI_JOB_DATA.GetDummyRows = False;

    /* Set Component Interface Get Keys */
    &SL_CI_JOB_DATA.KEYPROP_EMPLID = &emplid;
    &SL_CI_JOB_DATA.KEYPROP_EMPL_RCD = 0;

    If Not &SL_CI_JOB_DATA.Get() Then
        CheckErrorCodes(&Session, &emplid);
        &Error = True;

    Else
```

Second CI Example Continued

```
/* Get JOB Collection Field Properties -- Parent: PS_ROOT Collection PER_ORG_ASGN */
  &JOBCol = &SL_CI_JOB_DATA.COLL_JOB;
  &jobeffdt = &JOBCol.item(1).keyprop_effdt;

  /*   MessageBox(0, "", 0, 0, "effdt " | &effdt);
  MessageBox(0, "", 0, 0, " job effdt " | &jobeffdt);*/

  &JOBItm = &JOBCol.InsertItem(1);
  /* if effective date of the load is the same as the most current job row get the prior rows effective
  sequence to add to */
  If &jobeffdt = &effdt Then
    &jobcur = &JOBCol.Item(2);
    &effseq = &jobcur.KEYPROP_EFFSEQ + 1;
    /*   MessageBox(0, "", 0, 0, "item effseq " | &jobcur.KEYPROP_EFFSEQ);*/
  Else
    &effseq = 0;
  End-If;

  &JOBItm.KEYPROP_EFFDT = &effdt;
  &JOBItm.KEYPROP_EFFSEQ = &effseq;

  rem   MessageBox(0, "", 0, 0, "effseq " | &effseq);

  &JOBItm.PROP_ACTION = &ACTION;
  &JOBItm.PROP_ACTION_REASON = &ACT_REASON;
```

Second CI Example Continued

```
/* get the current comp rate to calculate new rate */
  &freq = &JOBltm.prop_comp_frequency;
  &currcomp = &JOBltm.prop_comprate;
  &flsa = &JOBltm.prop_flsa_status;
  &fullparttime = &JOBltm.prop_full_part_time;
  &stdhrs = &JOBltm.prop_std_hours;

  If &fullparttime = "F" Then
    If &flsa = "N" Then
      &newrate = Round((&currcomp + &baseamt) / 10, 0) * 10;
    Else
      &newrate = Round((&currcomp + &baseamt) / 100, 0) * 100;
    End-If;
  Else
    &mult = &stdhrs / 40;
    If &flsa = "N" Then
      &newrate = Round((&currcomp + (&baseamt * &mult)), 0);
    Else
      &newrate = Round((&currcomp + (&baseamt * &mult)), 0);
    End-If;
  End-If;
```

Second CI Example Continued

```
&JOBItm.KEYPROP_EFFDT = &effdt;
&JOBItm.KEYPROP_EFFSEQ = &effseq;

&JOBItm.PROP_ACTION = &ACTION;
&JOBItm.PROP_ACTION_REASON = &ACT_REASON;

/* Get COMPENSATION Collection Field Properties -- Parent: JOB Item */
&COMPENSATIONCol = &JOBItm.COLL_COMPENSATION;
&COMPENSATIONItm = &COMPENSATIONCol.Item(1);

&COMPENSATIONItm.PROP_COMPRATE = &newrate;

/* if save does not work write to log */
If Not &SL_CI_JOB_DATA.Save() Then
    CheckErrorCodes(&Session, &emplid);
    &Error = True;
    Error (MsgGet(20200, 15, ""));
    MessageBox(0, "Fail", 0, 0, "Failed to insert JOB Data for emplid " | &emplid);
End-If;
End-If;

End-If;

/* Cancel Instance of Component Interface */
&SL_CI_JOB_DATA.Cancel();
```

Summary

- XML
 - Major Components of XML
 - SOAP
 - XML Publisher for PeopleSoft
 - Publisher Security
 - Flat File Layout
 - Tags
 - Basic Setup
 - Report Categories
 - Data Source Creation
 - XML Application Packages
 - Register the Data Source
 - XML Files in SQL Dev
 - Putting it Together

Summary Continued

- Tips and Tricks
 - Helpful Links
 - Email from Application Engine
 - PL/SQL
 - PL/SQL in SQR
 - SQR Template
 - SQC Example
 - UTIL_FILE
 - How to find bad SubPages
 - Finding Hidden Folders
 - Where is email value stored?

Summary Continued

- Tips and Tricks Continued
 - How to find Message Catalogs by date
 - Find what access a Role gives a User
 - Find where the Component is in PeopleSoft Portal and Permission List
 - Find where the Page is in the PeopleSoft Portal System
 - Secrets of PSAUTHITEM
 - Security Overview of primary table relations
 - XLat changes in 8.9
 - Implementation of Roth 401(k)

Summary Continued

- Component Interface Quick Reference
 - PeopleCode Example
 - Second PeopleCode Example

Become a Complete Oracle Technology and Database Professional

- **Join the IOUG online at www.ioug.org and get immediate access to:**
 - Member Discounts and Special Offers
 - SELECT Journal
 - Library of Oracle Knowledge (LoOK)
 - Member Directory
 - Special Interest Groups
 - Discussion Forums:
 - Access to Local and Regional Users Groups:
 - 5 Minute Briefing: Oracle
 - Volunteer Opportunities

Questions/Comments/Discussion?



Save the Date!



May 3-7, 2009

**Orange County Convention Center West
Orlando, Florida**

Thank You!

Judi Doolittle, jhotsin@sandia.gov