3rd March

Form Customization

Form Customization:
Pre requisites for form customization.
Get the library files (Resource folder files , APPSTAND, APPTREE, APSTAND and TEMPLATE) from the au top and place them in C:\orant\FORMS60.
Then change the start in location to the place where you copied the library files.

Save it.
Now open the form builder →
When you are creating a new form, make sure that you are taking the source file as template file and develop the form.
Cd /shared-u11/oracle/erpdevappl/custom/1.0.0/forms/US

f60gen module=XXVLDT.fmb userid=user/Password module_type=FORM output_file=/shared-u11/oracle/erpdevappl/custom/1.0.0/forms/US/XXVLDT.fmx compile_all=special

Posted 3rd March by Krishnareddy

0 Add a comment

3rd March

PL/ SQL Content

PL/ SQL: [http://www.oracleappsonlinetraining.com/PL_SQl.html]

1. **Introduction** [http://www.oracleappsonlinetraining.com/Introduction.html]
2. **Exceptions** [http://www.oracleappsonlinetraining.com/Exception.html]
3. **Cursors** [http://www.oracleappsonlinetraining.com/Cursors.html]
4. **Procedures** [http://www.oracleappsonlinetraining.com/Procedures.html]
5. **Functions** [http://www.oracleappsonlinetraining.com/Functions.html]

6. **Packages** [http://www.oracleappsonlinetraining.com/Packages.html]
7. **Triggers** [http://www.oracleappsonlinetraining.com/Triggers.html]

Introduction:-

1) What is Pl/SQL ?

It is extension to SQL language.
PL/SQL = SQL + Programming features.
The following are the advantages of PL/SQL
1) We can use programming features like If stmt, loops, branching etc;
2) We can have user definied error messages by using the concept of exception handling.
3) We can perform related actions by using the concept of triggers.
4) Pl/SQL helps in reducing the network traffic.
PL/SQL Block structure:

```
----------------------
declare
..... -- Declare section
.....
.....
begin
..... -- Executable section
.....
.....
.....
.....
exception
..... -- Exception section
.....
end;
/
```

A Pl/SQL block contains 3 sections.
1) Declare section
2) Executable section
3) Exception Section
1) Declare section:

It is used to declare local variables, Cursor , exceptions etc;. All the lines between declare and begin is called declare section.This section is optional.
2) Executable Section:

The actual task which should be done is written in the executable section. All the lines between Begin and exception keywords is called as Executable section.
This section is mandatory
3) Exception Section:

If an exception is raised in the executable section,
control enters into exception section.
All the lines between exception and end is called exception section. This section is optional.
Ex1:

Write a PL/SQL block to display 'Hello World'.
For this program, we do not need any local variables.
So, we can start the program by using keyword begin.

Before the running this program, we need to make the environment variable serveroutput to ON.
To command to make the serveroutput to ON]

```
SQL> Set serveroutput on
Begin
dbms_output.put_line('Hello World');
end;
/
Hello World
Pl/SQL procedure successfully completed.
```

Ex 2:

Write a PL/SQL block to calculate sum of two numbers.
For this program, we need 3 variables, so we need declare section.
Syntax to declare variable:

<variable> <datatype>(size);

```
Declare
a number(3);
b number(3);
c number(4);
begin
a :=10;
b :=20;
c := a+b;
dbms_output.put_line ( ' The sum is ...'||c);
end;
/
The sum is ...30
Pl/SQL procedure successfully completed.
```

In the above program, there are two important points to learn.
i) := is assignment operator, which is used to assign value from the right hand side to the variable in the left hand side.
ii) || (pipe) is concatenation operator.
We can initilize at the time of declaration.

```
declare
a number(3) :=10;
b number(3) :=20;
```

In the abvoe program, we have hard coded the value 10 and 20 in the program. Instead of hard coding the value, we can accept the values from the user.
Ex 3:

Write a program to accept two values from the user and display its sum.

```
Declare
a number(3);
b number(3);
c number(4);
begin
a := &a;
b := &b;
c := a+b;
dbms_output.put_line('The sum is ...'||c);
end;
/
```

Enter a value for A:40
Enter a value for B:30
The sum is ...70
Pl/SQL procedure successfully completed.
Note: & operator is used to accept value from the user.
Ex 4:

Write a PL/SQL block to accept empno and increments his salary by 1000.
Note: To increment the salary (change the value) in a table, we need to use update command.

```
Declare
l_empno number(4);
begin
l_empno := &empno;
update emp set sal = sal+1000
where empno = l_empno;
end;
/
```

Enter a value for empno: 7900
Procedure successfully completed.
To make the above update command permanent, we can use commit after update command in PL/SQL block.
ex:

```
Declare
l_empno number(4);
begin
l_empno := &empno;
update emp set sal = sal+1000
where empno = l_empno;
commit;
end;
/
```

Writing a select stmt in a PL/SQL Block:

Write a pl/SQL block which accepts empno and display ename and salary.
As ename and sal are the values present in the emp table, to get those values we need to write a select stmt.
Note: Every select stmt in a PL/SQL block should have into clause.

```
Declare
l_empno number(4);
l_ename varchar2(20);
l_sal number(5);
begin
l_empno := &empno;
select ename,sal into l_ename, l_sal from emp
where empno = l_empno;
dbms_output.put_line(l_ename||'....'||l_sal);
end;
/
```

Note:

As the above select stmt selects two columns, we need two local variable to catch the value returned by the select stmt.

Using %TYPE attribute:

%TYPE attribute is used to declare the local variables.
Instead of hardcoding the datatype and size for local variable, we can use %TYPE attribute.
Ex:
l_ename varchar2(20); -- we are hard coding datatype and size
l_ename emp.ename%TYPE; --- The datatype of ename column
of emp table is applicable to the local variable.
The above program, i use %TYPE attribute to declare local variables.

```
Declare
l_empno emp.empno%TYPE;
l_ename emp.ename%TYPE;
l_sal emp.sal%TYPE;
begin
l_empno := &empno;
select ename,sal into l_ename, l_sal from emp
where empno = l_empno;
dbms_output.put_line(l_ename||'....'||l_sal);
end;
/
```

Using %ROWTYPE Attribute:

A ROWTYPE variable is capable of holding complete row
of table.
Ex:

Write a PL/SQL Block which accepts an empno and display ename, sal, hiredate and job.

```
declare
l_empno emp.empno%TYPE;
l_row emp%ROWTYPE;
begin
l_empno := &empno;
select * into l_row from emp
where empno = l_empno;
dbms_output.put_line(l_row.ename);
dbms_output.put_line(l_row.sal);
dbms_output.put_line(l_row.hiredate);
dbms_output.put_line(l_row.job);
end;
/
```

Note: we cannot print a ROWTYPE variable, we can print a value of a ROWTYPE variable.

Exceptions:-

1) What is Exception?

Every error in Oracle is an exception.

2) Types of exceptions?

Exceptions are divided into three types
1) Pre definied exceptions
2) NoN pre definied exceptions
3) User definied exceptions

Pre Definied Exceptions:

These exceptions will have exception name and exception number.
The following are some of the examples of predefinied exceptions.
EXCEPTION_NAME EXCEPTION_NUMBER
1) NO_DATA_FOUND
2) TOO_MANY_ROWS
3) ZERO_DIVIDE
4) VALUE_ERROR
5) DUP_VAL_ON_INDEX
1) NO_DATA_FOUND :

This exception is raised when select does not return any row in PL/SQL block.
ex:

```
declare
l_sal emp.sal%type;
begin
dbms_output.put_line('Welcome');
select sal into l_Sal from emp
where empno = 2255;
dbms_output.put_line('The sal is ....'||l_sal);
dbms_output.put_line('Thank You');
end;
/
```
Output:

Welcome
error
Note: In the above program, we get the output 'Welcome'.
This means that program execution is started.
As we dont have any employee with empno 2255, select stmt does not return any row.
When select stmt does not return any row, NO_DATA_FOUND exception is raised.
Once an exception is raised, control will not execute the remaining stmts of executable section, searches for Exception section.
As we do not have exception section in the program, it is terminated abnormally.

We can make sure that the program is completed normally by catching the exception using Exception section.
Syntax:

```
Declare
.....
.....
begin
.....
.....
.....
Exception
When <Exception_handler> then
.....
.....
end;
/
```

Ex:

```
---
declare
l_sal emp.sal%type;
begin
dbms_output.put_line('Welcome');
select sal into l_Sal from emp
where empno = 2255;
dbms_output.put_line('The sal is ....'||l_sal);
dbms_output.put_line('Thank You');
Exception
when NO_DATA_FOUND then
dbms_output.put_line('Invalid empno');
end;
/
```

Output:

```
------
Welcome
Invalid empno
Pl/SQL Procedure successfully completed.
```

2) TOO_MANY_ROWS:

TOO_MANY_ROWS exception is raised, when select stmt returns more than one row.

Ex:

```
----
declare
l_sal emp.sal%type;
begin
dbms_output.put_line('Welcome');
select sal into l_Sal from emp
where deptno=10;
dbms_output.put_line('The sal is ....'||l_sal);
dbms_output.put_line('Thank You');
end;
/
```

Output:

```
-----------
Welcome
Error
```

Note:

As we get the output 'Welcome', this means that program execution is started.
As the select stmt returns more than one row, TOO_MANY_ROWS exception is raised.
As we know, Once an exception is raised control will not execute the remaining lines of excutable section, searches for the Exception section.
As we do not have exception section, program is terminated abnormally.
We can avoid abnormal termination of the program by catching the Exception.

Ex:

```
---
declare
l_sal emp.sal%type;
begin
```

```
dbms_output.put_line('Welcome');
select sal into l_Sal from emp
where deptno=10;
dbms_output.put_line('The sal is ....'||l_sal);
dbms_output.put_line('Thank You');
Exception
When TOO_MANY_ROWS then
dbms_output.put_line( 'Select stmt returns more than one row');
end;
/
```

Output:

```
--------
Welcome
Select stmt returns more than one row.
Pl/SQL Procedure successfully completed.
```

3) ZERO_DIVIDE:

This exception is raised, when we divide a number by zero.

Ex:

```
----
Declare
a number(4);
begin
dbms_output.put_line('Welcome');
a := 10/0;
dbms_output.put_line(a);
dbms_output.put_line('Thank You');
end;
/
```

Output:

```
--------
Welcome
Error
```

Note:

In the above program, as we are dividing by zero, ZERO_DIVIDE exception is raised.
As we are not catching the exception, program is terminated abnormally.
As a developer, we need to make sure that programs are completed successfully at any case.
SO we need to handle exception which is raised by using the Exception Section.

Ex:

```
--------
Declare
a number(4);
begin
dbms_output.put_line('Welcome');
a := 10/0;
dbms_output.put_line(a);
dbms_output.put_line('Thank You');
Exception
When ZERO_DIVIDE then
dbms_output.put_line('DO not divide by 0');
end;
/
```

Output:

Welcome
DO not divide by 0.
Pl/SQL Procedure successfully completed.

4) VALUE_ERROR:

This exception is raised, when the value which is returned does not match with the datatype variable.
Ex:

```
Declare
l_ename number(10);
begin
dbms_output.put_line('Welcome');
select ename into l_ename from emp
where empno = 7369;
dbms_output.put_line('The employee name is...'||l_ename);
end;
/
```
Output:

Welcome
Error
Note:

As the select stmt returning char value, it cannot be stored in varible of number data.
In this case VALUE_ERROR exception is raised.
As we are not catching the exception, program is terminated abnormally.
We can avoid abnormal termination of the program by catching the exception using Exception Section.
Ex:

```
Declare
l_ename number(10);
begin
dbms_output.put_line('Welcome');
select ename into l_ename from emp
where empno = 7369;
dbms_output.put_line('The employee name is...'||l_ename);
Exception
when VALUE_ERROR then
dbms_output.put_line('Pl check the datatype of the local variables');
end;
/
```
Output:

Welcome
Pl check the datatype of the local variables

5) DUP_VAL_ON_INDEX:

This exception is raised when we try to insert a dulicate value on a primary key or unique key.

ex:

Create the following table:
```
create table student ( sno number(3) primary key,
sname varchar2(20),
marks number(3));
```
insert a row in the table:
```
insert into student values (101,'arun',40);
commit;
begin
dbms_output.put_line ( 'Welcome');
insert into student values (101,'vijay',50);
dbms_output.put_line ( 'Thank You');
end;
/
```
Output:

Welcome
Error
Note:

As we are inserting a duplicate value in a primary key column, DUP_VAL_ON_INDEX exception is raised. As we are not catching the exception program is terminated abnormally.
We can avoid abnormai termination of the program by catching the exception.
Ex:

```
begin
dbms_output.put_line ( 'Welcome');
insert into student values (101,'vijay',50);
dbms_output.put_line ( 'Thank You');
Exception
when DUP_VAL_ON_INDEX then
dbms_output.put_line('Do not insert duplicate value in a primary key');
end;
/
```
Output:

Welcome
Do not insert duplicate value in a primary key
When Others handler:

When others can handle any type of exception
Ex1:

```
Declare
a number(4);
begin
dbms_output.put_line('Welcome');
a := 10/0;
dbms_output.put_line(a);
dbms_output.put_line('Thank You');
Exception
When others then
```

```
dbms_output.put_line('Pl check the code');
end;
/
```

Output:

```
-----------
Welcome
Pl check the code
```

Note:

Exception that is raised is ZERO_DIVIDE.

We do not have ZERO_DIVIDE handler, but When Others can handler can handle this exception.

Ex2:

```
---------
declare
l_sal emp.sal%type;
begin
dbms_output.put_line('Welcome');
select sal into l_Sal from emp
where deptno=10;
dbms_output.put_line('The sal is ....'||l_sal);
dbms_output.put_line('Thank You');
Exception
When others then
dbms_output.put_line('Pl check the code');
end;
/
```

Output:

```
-------------
Welcome
Pl check the code
```

++++++++++++++++++++++++++++++++++++

Non predefinied exception:

These exceptions will have exceptio number , but does not have exception name.

Ex:

ORA-2292 exception. This exception is raised when we try to delete from row from the parent table if correspoding row exists in the child table.

First lets establish parent-child relationship between two tables.

```
create table student2( sno number(3) primary key,
sname varchar2(20),
marks number(3));
insert into student2 values (101, 'arun',40);
insert into student2 values (102, 'varun',50);
insert into student2 values (103, 'kiran',60);
create table library2 ( roll_no number(3) references student2(sno),
book_name varchar2(20));
insert into library2 values (101,'Java');
insert into library2 values (102,'C++');
insert into library2 values (102,'Oracle');
commit;
begin
dbms_output.put_line('Welcome');
```

```
delete from student2 where sno =101;
dbms_output.put_line('Thank You');
end;
/
```

Output:

```
Welcome
Error
```

Note: We are deleteting the row from the parent table and the corresponding row exists in the child table. So exception is raised. The exception which is raised in the above program is ORA-2292. This exception does not have any name. This is an example of non -predefinied exception.

The following steps are to followed to handle non-pre definied exception.

Step 1: Declare the exception

Step 2: Associate the exception

Step 3: Handle then exception.

Syntax:

Step 1: Declare the exception

<Exception_name> Exception;

Step 2: Associate the exception

raise_application_error (<exception_no> , <Exception_name>);

Step 3: Handle the exception

```
Exception
When < Exceptionb_name> then
.....
.....
.....
end;
/
```

Ex:

In the follwoing program , we perform the three step process to handle Non-pre definied exceptions.

```
Declare
MY_EX1 Exception;
Raise_application_error ( -2292 , MY_EX1 );
begin
dbms_output.put_line('Welcome');
delete from student2 where sno =101;
dbms_output.put_line('Thank You');
Exception
When MY_EX1 then
dbms_output.put_line('Cannot delete from the parent table');
end;
/
```

Output:

```
Welcome
Cannot delete from the parent table
```

3) User definied exceptions:

These exceptions are definied by the user.
Following steps are to be followed to handle user definied exceptions.
Step 1: Declare the exception
Step 2: Raise the exception
Step 3: Handle the exception.
Ex:

```
---
Declare
l_sal emp.sal%type;
my_ex1 exception;
begin
dbms_output.put_line('Welcome');
select sal into l_sal from emp
where empno =7902;
if l_sal > 2000 then
raise my_ex1;
end if;
dbms_output.put_line('The sal is ....'||l_sal);
Exception
When my_ex1 then
dbms_output.put_line(' Sal is too high');
When others then
dbms_output.put_line('Pl check the code');
end;
/
```

Output:

```
------------
Welcome
Sal is too high
```

Using raise_application_error:

```
----------------------------------------
```

raise_application_error is a procedure which is used to throw a user defined error error_number and error_message to the application.
Ex:

```
------
Declare
l_sal emp.sal%type;
begin
dbns_output.put_line('Welcome');
select sal into l_sal from emp where empno = 7902;
if l_sal > 2000 then
raise_application_error ( -20150, ' Sal is too high');
end if;
dbms_output.put_line('The sal is ....'||l_sal);
end;
/
```

Ouptut:

```
----------
Welcome
ORA-20150 , Sal is too high
```

Error Reporting functions:

```
-----------------------------------
```

Cursors:-

Cursor is a memory locations which is used to run SQL commands.
There are two types cursors
1) Implicit Cursors
2) Explicit Cursors
1) Implicit Cursors:

All the activited related to cursor like i) Opening the cursor ii) Processing the data in the cursor iii) closing the cursor
are done automatically.
Hence these cursors are called Implict cursors.
Implicit Cursor Attributes:

There are four Implicit cursor attributes
1) SQL%ISOPEN
2) SQL%FOUND
3) SQL%NOTFOUND
4) SQL%ROWCOUNT
1) SQL%ISOPEN:

It is a boolean attribute. It always returns false. It is not used in programming as it always returns false.
2) SQL%FOUND:

It is a boolean attribute.
Returns TRUE -- if the SQL command effects the data.
Returns FALSE -- if the SQL commands do not effect the data.
3) SQL%NOTFOUND:

It is a boolean attribute
Returns TRUE -- if the SQL command do not effect the data.
Returns FALSE -- if the SQL command effects the data
Note: It is exactly negation to SQL%FOUND
4) SQL%ROWCOUNT:

Returns no of rows effected by the SQL command.
Using SQL%FOUND:

```
Begin
Update emp set sal=2000
where empno=1111;
end;
/
```
Output:

PL/SQL Procedure successfully completed.
By looking at the above message, we cannot know whether your update command is effecting the data or not.
To overcome this problem, we have SQL%FOUND attribute.
Have a look at this program
```
Begin
Update emp set sal=2000
where empno=1111;
if SQL%FOUND then
dbms_output.put_line('Update is successfull');
```

```
else
dbms_output.put_line('Update is failed');
end if;
end;
/
```

Output:

Update is failed.
PL/SQL Procedure successfully completed.

Using SQL%NOTFOUND:

SQL%NOTFOUND is exactly opposite to SQL%FOUND.
We rewrite the above program using SQL%NOTFOUND

```
Begin
Update emp set sal=2000
where empno=1111;
if SQL%NOTFOUND then
dbms_output.put_line('Update is failed');
else
dbms_output.put_line('Update is successful');
end if;
end;
/
```

Output:

Update is failed.
PL/SQL Procedure successfully completed.

Using SQL%ROWCOUNT:

SQL%ROWCOUNT attribute is used to find the no of rows effected by SQL command.

```
begin
update emp set sal=2000
where deptno=10;
dbms_output.put_line(SQL%ROWCOUNT||' rows updated');
end;
/
```

Output:

---

3 rows updated.
Note: As a developer, we cannot control the implicit cursor.
We can you these implicit cursor attributes to know whether the command is effecting the data or not.

Explicit Cursors:

Explicit cursors are used to run select stmt which returs more than one row in a PL/SQL block

Steps to use Explicit cursors:

Step 1: Declare the cursor
Step 2: Open the cursor
Srep 3: Fetch the data from the cursor to the local variables
Step 4: close the cursor

Syntax of the above four steps:

Step 1: Declaring the cursor
cursor < cursor_name>
is < select stmt >;
step 2: Open the cursor
open < cursor_name >;
step 3: Fetch the data from the cursor to the local variables
fetch < cursor_name > into < var1 > , < var2> , ..., < varn >;;
step 4: close the cursor
close < cursor_name>;
Explicit cursor attributes:

There are four explicit cursor attributes
1) %ISOPEN
2) %FOUND
3) %NOTFOUND
4) %ROWCOUNT
1) %ISOPEN:

It is a boolean attribute.
Returns TRUE -- if the cursor is open
Returns FALSE -- if the cursor is closed
2) %FOUND:

It is a boolean attribute
Returns TRUE -- if the fetch stmt is successfull
Returns FALSE -- if the fetch stmt fails
3) %NOTFOUND:

It is boolean attribute
Returns TRUE -- if the fetch stmt fails.
Returns FALSE -- if the fetch stmt is successfull
Note: 1) It is exactly opposite to %FOUND attribute
2) This attribute is used to break the loop of the fetch stmt.
4) %ROWCOUNT:

Returns no of rows fetched by the fetch stmt.
Example of Explicit cursor:

Write a PL/SQL block to display ename and sal of employees working in deptno no

```
Declare
cursor c1
is select ename , sal from emp
where deptno=10;
l_ename emp.ename%type;
l_sal emp.sal%type;
begin
open c1;
loop
fetch c1 into l_ename , l_sal;
exit when c1%notfound;
dbms_output.put_line( l_ename||'....'||l_sal);
end loop;
close c1;
end;
```

```
/
Output:
------------
CLARK 2450
KING 5000
MILLER 1300
Pl/SQL Proceudure successfully completed.
Ex2: Write a PL/SQL procedure to display dname , loc from dept table
Declare
cursor c1
is select dname , loc from dept;
l_dname dept.dname%type;
l_loc dept.loc%type;
begin
open c1;
loop
fetch c1 into l_dname, l_loc;
exit when c1%notfound;
dbms_output.put_line(l_dname||'.....'||l_loc);
end loop;
close c1;
end;
/
Output:
--------------
Accounting New York
Research Dallas
Sales Chicago
Operations Boston
Pl/SQL Procedure successfully completed.

Cursor For loops:
------------------------
It is shortcut way of writing explicit cursors.
When we use cursor for loops , following steps are not required.
1) Open the cursor
2) Fetch stmt
3) exit when condition
4) closing the cursor
5) declaring the local variables
Ex:
----------
Write a PL/SQL block which display ename and sal of employees working in deptno 10
Declare
cursor c1
is select ename , sal from emp
where deptno=10;
begin
for emp_rec in c1 loop
dbms_output.put_line(emp_rec.ename||'.....'||emp_rec.sal);
end loop;
end;
/
Output:
```

CLARK 2450
KING 5000
MILLER 1300
Pl/SQL Proceudure successfully completed.
Note: In the above program emp_rec in implicitly declared record variable,
which is capable of storing one row of the cursor.
Procedures:-
A Procedure is a named PL/SQL block which is compiled and stored in the database for repeated execution.
Basic Syntax :

Create or replace procedure <procedure_name>
is
begin
.....
.....
.....
end;
/
Ex 1:

Create or replace procedure p1
is
begin
dbms_output.put_line('Hello World');
end;
/
Procedure created.
To execute the procedure:

Exec command is used to execute the procedure.
SQL> Exec p1
Hello World
A procedure can have three types of parameters.
1) IN Parameter
2) OUT Parameter
3) IN OUT Parameter
In Parameters are used to accept values from the user.
Ex 2:

Create a procedure which accepts two numbers and display its sum.
create or replace procedure add_num (a IN number,
b IN number)
is
c number(3);
begin
c := a+b;
dbms_output.put_line(' The sum is '||c);
end;
/
Procedure created.
To execute the procedure:

```
SQL> exec add_num (10,20)
```

Ex 3:

Create a Procedure which accepts an empno and increments his salary by 1000.

```
create or replace procedure inc_sal ( a in number)
is
begin
update emp set sal = sal+1000
where empno = a;
end;
/
```

Procedure created.

TO execute the procedure:

```
SQL> exec inc_sal(7900)
```

We can improve the above procedure code by using %type attribute in procedure parameters.

The above procedure can be re-written as below :

```
create or replace procedure inc_sal ( a in emp.empno%type)
is
begin
update emp set sal = sal+1000
where empno = a;
end;
/
```

Ex 4:

Create a procedure which accepts empno and display ename and salary.

```
create or replace procedure display_emp ( l_empno emp.empno%type)
is
l_ename emp.ename%type;
l_sal emp.sal%type;
begin
select ename, sal into l_ename,l_sal from emp
where empno = l_empno;
dbms_output.put_line(l_ename||'....'||l_sal);
exception
when no_data_found then
dbms_output.put_line('Invalid empno');
end;
/
```

Ex 5:

Create a procedure which accepts deptno and display ename and salary of employees working in that department.

```
create or replace procedure display_emp1 (l_deptno emp.deptno%type)
is
cursor c1
is select ename,sal from emp
where deptno = l_deptno;
begin
for emp_rec in c1 loop
dbms_output.put_line(emp_rec.ename||'...'||emp_rec.sal);
```

```
end loop;
end;
```

Ex 6:

We can call a procedure from another procedure.

```
create or replace procedure demo1
is
begin
dbms_output.put_line('This is from demo1');
end;
/
create or replace procedure demo2
is
begin
dbms_output.put_line ('Welcome');
demo1;
dbms_output.put_line ('Thank you');
end;
/
SQL> Exec demo2
```

Ex 7:

We can call multiple procedures at a time using PL/SQL block.

```
begin
p1;
add_num(10,20);
inc_sal(7900);
end;
/
```

Ex 8:

If there are any syntax errors in the procedure code, then the procedcure is created with compilation errors.

```
create or replace procedure add_num ( a IN number,
b IN number)
is
c number(3);
begin
c := a+b;
dbms_outut.put_line(' The sum is '||c);
end;
/
```

Procedure is created with compilation errrors.

To see the errors, use the following command.

```
SQL> sho err
```

We get error information.

Rectify the error and re compile the code to create procedure successfully.

Ex 9:

Sub procedure: A procedure inside another procedure is called as Sub procedure.

```
create or replace procedure test
is
procedure sample
```

```
is
begin
dbms_output.put_line('This is from sample');
end;
begin
dbms_output.put_line('This is from test');
sample;
end;
```

In the above example procedure sample is called as Sub procedure.
A Sub procedure can be invoked from the main procedure only.

```
SQL> EXEC test
This is from test
This is from sample
```

We cannot invoke the Sub procedure independently.
The following command will give error.

```
SQL>EXEC sample
```

Ex 10:

OUT parameters are used to return the values to the calling environment.
create a procedure which accepts empno and return salary.

```
create or replace procedure ret_sal( l_empno in emp.empno%type,
l_sal out emp.sal%type)
is
begin
select sal into l_sal from emp
where empno = l_empno;
end;
```

As the procedure is returning a value using OUT parameter,
we need to have a bind variable to catch the value. We need to follow a 3 step process to execute the above procedure.
Step 1: Create bind variable
Step 2: Execute the procedure using bind variable
Step 3: Print the value in the bind variable.
Step 1: creating Bind variable

```
SQL> variable g_sal number
```

Step 2: Invoking the procedure using bind variable

```
SQL> Exec ret_sal( 7900, :g_sal)
```

Step 3: Print the value in the bind variable

```
SQL> Print g_sal
```

Ex 11:

IN OUT parameters are used to accept the value as well as return the values to the calling environment.
Create a procedure which accepts a number and return its square.

```
create or replace procedure cal_square( a In OUT number)
is
begin
a := a*a;
end;
/
```

To run the above proceure we need to follow a four step process.
Step 1: Create Bind variable
Step 2: Initiate the Bind variable
Step 3: Invoke the procedure using bind varaible

Step 4: Print the value in the bind variable
Step 1:
SQL> Variable n number
Step 2:
begin
:n :=5;
end;
/
Step 3:
SQL> Exec cal_square (:n)
Step 4:
SQL> Print n
Ex 12:

To see the list of procedures, use the following queries
SQL> select object_name from user_objects where
object_type='PROCEDURE';
or
SQL> select procedure_name from user_procedures.

Ex 13:

Using Default keyword:
---
create or replace procedure add_num3(a number,
b number default 100,
c number default 200)
is
d number(5);
begin
d := a+b+c;
dbms_output.put_line('The sum is ...'||d);
end;
/
Procedure created.
To execute the procedure
SQL> EXEC add_num3(10,20,30)

Output: The sum is 60
SQL> Exec add_num3(10,20)
Output : The sum is 230
Note: Default value is considered if we do not pass any value.
SQL> You need to use arrow operator if you pass values to specific parameters
Ex:
SQL> Exec add_num3(a=>10, c =>20)
Output: The sum is 130
Default value 100 is considered for parameter b.
ex 14:

If there are any errors in the procedure code, then procedure is created with compilation errors.
To see the compilation errors SHO ERR command is used.
We need to rectify the errors and recreate the procedure sucessfully.
Ex 15:

To see the code of the existing procedure
select text from user_source
where name =ADD_NUM3;
TO drop a procedure:

SQL> Drop Procedure <procedure_name>;
Ex:
SQL> Drop procedure add_num;

Functions:-

Function is a PL/SQL block which must and should return single value.
Syntax:

Create or replace function <Function_name>
(<Par_name> <mode> <datatype>,
,, ,, ,,)
return datatype
is
Begin
.....
.....
end;
/
ex1:

Create a function which accepts two numbers and display its sum.

```
create or replace function add_num_f1 ( a number, b number)
return number
is
c number(5);
begin
c :=a+b;
return c;
end;
/
```

To invoke a function from a pl/Sql block:

```
declare
n number(5);
begin
n := add_num_f1(20,40);
dbms_output.put_line('The sum is '||n);
end;
/
```

We can invoke functions from select stmt:

```
select add_num_f1(30,50) from dual;
```

Functions can be invoked as part of an expression:

```
select 100 + add_num_f1(50,10) from dual;
```

Ex2:
---
create a function which accepts sal and returns tax value (10% of sal is tax).

```
create or replace function cal_tax ( a number)
```

```
is
begin
return a*10/100;
end;
/
```

Note: A function can return a value using return statement.

Ex 3:

Have a look at the following function:

```
create or replace function add_num_f2 ( a number, b number)
return number
is
c number(5);
begin
insert into dept values (50,'HR','HYDERABAD')
c :=a+b;
return c;
end;
/
```

The above function gets created.

The above function can be invoked from the pl/SQL block

```
declare
n number(5);
begin
n := add_num_f2(20,40);
dbms_output.put_line('The sum is '||n);
end;
/
```

But, we cannot invoke the above function using select stmt.

ex:

```
select add_num_f2(30,50) from dual; -- will give us error.
```

Note: So, functions with dml commands cannot be invoked from select stmt.

TO see the list of all the functions

```
select object_name from user_objects
where object_type = 'FUNCTION';
```

To drop a function

```
drop function <function_name>;
```

ex:

```
drop function add_num_f2;
```

Functions are mainly used for calculation purposes.

Rest of the activities, prefer procedures.

Packages:-

cPackages are logically related sub programs.

Package creating involves two steps.

Step 1: Creating Package specification (PKS)

Step 2: Creating Package Body (PKB)

Package Specification:

It contains declaration of sub programs
Syntax:
---
create or replace package <package_name>
is
declaration of procedures;
declaration of functions;
end;
/

Package Body:

It contains definition of sub programs
Syntax:
---
create or replace package body <package_name>
is
definition of procedures;
definition of functions;
end;
/

Ex:

Lets create a package with two procedures and function.
Procedure add_num -- which takes two parameters and display its sum.
Procedure display_emp -- which accepts empno and display ename and sal.
Function cal_tax -- which accepts sal and returns tax value (10% of sal is tax value).
Package Specification:
---
create or replace package test_pack
is
procedure add_num (a number,
b number);
procedure display_emp (l_empno emp.empno%type);
function cal_tax (l_sal emp.sal%type)
return number;
end test_pack;
/
Package body:
---
create or replace package body test_pack
is
procedure add_num (a number,
b number)
is
c number;
begin
c := a+b;
dbms_output.put_line('The sum is '||c);
end;
procedure display_emp (l_empno emp.empno%type)
is

```
l_ename emp.ename%type;
l_sal emp.sal%type;
begin
select sal into l_sal from emp
where empno = l_empno;
dbms_output.put_line(l_ename||'.....'||l_sal);
end;
function cal_tax ( l_sal emp.sal%type)
is
l_tax number;
begin
l_tax := l_sal *10/100;
return l_tax;
end;
end test_pack;
/
```

To invoke sub programs inside the package:

```
SQL> EXEC test_pack.display_emp (7900)
SQL> select empno, ename, sal, test_pack.cal_tax (sal) from emp;
```

Procedure overloading using packages:

We can achieve procedure overloading using Packages.
Basing on the no of parameters and datatype of the parameters,
the appropriate procedure is invoked.
ex:

```
Create or replace package test_pack2
is
procedure p1 ( a number,
b number);
procedure p1 ( a number);
end test_pack2;
/
create or replace package body test_pack2
is
procedure p1 ( a number,
b number)
is
c number;
begin
c := a+b;
dbms_output.put_line('The sum is'||c);
end;

procedure p1 ( a number)
is
begin
dbms_output.put_line('The square of the number is '||a*a);
end;
end test_pack2;
/
```

In the above package there are two procedures with the same name.
Appropriate procedure is invoked basing on the no of parameters which are passed

at the time of calling the procedure.
Ex:
--
SQL> exec test_pack2(10, 20);
The sum is 30
SQL> exec test_pack2(10);
The square of the number is 100

To drop the package:

We need to drop package bodu first and then the package specification.
Drop package body <package_name>;
Drop package <package_name>;
Ex:

Drop package body test_pack2;
Drop package test_pack2;
Guidelines of the packages:
---
1) Helps in modularity of the code.
2) Packages cannot be nested.
3) Packages cannot be parameterized.
Triggers:-
Trigger is a PL/SQL block which is executed automatically
basing on a event.
Triggering events: Insert, Update, Delete
Trigger timings: Before, after, instead of
Syntax:

Create or replace trigger <trg_name>
<timing> <event> on <table_name>
begin
.....
.....
.....
end;
/
ex:
---
create or replace trigger trg1
after insert on dept
begin
dbms_output.put_line('Thank You');
end;
/
Trigger Created.
Now, when we peroform the event, trigger is executed,.
ex:
---
insert into dept values (52,'HR','HYDERABAD');
Thank You
1 row created.
We get the message, 'Thank You'.
That means trigger is executed.

We can create triggers on multiple events.
ex:

```
create or replace trigger trg1
after insert or update or delete on dept
begin
dbms_output.put_line('Thank You');
end;
/
```
Trigger created.
Now, for all the three events , triggger is fired.
ex:
--
```
Update dept set loc='DELHI'
where deptno =10;
```
Thank You
1 Row updated.
```
delete from dept where deptno=50;
```
Thank you
1 Row deleted.
In the above program, we get the same message for all the events.
We can also have different messages to be displayed, basing on the events.
Ex:

```
create or replace trigger trg1
after insert or update or delete on dept
begin
if inserting then
dbms_output.put_line('Thank You for inserting');
elsif updating then
dbms_output.put_line('Thank You for updating');
else
dbms_output.put_line('Thank You for deleting');
end if;
end;
/
```
Trigger created.
In the above program, inserting and updating are the key words which are used to identify the events.
Triggers can be classified into two types, basing on the no of times it is executed.
1) Statement level triggers
2) Row level triggers
1) Statement level triggers are executed only once, irrespective of no of rows effected by the event.
2) Row level triggers are executed for every row effected by the event.
To create a row level trigger, we need to use
for-each-row clause.
ex:

```
create or replace trigger trg1
after update on emp for each row
begin
dbms_output.put_line('Thank you for updating');
end;
```

/
Trigger created.
update emp set sal=2000
where deptno=10;
Thank you for updating
Thank you for updating
Thank you for updating
3 rows updated.
As, the update command is effecting 3 rows, trigger is executed 3 times.
These kind of triggers are called row level triggers.
Triggers are used to enforce business rules by
using :OLD and :NEW qualifiers.
ex:

Create a trigger which restrict insert operation
if sal >5000.
create or replace trigger trg1
before insert on emp for each row
begin
if :new.sal >5000 then
raise_application_error(-20150,
' Sal cannot be more than 5000');
end if;
end;
/
Trigger Created.
Event:
----
insert into emp(empno, ename,sal, deptno)
values (1111,'ARUN', 6000,10);
ERROR:
ORA-20150, sal cannot be more than 5000

Ex:

Create a trigger which restrict delete operation on emp
if job is president.
create or replace trigger trg1
before delete on emp for each row
begin
if :OLD.JOB='PRESIDENT' then
raise_application_error(-20151,
' cannot delete president');
end if;
end;
/
Trigger created.
Event:
----
delete from emp where ename='KING';
Error:
ORA-20151, cannot delete president
Instead of triggers:

Instead of triggers are helpful to perform DML operations on complex view.
Example of complex view:

```
-----------------------
create or replace view emp_dept_v
as
select e.empno, e.ename, e.sal, e.deptno, d.dname, d.loc
from emp e, dept d
where e.deptno = d.deptno;
```

View created.
Generally, we cannot insert row into complex view.
But, by using the instead of triggers, we can do it.
ex:

```
----
create or replace trigger trg1
instead of insert on emp_dept_v for each row
begin
insert into dept values (:NEW.deptno,:NEW.dname, :NEW.loc);
insert into emp ( empno, ename,sal, deptno) values
(:NEW.empno, :NEW.ename,:NEW.sal, :NEW.deptno);
end;
/
```

Trigger Created.
Event:

```
------
insert into emp_dept_v values (2121,'VIJAY',3000,60,'TRAINING','HYDERABAD');
```

1 Row created.
To see the list of triggers:

```
------------------------------
select trigger_name from user_triggers;
```

To drop a trigger:

```
-----------------
Drop trigger <trigger_name>;
```

Ex:

```
-----
Drop trigger trg1;
```

Trigger Droped.

Posted 3rd March by Krishnareddy

0 Add a comment

3rd March XML Publisher Questions with Answers

Overview: Oracle XML Publisher is a template-based publishing solution delivered with the Oracle E-Business Suite. It provides a new approach to report design and publishing by integrating familiar desktop word processing tools with existing E-

Business Suite data reporting. At runtime, XML Publisher merges the custom templates with the concurrent request data extracts to generate output in PDF, HTML, RTF, EXCEL (HTML), or even TEXT for use with EFT and EDI transmissions
Basic Need for XML: Consider the following scenarios
We have a RDF report with tabular layout which prints in English
New Requirements:

1. User1 wants the same Report needs to be printed in Spanish
2. User2 wants the Same Report needs to be printed in chart format
3. User3 wants the Same Report output in Excel
4. User4 wants the Same Report output to be published on intranet or internet
5. User5 wants the Same Report output eliminating few columns and adding few other

A new RDF needs to be created for each requirement stated above or an existing RDF needs to be modified with huge amount of effort but whereas with XML Publisher it can be done very easily.
XML Publisher separates a reports data, layout and translation components into three manageable pieces at design time; at runtime all the three pieces are brought back together by XML Publisher to generate the final formatted, translated outputs like PDF, HTML, XLS and RTF. In future, if any there is any change in layout we just need to add/modify the Layout file
Data Logic: Data extracted from database and converted into an XML string.
Layout: The layout templates to be used for the final output are stored and managed in the Template Manager.
Translation: The translation handler will manage the translation that is required at runtime
In brief the steps are as follows:-
a. Create a procedure and register it as Concurrent Program so that we write XML tags into output file.
b. Build a Data Definition & XML Template using XML Publisher.
c. Create a relation between XML Template & Concurrent Program and run the concurrent program
Requirements for XML Data Object Reports

1. Oracle XML Publisher Release 5.5 patch 4206181
2. Template Builder 5.5

Template builder is used to create template/layout for your report. Usually Template builder 5.5 is available in Oracle XML Publisher patch itself but you can also download it from http://edelivery.oracle.com [http://edelivery.oracle.com/] . First select Oracle Application Server Products then select your platform and then locate the Oracle XML Publisher Release 5.6.2 Media Pack v1 for Microsoft Windows, as below:
Download the Desktop edition from the below:
When you download the XML Publisher Desktop edition you get a Zip file containing setup for XML Publisher Desktop Install Shield, this installs some components into Microsoft Word.
After installing, the Word Add-Ins is attached to the menu bar for the word document. This menu lets you attach an XML data source document, add the XML data to your template, set preferences and preview the output.
In detail along with screenshots:-
A concurrent program is written that spit out an XML file as output such concurrent program can be of type SQL or PL/SQL or Oracle Report or any other supportable type, provided it can produce a XML output.
1. Here I have a very simple PL/SQL procedure, which fetch the records from AR tables and write the output in xml tags.

```
CREATE OR REPLACE PROCEDURE APPS.Demo_XML_Publisher (errbuf
```

```
VARCHAR2,retcode NUMBER,v_customer_id VARCHAR2)
AS

/*Cursor to fetch Customer Records*/
CURSOR xml_parent
IS
SELECT   customer_name , customer_id
FROM      ra_customers
WHERE   customer_id = to_number(v_customer_id);

/*Cursor to fetch customer invoice records*/
CURSOR xml_detail(p_customer_id1 NUMBER)
IS
SELECT ra.customer_trx_id customer_trx_id, ra.ship_to_customer_id
ship_to_customer_id, ra.trx_number trx_number,aps.amount_due_original
ams
FROM  ra_customer_trx_all ra, ar_payment_schedules_all aps
WHERE  ra.ship_to_customer_id = p_customer_id1
AND aps.customer_trx_id = ra.customer_trx_id
AND      ROWNUM<4;

BEGIN
/*First line of XML data should be <?xml version="1.0"?>*/
FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<?xml version="1.0"?>');
FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<CUSTOMERINFO>');
FOR v_customer IN xml_parent
LOOP
/*For each record create a group tag <P_CUSTOMER> at the start*/
FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<P_CUSTOMER>');
/*Embed data between XML tags for ex:-
<CUSTOMER_NAME>ABCD</CUSTOMER_NAME>*/
FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<CUSTOMER_NAME>' ||
v_customer.customer_name
|| '</CUSTOMER_NAME>');
FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<CUSTOMER_ID>' ||
v_customer.customer_id ||
'</CUSTOMER_ID>');

FOR v_details IN xml_detail(v_customer.customer_id)
LOOP
/*For  customer invoices create a group tag <P_INVOICES> at the start*/

FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<P_INVOICES>');
FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<CUSTOMER_TRX_ID>' ||
v_details.customer_trx_id || '</CUSTOMER_TRX_ID>');
FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<CUSTOMER_ID>' ||
v_details.ship_to_customer_id || '</CUSTOMER_ID>');
FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<INVOICE_NUMBER>'||
v_details.trx_number||'</INVOICE_NUMBER>');
FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<AMOUNT_DUE_ORIGINAL>'||
v_details.trx_number||'</AMOUNT_DUE_ORIGINAL>');

/*Close the group tag </P_INVOICES> at the end of customer invoices*/
FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'</P_INVOICES>');
END LOOP;

/*Close the group tag </P_CUSTOMER> at the end of customer record*/
```

```
FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'</P_CUSTOMER>');
END LOOP;

/*Finally Close the starting Report tag*/
FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'</CUSTOMERINFO>');
exception when others then
FND_FILE.PUT_LINE(FND_FILE.log,'Entered into exception');
END Demo_XML_Publisher;
/
```

2. Create an executable SampleXmlReport for the above procedure Demo_XMML_Publisher.
Go to Application Developer Responsibility->Concurrent->Executable
3. Create a new concurrent program SampleXmlReport that will call the SampleXmlReport executable declared above. Make sure that output format is placed as XML.
Go to Application Developer Responsibility -> Concurrent ->Program
4. Make sure we declare the parameters for the procedure.
5. Add this new concurrent program with Receivables request group. Either using the following code or through below application screen.

```
DECLARE
BEGIN
FND_PROGRAM.add_to_group
(
PROGRAM_SHORT_NAME  =>'CUST_XML_SAMPLE'
,PROGRAM_APPLICATION =>'AR'
,REQUEST_GROUP       => 'Receivables All'
,GROUP_APPLICATION   =>'AR'
) ;
commit;
exception
when others then
dbms_output.put_line('Object already exists');
END ;
/
```

Go to System Administrator Responsibility ->Security ->Responsibility->Request
6. From the receivables responsibility (depends on which responsibility we added our concurrent program here it is receivables)
From the menu View->Requests->Submit A New Request->Single Request
Note: The layout field is blank as we haven't attached any Template or layout to this concurrent program yet.
By submitting the above request we get the output in xml (depending on procedure) as follows:

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

- [http://oamdev.ventanamed.test:8000/OA_CGI/FNDWRR.exe?temp_id=2392214407]
<CUSTOMERINFO>
- [http://oamdev.ventanamed.test:8000/OA_CGI/FNDWRR.exe?temp_id=2392214407]
<P_CUSTOMER>
<CUSTOMER_NAME>UNIV OF CHICAGO HOSP</CUSTOMER_NAME>
<CUSTOMER_ID>1119</CUSTOMER_ID>
- [http://oamdev.ventanamed.test:8000/OA_CGI/FNDWRR.exe?temp_id=2392214407]
<P_INVOICES>
```

<CUSTOMER_TRX_ID>929476</CUSTOMER_TRX_ID>
<CUSTOMER_ID>1119</CUSTOMER_ID>
<INVOICE_NUMBER>2484403</INVOICE_NUMBER>
<AMOUNT_DUE_ORIGINAL>8000</AMOUNT_DUE_ORIGINAL>
</P_INVOICES>
- [http://oamdev.ventanamed.test:8000/OA_CGI/FNDWRR.exe?temp_id=2392214407]
<P_INVOICES
<CUSTOMER_TRX_ID>929374</CUSTOMER_TRX_ID>
<CUSTOMER_ID>1119</CUSTOMER_ID>
<INVOICE_NUMBER>2484267</INVOICE_NUMBER>
<AMOUNT_DUE_ORIGINAL>380.68</AMOUNT_DUE_ORIGINAL>
</P_INVOICES>
- [http://oamdev.ventanamed.test:8000/OA_CGI/FNDWRR.exe?temp_id=2392214407]
<P_INVOICES>
<CUSTOMER_TRX_ID>806644</CUSTOMER_TRX_ID>
<CUSTOMER_ID>1119</CUSTOMER_ID>
<INVOICE_NUMBER>2421373</INVOICE_NUMBER>
<AMOUNT_DUE_ORIGINAL>615.96</AMOUNT_DUE_ORIGINAL>
</P_INVOICES>
</P_CUSTOMER>
</CUSTOMERINFO>

7. Save the above code as SampleXmlReport.xml

Note: Before saving the XML string in notepad remove the dashes –

8. Create Template/Layout for the report using Template Builder. Here is a sample template.

Note the following:

The data fields that are defined on the template

For example: Customer Name, Customer Id

The elements of the template that will repeat when the report is run.

For example, Customer trx id , Invoice Number and Original Amount Due. All these fields on the template will repeat for each Employee that is reported.

9. Mark up your template layout.

Like a mail-merge document there's placeholders for the data you're going to add, and then you can add whatever formatting you like.

10. Now the next step is to select Data > Load XML Data from the toolbar menu, then pick up the XML data a file ie.SampleXmlReport.xml. Once the data is loaded, a "Data Loaded Successfully" dialog box comes up and you can then start adding data items to the template.

11. To add data items from the XML file into your report, you locate in the document the placeholder for the field you're going to add, highlight it and then select Insert > Field from the toolbar. A dialog box comes up with all of the available data items, you select the one you want and click insert as shown below:

12. You can add repeating rows into your document by selecting Insert > Table/Form from the toolbar. This brings up a different dialog box that lets you drag a parent node – in this case, "P Invoices" – into the middle section, which becomes your repeating rows.

13. Calculate the average for amount due original. Select Insert->Field from the Add Ins toolbar. Then select the tag that calculates the average for that particular field.

14. Once we are done with adding up all the fields in the template save it as an rtf which looks as below:

To confirm the output from the template we build. Click on preview and select the type in which format the output is required.

15. Adding the Template to ORACLE Application.

In order to add the template to application the user should have the responsibility XML Publisher Administrator assigned.

In this step we do 2 processes, registering concurrent program as Data Definition in template manager
And register the template using the data definition created.
Go to XML Publisher Administrator->Data Definitions->Create Data definition.
Here we fill all the details to create data definition
NOTE: Make sure the code of the data definition must be the same as the short name of the Concurrent Program we registered for the procedure. So that the concurrent manager can retrieve the templates associated with the concurrent program
We can add our xml file SampleXmlReport.xml in data definition here:
16. Now create the template with the help of template manager
At the runtime the concurrent managers request interface will present the list of available templates with respect to the data definition registered. We will upload the rtf template file created here. We will select the language and territory.
We can upload different templates for different languages and territories.
17. Now run the concurrent program to get the desired output.
From the receivables responsibility, use the submit request form to run the concurrent request.
The default layout is displayed which we can change as per the requirement. For changing the template click on options to look for all templates register with that concurrent program.
Here in the above example my template is SampleXmlReport which is displayed in layout field.
And once the request is submitted we will get the output in PDF as
This is the final output as desired in the PDF format.

Dynamic Content using Sub Templates

By Tim Dexter-Oracle on Dec 01, 2011

[https://blogs.oracle.com/xmlpublisher/entry/dynamic_content_using_sub_templates]

I have written about sub templates in the past on a few occasions; the principle behind them is pretty simple. If you have common report components that can be shared across reports; be they blocks of text like standard contract clauses or maybe some common calculation or function, drop them into a sub template and share the love. Develop once, use everywhere!
A colleague was recently tasked with conditionally bringing into a report output, paragraphs of static text based on some user preferences. That's an ideal candidate for a sub template approach; drop all of the paragraphs in to an RTF subtemplate and then just conditionally pull them in based on some boolean expressions.
You might, quite naturally think about conditionally importing a series of sub templates rather than defining one, with all the content. However, XSL does not allow the conditional import of sub templates so you must take the single template approach. You can of course, import multiple sub templates if you have a lot of content to bring in but in most cases I would expect a single sub template will suffice.

BIP does need to know what those paragraphs need to be for each user whether that's as a set of parameter values or a data element in the incoming data set. For this example I have used both approaches and they work all flavors of BIP. Implementation of the sub template onto the servers is going to be a little different but the main principle is the same. I have mercilessly ripped out a nice graphic from Leslie's (doc writer extraordinaire) documentation.

This is for the 11g version that supports loading sub templates into the report catalog as objects. They can then be referenced in your main template using the import statement:

<?import:xdoxsl:///subtemplatefolder/subtemplatename.xsb?>

The subtemplate folder is going to be from the /SharedFolders or /My Folders root. For instance, I have a sub template 'paragraphs' loaded into a 'test' folder under Shared Folders. The import statement in my main template is '<? import:xdoxsl:///Test/ParaSubTemplate.xsb?>'
Update from Leslie

For those of you testing using your own My Folder area. The syn tax is <?import:xdoxsl:///~username/path to subtemplate.xsb?> where username is your user name. For example: <?import:xdoxsl:///~tdexter/Subtemplates/Template1.xsb?>
Recommend you move them into the shared folder area in production.
For 10g you will either need to drop them into an accessible directory and use the file URI or mount them into the web server directory structure and access them via an http URI. I normally mount them in a directory under the 'xmlpserver' directory e.g J2EE_HOME\applications\xmlpserver\xmlpserver\subtemplates, a template is then accessible via the URI 'http://server:port/subtemplates/template.rtf'
Make sure you set the Allow External References property to true for the report so that the sub template can be accessed.

The actual content of the sub template is pretty straight forward. It's a series of paragraphs bounded by the 'template' command e.g.

```
<?template:para1?>
…
…
<?end template?>
<?template:para2?>
…
…
<?end template?>
<?template:para3?>
…
…
<?end template?>
```

Now we have the dynamic content defined it's a case of conditionally bringing it into the main template. For this example I have demonstrated two approaches; both rely on the required paragraph information to be in the main dataset:

1. Using parameters to allow the user to select the appropriate paragraphs to be brought in. This means creating the parameters and ensuring that you have set the property on the data model to include the parameter values in the XML result set.

Once that's done its just a simple case of using id statements to check if a given paragraph should be included:

```
<?if:.//PARA1='1'?><?call:para1?><?end if?>
```

This assumes my parameter is called PARA1 and that a '1' means include it, it could easily be a 'Y' or 'True' value, you are just testing for it.

2. Including a value in the data to define what paragraphs should be included. If you

have stored what paragraphs should be included for a given entity i.e. customer, supplier, employee, etc. Then you can extract those values into the data set and test for them. For this demo I have a 5 character set of '1's and '0's to represent the paragraphs that need to be included e.g. 10110. I just use a substring command to find out if a particular paragraph needs to be included.

```
<?if:substring(.//PARAS,1,1)='1'?><?call:para1?><?end if?>
```

Where PARAS is the element holding the '1's and '0's string to be parsed.
You can of course use some other means of marking whether a paragraph needs to be included or not. It's just a case of parsing out the values with a substring or similar command.
You should be able to generate dynamic content such as this:

XML Publisher Interview Questions and Answers

1. What are the XML publisher tables.
   Ans>PER_GB_XDO_TEMPLATES
   XDO_DS_DEFINITIONS_B
   XDO_DS_DEFINITIONS_TL
   XDO_DS_DEFINITIONS_VL
   XDO_LOBS
   XDO_TEMPLATES_B
   XDO_TEMPLATES_TL
   XDO_TEMPLATES_VL
   XDO_TEMPLATE_FIELDS
   XDO_TRANS_UNITS
   XDO_TRANS_UNIT_PROPS
   XDO_TRANS_UNIT_VALUES
2. how to create report with out .rdf?
   Ans> Using Data template…..see below link
   http://appstechnical01.blogspot.in/2012/08/using-multiple-quires-data-template-code.html
   [http://appstechnical01.blogspot.in/2012/08/using-multiple-quires-data-template-code.html]
3. how to write a loop in rtf template design ?
   Ans> <?for-each:G_invoice_no?>
   ……..<?end for each?>
4. how to design sub templates in rtf layout ?
   Ans> using following tags..
   <?template:template_name?>
   This is Last Page
   <?end template?>
5. how to call a header or footer ?
   Ans> using this tag <?call:header?> and <?call:footer?>
   We have to use header section and footer section of the page.

6. How to break the page in specific condition ?
   Ans> <?split-by-page-break:?>
7. How to use section break ?
   Ans> <?for-each@section:G_CUSTOMER(This is group name)?>
8. How to create multi layouts in XMLP ?
   Ans> using below conditions

```
<?choose:?>
<?when:CF_CHOICE='VENDOR'?>
Your template….
<?end when?>
<?when:CF_CHOICE='INVOICE'?>
Your template….
<?end when?>
<?when:CF_CHOICE='RECEIPT'?>
Your template….
<?end when?>
<?end choose?>
```

9. How to calculate the running total in XMLP?

Ans> <?xdoxslt:set_variable($_XDOCTX, 'RTotVar', xdoxslt:get_variable($_XDOCTX, 'RTotVar') + ACCTD_AMT(This is column name))?><?xdoxslt:get_variable($_XDOCTX, 'RTotVar')?>

10. How to submit a layout in the backend ?

Ans> we have to write a procedure for this using the below code

```
fnd_request.add_layout
(template_appl_name => 'application name',
template_code => 'your template code',
template_language => 'En',
template_territory => 'US',
output_format => 'PDF'
);
```

11. How to display the images in XMLP?

Ans> url:{'http://image location'}

For example, enter:

url:{'http://www.oracle.com/images/ora_log.gif'}

url:{'${OA_MEDIA}/image name'}

12. How to pass the page numbers in rtf layout?

Ans> <REPORT>
<PAGESTART>200<\PAGESTART>
....
</REPORT>
Enter the following in your template:
<?initial-page-number:PAGESTART?>

13.How to display last page is differently in XML Publisher Reports.

Ans> what you want to dispay the test anything write in last of page

last page header

<?start@last-page-first:body?> <?end body?>

more questions i will update soon.....

we have any doubts on this please give the comment....i will respond...u r comment

What is BI Publisher?

A. It is a reporting tool for generating the reports. More than tool it is an engine that can be

integrated with systems supporting the business.

Is BI Publisher integrated with Oracle Apps?

Yes, it is tightly integrated with Oracle Apps for reporting needs. In 11.5.10 instances xml publisher was used, in R12 we can it BI Publisher

What is the difference between xml publisher and BI Publisher?

Name is the difference, initially it was released on the name of xml publisher(the initial patchset), later on they have added more features and called it Business Intelligence Publisher. In BI by default we have integration with Datadefinitions in R12 instance. Both these names can be used interchangeably

What are the various components required for developing a BI publisher report?

Data Template, Layout template and the integration with Concurrent Manager.

How does the concurrent program submitted by the user knows about the datatemplate or layout template it should be using for generating the output?

The concurrent program 'shortname' will be mapped to the 'code' of the Datatemplate. Layout template is attached to the datatemplate, this forms the mapping between all the three.

What is a datatemplate?

Datatemplate is an xml structure which contains the queries to be run against the database so that desired output in xml format is generated, this generated xml output is then applied on to the layout template for the final output

What is a layout template?

Layout template defines how the user views the output, basically it can be developed using Microsoft word document in rft (rich text format) or Adobe pdf format. The data output in xml format (from Data template) will be loaded in layout template at run time and the required final output file is generated.

What are the output formats supported by layout template?

xls, html, pdf, eText etc are supported based on the business need.

Do you need to write multiple layout templates for each output type like html/pdf?

No, only layout template will be created, BI Publisher generates desired output format when the request is run

What is the default output format of the report?

The default output format defined during the layout template creation will be used to generate the output, the same can be modified during the request submission and it will overwrite the one defined at layout template

Can you have multiple layout templates for a singe data template?

Yes, multiple layouts can be defined, user has a choice here to use one among them at run time during conc request submission

Where do you register data and layout templates?

Layout template will be registered under xml publisher administrator

responsibility>Templates tab.

Data template will be registered under xml publisher admininstrator responsibility> Data Definitions

I want to create a report output in 10 languages, do I have to create 10 layout templates?

No, BI Publisher provides the required translation for your templates, based on the number of languages installed in your oracle apps environment requires outputs are provided

What is the required installation for using BI Pub report?

BI Publisher deskop tool has be installed. Using this tool you can preview or test the report before deploying the same on to the instance.

How do you move your layout or data template across instances?

xdoloader is the utility that will be used.

What is the tool to map required data output and layout templates so that they can be tested in local machine?

Template viewer will be used for the same.

Which component is responsible for generating the output in xml format before applying it to layout template?

DataEngine will take DataTemplate as the input and the output will be generated in xml format which will then be applied on layout template

Can BI publisher reports be used in OAF pages?

XDO template utility helper java classes are provided for the same.

Name some business use cases for BI reports?

Bank EFT, customer documents, shipping documents, internal analysis documents or any transactional documents

How do you pass parameters to your report?

Concurrent program parameters should be passed, ensure that the parameter name/token are same as in the conc prog defn and the data template

What are the various sections in the data template?

Parameter section
Trigger Section
Sql stmt section
Data Structure section
Lexical Section

What does lexical section contain?

The required lexical clause of Key Flex field or Descriptive FF are created under this section

What triggers are supported in Data template?

Before report and After report are supported

Where is the trigger code written?

The code is written in the plsql package which is given under 'defaultpackage' tag of data template.

what is the file supporting the translation for a layout template?

A. xliff is the file that supports the translation, you can modify the same as required.

Q. How do you display the company logo on the report output?

A. Copy and paste the logo (.gif. or any format) on the header section of .rtf file . Ensure you resize per the company standards.

RTF Template : Working with variables

Let's see how we can use the variables to store temporary data or use for calculation. This is achieved using "xdoxslt:" function. These are the BI Publisher extension of standard xslt functions.
Use xdoxslt:set_variable () function to set /initialize the variable and xdoxslt:get_variable() function to get the variable value. $_XDOCTX is the System variables to set the XDO Context.

```
/*initialize a variables*/
<?xdoxslt:set_variable($_XDOCTX, 'counter', 0)?>
/*update the variable's value by adding the current value to MY_CNT, which is XML element */
<?xdoxslt:set_variable($_XDOCTX, 'counter', xdoxslt:get_variable($_XDOCTX, 'counter') + MY_CNT)?>
/* accessing the variables */
<?xdoxslt:get_variable($_XDOCTX, 'counter')?>

/*Working in a loop*/
<?xdoxslt:set_variable($_XDOCTX, 'counter', 0)?>
<?for-each:G1?>
/*increment the counter*/
<?xdoxslt:set_variable($_XDOCTX, 'counter', xdoxslt:get_variable($_XDOCTX, 'counter') + 1)?>
<?end for-each?>
<?xdoxslt:get_variable($_XDOCTX, 'counter')?>
```

Hope this help in understanding variable gimmicks.
Need help on RTF template design or BI Publisher implementation, please contact info@adivaconsulting.com [mailto:info@adivaconsulting.com]

What is XML Publisher?

Oracle XML Publisher is a template-based publishing solution delivered with the Oracle E-Business Suite. It provides a new approach to report design and publishing by integrating familiar desktop word processing tools with existing E-Business Suite data reporting. At runtime, XML Publisher merges the custom templates with the concurrent request data extracts to generate output in PDF, HTML, RTF, EXCEL (HTML), or even TEXT for use

with EFT and EDI transmissions.

XML Publisher has been ported over to the PeopleSoft enterprise ERP suite. XML Publisher will be the reporting tool of choice from Oracle when Fusion Applications become available.

What are the various sections in the data template?

The data template is an XML document that consists of 5 basic sections:

1. Properties
2. Parameters
3. Triggers
4. Data Query
5. Data Structure

How do you migrate layout or data template across instances?

We can use XDOLoader utility to migrate both data template and layout. Below is the example:

Data Template:

```
java oracle.apps.xdo.oa.util.XDOLoader UPLOAD \
-DB_USERNAME ${apps_user} \
-DB_PASSWORD ${apps_pwd} \
-JDBC_CONNECTION ${TNS_STRING} \
-LOB_TYPE DATA_TEMPLATE \
-APPS_SHORT_NAME XXCUST \
-LOB_CODE XX_DT_REPORT \
-LANGUAGE en \
-TERRITORY US \
-XDO_FILE_TYPE XML \
-NLS_LANG ${NLS_LANG} \
-FILE_CONTENT_TYPE 'text/html' \
-FILE_NAME XX_DTEMPLATE.xml \
-LOG_FILE XX_DTEMPLATE.xml.log \
-CUSTOM_MODE FORCE
FIN_STATUS=$?
```

Layout:

```
java oracle.apps.xdo.oa.util.XDOLoader UPLOAD \
-DB_USERNAME ${apps_user} \
-DB_PASSWORD ${apps_pwd} \
-JDBC_CONNECTION ${TNS_STRING} \
-LOB_TYPE TEMPLATE_SOURCE \
-APPS_SHORT_NAME XXCUST \
-LOB_CODE XX_DT_REPORT \
```

```
-LANGUAGE en \
-TERRITORY US \
-XDO_FILE_TYPE RTF \
-NLS_LANG ${NLS_LANG} \
-FILE_CONTENT_TYPE 'application/rtf' \
-FILE_NAME XX_LAYOUT.rtf \
-LOG_FILE XX_LAYOUT.rtf.log \
-CUSTOM_MODE FORCE
```

Do we need to create multiple layout templates for printing report in multiple languages?

We can achieve multi language report by two ways

1. Different layout template for different languages This approach can be chosen if the layout is also different from language to language.

2. Generate XLIFF template for each language
   XLIFF (XML Localization Interchange File Format) : format for exchanging localization data. XML based format that enables translators to concentrate on the text to be translated. We use this option when we want to use the same layout and apply specific translation.

Can you have multiple layout templates for a singe data template?

Yes! Multiple layouts can be attached, so that user will have a choice here to use one among them at the time of concurrent program submission

How to get a output of a XMLP report in different formats like PDF, DOC, XLS, TXT ?

While submitting the concurrent program you select the output format in options form of "Up on Completions" selection.

What is Data Template and Layout Template?

Data Template:
Datatemplate is an xml structure which contains the queries to be run against the database so that desired output in xml format is generated, this generated xml output is then applied on to the layout template for the final output

Layout Template:
Layout template defines how the user views the output, basically it can be developed using Microsoft word document in rft (rich text format) or Adobe pdf format. The data output in xml format (from Data template) will be loaded in layout template at run time and the required final output file is generated.

How does the concurrent request relate both data

template and layout template it should be using for generating the output?

The concurrent program 'short name' will be mapped to the 'code' of the Data template. Layout template is attached to the data template, this forms the mapping between all the three.

What are the various components required for developing a BI publisher report?

1. Data Template
2. Layout template
3. Integration with Oracle Concurrent Manager

What is the difference between XML publisher and BI Publisher?

Name is the difference, XML Publisher (today) is only found within E-Business Suite. Prior to release 12, it was added in via a patch. In R12, it comes pre-installed. Both these names can be used interchangeably.
XML Pub operates entirely within EBS and can only be used within EBS. BIP can be installed as a standalone version running off of several OC4J compliant engines, such as Application Server and Tomcat. BIP can be pointed anywhere, so you could do reports out of an OLTP or warehouse database, MSSQL and even within EBS.
Licensing is already included in EBS, and the standalone costs whatever plus maintenance.
BI Publisher is based on OC4J, which is a J2EE compliant engine, which for all practical purposes, is a Web server. You see OC4J in lots of other Oracle products which typically have one thing in common: they expose their application to the user via a Web interface. XMLP is also based on this because EBS runs off of Application Server.
Both include a desktop component, which is an add-in to Word, and the work there is the same regardless of which flavor you're using. You can create templates and publish them, there being a slight difference in how each version does that.

What triggers are supported in Data template?

There are two different triggers supported by Data Template:

1. Before Trigger
2. After Trigger

What are the main XML Publisher database Tables?

Here are the few important tables of XMLP:
XDO_DS_DEFINITIONS_B
XDO_DS_DEFINITIONS_TL
XDO_DS_DEFINITIONS_VL
XDO_LOBS
XDO_TEMPLATES_B
XDO_TEMPLATES_TL

XDO_TEMPLATES_VL
XDO_TEMPLATE_FIELDS
XDO_TRANS_UNITS
XDO_TRANS_UNIT_PROPS
XDO_TRANS_UNIT_VALUES

Where and What is the code written in the Data Template Trigger?

The code is written in the plsql package which is mentioned under 'defaultpackage' tag of data template.

what is the file supporting the translation for a layout template?

XLIFF is the file that supports the translation, we have one XLIFF file for each language.

How to do Looping in XMLP?

you can use the below syntax for looping:

<?for-each:looping_node_name?>
.
.
.
<?end for-each?>

How to create sub templates in rtf Template?

Here is the Sub Template tag:

<?template:header?>
…sub template design/tags…
<?end template?>

Here 'header' is the sub template name.

After that you call 'header' sub template any where in the rtf with below syntax:
<?call:header?>

How do you pagebreaks after specific rows?

We can use below tag for page break:
<?split-by-page-break?>
But to use this conditionally we can use below syntax:
<?if:position() mod 5=0 ?>
<?split-by-page-break:?>

<?end if?>

Note: we should not use split-by-page-break in a table layout.

How to create a conditional Layout in XMLP?

Here is the syntax for creating different layouts based on condition:

```
<?choose:?>

    <?when:CF_CHOICE='VENDOR'?>
        ..design your Layout here..
     <?end when?>

     <?when:CF_CHOICE='INVOICE'?>
          ..design your Layout here..
      <?end when?>

     <?when:CF_CHOICE='RECEIPT'?>
          ..design your Layout here..
      <?end when?>

<?end choose?>
```

How to restrict rows in a Group based on some condition in XMLP?

We can use below syntax to restricting the rows in the Group based on condition:

Syntax:
<?for-each:Group_Name[./Field_name with Condition]?>

Example:
<?for-each:EMP_DETAILS[EMP_ADDRESS!=”]?>
In this examples it will filter the Records with EMP_ADDRESS is not Null and Displays in the Report.

How to handle NULL values as Exceptions in XMLP?

We use section:not property to handle NULL values.
Syntax:
<?if@section:not(//GROUP_NAME)?>Message/Action<?end if?>

Example:
<?if@section:not(//G_HEADERS)?>No Data Found for the Parameters Provided Master
<?end if?>

Displays the Message
" No Data Found for the Parameters Provided Master "

How to find String Length in XMLP?

Tag to find the length of the String

<?string-length(Field_NAME)?>
Example:
<Name>Samantha</Name> (i.e Data Source)
Template Layout:
<?Name?> is <?string-length(Name)?> characters length Word.
Output:
Samantha is 8 characters length Word.

How to use Variables in XMLP?

Here is a small example of how to use variables in XMLP rtf layout:

Declaring the Variable R
<?xdoxslt:get_variable($_XDOCTX, 'R')?>
Declaring the Variable R and Assigning the Values 4 to R
<?xdoxslt:set_variable($_XDOCTX, 'R', 4)?>
Get the Variable value
<?xdoxslt:get_variable($_XDOCTX, 'R')?>
This adds 5 to variable R and displays it
<?xdoxslt:set_variable($_XDOCTX, 'R', xdoxslt:get_variable($_XDOCTX, 'R')+5)?>
This subtracting 2 to varaible R and displays it
<?xdoxslt:set_variable($_XDOCTX, 'R', xdoxslt:get_variable($_XDOCTX, 'R')-2)?>
Similarly U can use any mathematical Operation to the variable R.

How to Sort Data in the XMLP?

Syntax:-

Case 1:-
<?sort:Field_Name?><?Field_Name?>
It sorts the data in Ascending order by Default.if Order By is not mentioned

Case 2:-
<?sort:Field_Name;'Order by'?>
It sorts the data based on Order By mentioned here like Ascending Or Descending

Case 3:-
<?sort:Field_Name;'Order by';data-type='text'?>
It sorts the String Data based on Order By mentioned here like Ascending Or Descending

Examples:-

Case 1:-

<?for-each:EMP_DETAILS?>
<?sort:EMP_NO?><?EMP_NAME?><?end for-each?>
Case 2:-
<?for-each:EMP_DETAILS?>
<?sort:EMP_NO;'descending'?><?EMP_NAME?><?end for-each?>
Case 3:-
<?for-each:EMP_DETAILS?>
<?sort:EMP_NAME;'ascending';data-type='text'?><?EMP_NAME?><?end for-each?>

How to repeat the Header of the template on each and every page of the output in XMLP?

Use <@section:?> tag to repeat a section in any specific region of a layout

(Or)You can also use header part of rtf (MS word feature) to insert the information you want to repeat on all pages.

What are the different ways to show/insert an image on XMLP layout?

There are 4 different ways to insert/show an image in rtf:

1. Insert an image directly on rtf just by using MS word standard feature
2. Using OA_MEDIA: storing on server and giving physical path of the image
3. Using URL of a image hosted on some website
4. Retrieving the image store in a database as BLOB type

Can XML publisher reports be used in OAF pages?

We can use **XDO template utility** helper java classes to embed XMLP report on OA Pages

Which component is responsible for generating the output in XML format before applying it to layout template?

DataEngine will take DataTemplate as the input and the output will be generated in xml format which will then be applied on layout template
More information: http://oracleappsdna.com/2014/10/developing-xml-publisher-reports-using-data-template/ [http://oracleappsdna.com/2014/10/developing-xml-publisher-reports-using-data-template/]

What is the default output format of XMLP report?

The default output format defined during the layout template creation will be used to generate the output, the same can be modified during the request submission and it will overwrite the one defined at layout template

Posted 3rd March by Krishnareddy

0 Add a comment

4th September 2012 SQL*LOADER WITH EXAMPLES

What is SQL*Loader and what is it used for?

SQL*Loader is a bulk loader utility used for moving data from external files into the Oracle database. Its syntax is similar to that of the DB2 Load utility, but comes with more options. SQL*Loader supports various load formats, selective loading, and multi-table loads.

How does one use the SQL*Loader utility?

One can load data into an Oracle database by using the sqlldr (sqlload on some platforms) utility. Invoke the utility without arguments to get a list of available parameters. Look at the following example:

```
sqlldr scott/tiger control=loader.ctl
```

This sample control file (loader.ctl) will load an external data file containing delimited data: load data

```
infile 'c:\data\mydata.csv'
into table emp ( empno, empname, sal, deptno )
fields terminated by "," optionally enclosed by '"'
```

The mydata.csv file may look like this:

```
10001,"Scott Tiger", 1000, 40
10002,"Frank Naude", 500, 20
```

Another Sample control file with in-line data formatted as fix length records. The trick is to specify "*" as the name of the data file, and use BEGINDATA to start the data section in the control file.

```
load data
infile *
replace
into table departments
( dept position (02:05) char(4),
deptname position (08:27) char(20) )
begindata
COSC COMPUTER SCIENCE
ENGL ENGLISH LITERATURE
MATH MATHEMATICS
POLY POLITICAL SCIENCE
```

Is there a SQL*Unloader to download data to a flat file?

Oracle does not supply any data unload utilities. However, you can use SQL*Plus to select and format your data and then spool it to a file:

```
set echo off newpage 0 space 0 pagesize 0 feed off head off trimspool on
spool oradata.txt
select col1 ',' col2 ',' col3
from tab1
where col2 = 'XYZ';
spool off
```

Alternatively use the UTL_FILE PL/SQL package:
Remember to update initSID.ora, utl_file_dir='c:\oradata' parameter

```
declare
fp utl_file.file_type;
begin
fp := utl_file.fopen('c:\oradata','tab1.txt','w');
utl_file.putf(fp, '%s, %s\n', 'TextField', 55);
utl_file.fclose(fp);
end;
/
```

You might also want to investigate third party tools like TOAD or ManageIT Fast Unloader from CA to help you unload data from Oracle.

Can one load variable and fix length data records?

Yes, look at the following control file examples. In the first we will load delimited data (variable length):

```
LOAD DATA
INFILE *
INTO TABLE load_delimited_data
FIELDS TERMINATED BY "," OPTIONALLY ENCLOSED BY '"'
TRAILING NULLCOLS
( data1, data2 )
BEGINDATA
11111,AAAAAAAAAA
22222,"A,B,C,D,"
```

If you need to load positional data (fixed length), look at the following control file example:

```
LOAD DATA
INFILE *
INTO TABLE load_positional_data
( data1 POSITION(1:5),
data2 POSITION(6:15) )
BEGINDATA
11111AAAAAAAAAA
22222BBBBBBBBBB
```

Can one skip header records load while loading?

Use the "SKIP n" keyword, where n = number of logical rows to skip. Look at this example:

```
LOAD DATA
INFILE *
INTO TABLE load_positional_data
SKIP 5
( data1 POSITION(1:5),
```

```
data2 POSITION(6:15) )
BEGINDATA
11111AAAAAAAAAA
22222BBBBBBBBBB
```

Can one modify data as it loads into the database?

Data can be modified as it loads into the Oracle Database. Note that this only applies for the conventional load path and not for direct path loads.

```
LOAD DATA
INFILE *
INTO TABLE modified_data
( rec_no "my_db_sequence.nextval",
region CONSTANT '31',
time_loaded "to_char(SYSDATE, 'HH24:MI')",
data1 POSITION(1:5) ":data1/100",
data2 POSITION(6:15) "upper(:data2)",
data3 POSITION(16:22)"to_date(:data3, 'YYMMDD')" )
BEGINDATA
11111AAAAAAAAAA991201
22222BBBBBBBBBB990112
```

```
LOAD DATA
INFILE 'mail_orders.txt'
BADFILE 'bad_orders.txt'
APPEND
INTO TABLE mailing_list
FIELDS TERMINATED BY ","
( addr,
city,
state,
zipcode,
mailing_addr "decode(:mailing_addr, null, :addr, :mailing_addr)",
mailing_city "decode(:mailing_city, null, :city, :mailing_city)",
mailing_state )
```

Can one load data into multiple tables at once?

Look at the following control file:

```
LOAD DATA
INFILE *
REPLACE
INTO TABLE emp
WHEN empno != ' '
( empno POSITION(1:4) INTEGER EXTERNAL,
ename POSITION(6:15) CHAR,
deptno POSITION(17:18) CHAR,
mgr POSITION(20:23) INTEGER EXTERNAL )
INTO TABLE proj
WHEN projno != ' '
( projno POSITION(25:27) INTEGER EXTERNAL,
empno POSITION(1:4) INTEGER EXTERNAL )
```

Can one selectively load only the records that one need?

Look at this example, (01) is the first character, (30:37) are characters 30 to 37:

```
LOAD DATA
INFILE 'mydata.dat'
BADFILE 'mydata.bad'
DISCARDFILE 'mydata.dis'
APPEND
INTO TABLE my_selective_table
WHEN (01) <> 'H' and (01) <> 'T' and (30:37) = '19991217'
( region CONSTANT '31',
service_key POSITION(01:11) INTEGER EXTERNAL,
call_b_no POSITION(12:29) CHAR )
```

Can one skip certain columns while loading data?

One cannot use POSTION(x:y) with delimited data. Luckily, from Oracle 8i one can specify FILLER columns. FILLER columns are used to skip columns/fields in the load file, ignoring fields that one does not want. Look at this example:

```
LOAD DATA
TRUNCATE
INTO TABLE T1
FIELDS TERMINATED BY ','
( field1,
field2 FILLER,
field3 )
```

How does one load multi-line records?

One can create one logical record from multiple physical records using one of the following two clauses:
CONCATENATE: - use when SQL*Loader should combine the same number of physical records together to form one logical record.
CONTINUEIF - use if a condition indicates that multiple records should be treated as one. Eg. by having a '#' character in column 1.

How can get SQL*Loader to COMMIT only at the end of the load file?

One cannot, but by setting the ROWS= parameter to a large value, committing can be reduced. Make sure you have big rollback segments ready when you use a high value for ROWS=.

Can one improve the performance of SQL*Loader?

A very simple but easily overlooked hint is not to have any indexes and/or constraints (primary key) on your load tables during the load process. This will significantly slow down load times even with ROWS= set to a high value.
Add the following option in the command line: DIRECT=TRUE. This will effectively bypass most of the RDBMS processing. However, there are cases when you can't use direct load. Refer to chapter 8 on Oracle server Utilities manual.
Turn off database logging by specifying the UNRECOVERABLE option. This option can only be used with direct data loads.
Run multiple load jobs concurrently.

What is the difference between the conventional and direct path loader?

The conventional path loader essentially loads the data by using standard INSERT

statements. The direct path loader (DIRECT=TRUE) bypasses much of the logic involved with that, and loads directly into the Oracle data files. More information about the restrictions of direct path loading can be obtained from the Utilities Users Guide.

Posted 4th September 2012 by Krishnareddy

Labels: SQL LOADER

1 View comments

3rd September 2012

Steps for Developing Custom Forms in E-Business Suite

Steps for Developing Custom Forms in E-Business Suite

Follow these steps to create custom forms in E-Business Suite:

1. **Create TEMPLATE form** Make a copy of TEMPLATE.fmb and rename it to your custom form name. Your form name will begin with XX. For developing HRMS-related screens, use HRTEMPLT.fmb.
2. **Develop form** Develop your form as per programming guidelines and standards in Oracle Applications Forms Development Guide. Detailed guidelines are available at

   [http://www.blogger.com/blogger.g?blogID=955765254185387334]

   http://download.oracle.com/docs/cd/B34956_01/current/acrobat/120devg.pdf
   [http://download.oracle.com/docs/cd/B34956_01/current/acrobat/120devg.pdf] .

3. **Generate runtime file** Transfer the FMB file to midtier and compile it to generate an FMX compiled file.
4. **Register** Register the form with Custom Application.
5. **Create form function** Create a form function attached to the form.
6. **Attach to menu** Attach the form function to the menu.
7. **Assign to responsibility** Assign the menu to the responsibility.

Navigate to Application Developer responsibility and go to Form and enter your form details as follows
Form: Form Name(.fmb name)

Application: Application(Provide Custom Application if you have one)

User Form name: This is the form name by which you call it (like Purchase Orders,Item Inquiry etc.,)

[http://4.bp.blogspot.com/-KF7AyNI8tYM/UEN-onfM33I/AAAAAAAACQs/7pLsmRoPsp8/s1600/1.png]

Now, create a Function to which your form should be attached. For this, navigate to Function in Application Developer Responsibility and provide the details.

[http://4.bp.blogspot.com/-gHImNaBHtNM/UEN-ooguLeI/AAAAAAAACQo/0Ia3t4mVLg8/s1600/2.png]

In the Form tab, provide the Form name which you gave in the previous screen where you created your new form

[http://4.bp.blogspot.com/-mBupVIA-KMc/UEN-olMeTWI/AAAAAAAACQk/olSCUvVoQ_A/s1600/3.png]

The last thing left is attaching our form to a menu. Attaching a form to menu is how we could be able to see our form in the desired responsibility.

To check the menu details for a particular responsibility, navigate to system administrator responsibility and query the responsibility you want attach.

[http://2.bp.blogspot.com/-oI8Dr4WjN2o/UEN-pWNivWI/AAAAAAAACQ4/SWG_bwNlUAs/s1600/4.png]

Now switch to application developer responsibility and attach your form to desired responsibility. Here, we are attaching to Inventory Responsibility.

[http://4.bp.blogspot.com/-K4QbxX9wFtw/UEN-pu9bIuI/AAAAAAAACQ0/HgVRD0VeZ4U/s1600/5.png]

Save the work and see your form in the responsibility which you have attached

[http://2.bp.blogspot.com/-x2tu4y6m-GQ/UEN-qIFbQqI/AAAAAAAACQ8/ccjkHtziOx4/s1600/6.png]

That's it, your Form is successfully registered in oracle apps. Now move the .fmb file to your custom top or the base top (INV_TOP in this case)

After moving the file , run the below command:
frmcmp_batch Module=ASPITEM.fmb Userid=apps/apps Output_file=ASPITEM.fmx Compile_all=special batch=Yes

Posted 3rd September 2012 by Krishnareddy

1 View comments

3rd September 2012

Oracle Apps Inventory / Purchasing / GL Accounting Periods

Oracle Apps Inventory / Purchasing / GL Accounting Periods

Whenever you do Receiving inventory transactions, you often get errors if your Inventory/ purchasing/ GL periods are not open.

Here is the navigation where you can setup the periods for your material receipts:

1. Ensure that the GL Period has Status=Open:
Setup > Financial > Accounting > Open and Close periods
-OR-
Setup > Open and Close periods
[http://www.blogger.com/blogger.g?blogID=955765254185387334]

2. Ensure the Purchasing Period is Open:
Setup > Financial > Accounting > Control Purchasing Period

3. Please note that the Inventory period must also be Open:
Accounting close cycle > Inventory Accounting Periods

Use the following Query to check the status of the same:

```
select a.period_name,
       a.period_num,
       a.gl_status,
       b.po_status,
       c.ap_status
from
   (select period_name, period_num,
    decode(closing_status,'O','Open',
                          'C','Closed',
                          'F','Future',
                          'N','Never',
           closing_status) gl_status
    from gl_period_statuses
    where application_id = 101
    and start_date >= '01-JAN-98
    and end_date < '01-JAN-99'
    and set_of_books_id = &&set_of_books_id) a,
   (select period_name,
    decode(closing_status,'O','Open',
                          'C','Closed',
                          'F','Future',
                          'N','Never',
           closing_status) po_status
    from gl_period_statuses
    where application_id = 201
    and start_date >= '01-JAN-98'
    and end_date < '01-JAN-99'
    and set_of_books_id = &&set_of_books_id) b,
   (select period_name,
    decode(closing_status,'O','Open',
                          'C','Closed',
                          'F','Future',
                          'N','Never',
           closing_status) ap_status
    from gl_period_statuses
    where application_id = 200
    and start_date >= '01-JAN-98'
```

```
    and end_date < '01-JAN-99'
    and set_of_books_id = &&set_of_books_id) c
where a.period_name = b.period_name
and   a.period_name = c.period_name
order by a.period_num
```

Posted 3rd September 2012 by Krishnareddy

1 View comments

29th August 2012

Script to allocate responsibilities and create users

Self Service HR Script 3 to allocate responsibilities and create users

```
set scan on ;
set scan off ;

CREATE OR REPLACE PACKAGE BODY xx_sshr_allocate_resp_pkg IS
--DO NOT RUN THIS WITHOUT CHANGING XXPRD
--REPLACE XXPRD BY RESULT OF BELOW SQL FROM PRODUCTION
--select instance_name from v$instance ;

 --Created in Nov 06 by Anil Passi
 /*
 When      By          Why
 ------------------------------------------------
 29Nov06     AnilPassi  To allocate responsibilities
 01Dec06     Anil Passi To create new users too
                  Send emails to new users with their password etc
                  Send emails to existing users that they now have sshr
 */
 g_instance_name          VARCHAR2(100) := 'JUNK';
 g_debug_procedure_context VARCHAR2(50);
 g_debug_header_context CONSTANT VARCHAR2(80) :=
'xxxx_sshr_allocate_resp_pkg.';

 PROCEDURE debug_begin_procedure IS
 BEGIN

   fnd_log.STRING(log_level => fnd_log.level_statement
             ,module    => g_debug_header_context ||
                       g_debug_procedure_context
             ,message   => 'Begin ' || g_debug_procedure_context);

   IF fnd_global.conc_request_id > 0 AND
     fnd_profile.VALUE('AFLOG_ENABLED') = 'Y'
```

```
 THEN
  fnd_file.put_line(which => fnd_file.log
                ,buff  => 'Begin ' || g_debug_procedure_context);
 END IF;

END debug_begin_procedure;

PROCEDURE debug_stmt(p_msg IN VARCHAR2) IS
BEGIN

 fnd_log.STRING(log_level => fnd_log.level_statement
             ,module    => g_debug_header_context ||
                           g_debug_procedure_context
             ,message   => p_msg);
 IF fnd_global.conc_request_id > 0
 THEN
  fnd_file.put_line(which => fnd_file.log
                ,buff  => p_msg);
 END IF;
END debug_stmt;

PROCEDURE debug_end_procedure IS
BEGIN
 fnd_log.STRING(log_level => fnd_log.level_statement
             ,module    => g_debug_header_context ||
                           g_debug_procedure_context
             ,message   => 'End ' || g_debug_procedure_context);
 IF fnd_global.conc_request_id > 0 AND
   fnd_profile.VALUE('AFLOG_ENABLED') = 'Y'
 THEN
  fnd_file.put_line(which => fnd_file.log
                ,buff  => 'End ' || g_debug_procedure_context);
 END IF;

END debug_end_procedure;

PROCEDURE set_debug_context(p_procedure_name IN VARCHAR2) IS
BEGIN
 g_debug_procedure_context := p_procedure_name;
 debug_begin_procedure;
END set_debug_context;

FUNCTION is_user_creation_possible(p_person_id IN INTEGER
                           ,p_xxdp      OUT xx_windows_logon_table%ROWTYPE)
 RETURN VARCHAR2 IS
 CURSOR c_check IS
  SELECT xxdp.*
  FROM per_people_x ppx, xx_windows_logon_table xxdp
  WHERE ltrim(ppx.employee_number
            ,'0') = ltrim(xxdp.emp_no
                     ,'0')
```

```
    AND ppx.person_id = p_person_id;
  p_check c_check%ROWTYPE;
BEGIN
  OPEN c_check;
  FETCH c_check
    INTO p_check;
  CLOSE c_check;
  p_xxdp := p_check;
  IF p_check.emp_no IS NULL
  THEN
    RETURN 'No emp_no record in Network Login Table';
  ELSIF p_check.nt_login IS NULL
  THEN
    RETURN 'No NT Login Available for this Person in Network Login Table';
  ELSIF p_check.college_email_address IS NULL
  THEN
    RETURN 'No Email Address for this Person in Network Login Table';
  END IF;
  RETURN NULL;
END is_user_creation_possible;

FUNCTION get_email_from_emp_no(p_emp_no_email  IN VARCHAR2
                    ,p_test_email IN VARCHAR2) RETURN VARCHAR2 IS
BEGIN

  IF g_instance_name = 'XXPRD'
  THEN

    RETURN p_emp_no_email;
  ELSE
    RETURN p_test_email;
  END IF;
END get_email_from_emp_no;

FUNCTION does_fu_exist(p_fu_name IN VARCHAR2) RETURN BOOLEAN IS
  CURSOR c_check IS
    SELECT 'x' FROM fnd_user fu WHERE fu.user_name = upper(p_fu_name);

  p_check c_check%ROWTYPE;
BEGIN
  OPEN c_check;
  FETCH c_check
    INTO p_check;
  IF c_check%FOUND
  THEN
    CLOSE c_check;
    RETURN TRUE;
  END IF;
  CLOSE c_check;
  RETURN FALSE;
END does_fu_exist;
```

```
  PROCEDURE send_email_to_new_user(p_xxdp       IN
xx_windows_logon_table%ROWTYPE
                          ,p_user_name  IN VARCHAR2
                          ,p_password   IN VARCHAR2
                          ,p_test_email IN VARCHAR2) IS
  BEGIN
    DECLARE
    BEGIN
      send_html_email(p_to           =>
get_email_from_emp_no(p_xxdp.college_email_address
                                          ,p_test_email)
                 ,p_from         => nvl(p_test_email
                                  ,'xxmail@gmail.com')
                 ,p_subject      => 'Welcome to Self Service HR'
                 ,p_text         => 'Welcome to Self Service HR'
                 ,p_html         => '<b>Your User Name</b> : ' ||
                                 p_user_name ||
                                 '<br/><b>Your Password</b> : ' ||
                                 p_password || '<br/><br/>' ||
                                 'This user name and password gives you access the new Self
Service HR.' ||
                                 '<br/><br/>Self Service HR enables Company staff to view and
update their own personal data. The <br/>information is current and any changes made
will be implemented immediately.' ||
                                 '<br/><br/>Please go to Spectrum following this link <br/><a
href="http://anilpassi.com">http://anilpassi.com</a>' ||
                                 '<br/>where you can log into Self Service HR, find out more and
read the FAQs.'
                 ,p_smtp_hostname => 'localhost'
                 ,p_smtp_portnum  => '25');

    END;
  END send_email_to_new_user;

  PROCEDURE send_email_to_existing_user(p_xxdp       IN
xx_windows_logon_table%ROWTYPE
                                 ,p_test_email IN VARCHAR2) IS
  BEGIN
    DECLARE
    BEGIN
      send_html_email(p_to           =>
get_email_from_emp_no(p_xxdp.college_email_address
                                          ,p_test_email)
                 ,p_from         => nvl(p_test_email
                                  ,'xxmail@gmail.com')
                 ,p_subject      => 'Welcome to Self Service HR'
                 ,p_text         => 'Welcome to Self Service HR'
                 ,p_html         => 'We are writing to let you know that the next time you log
into Oracle Apps you will see a new<br/> responsibility, <b>XX HR Employee Self
Service</b>. This responsibility gives you access the new<br/> Self Service HR feature in
```

```
Oracle Apps.' ||
                                '<p></p>Self Service HR enables staff to view and update their
own personal data.' ||
                                '<p></p>Please go to this link<br/><a
href="http://anilpassi.com">http://anilpassi.com</a><br/>to find out more and read the
FAQs.' ||
                                '<br/><br/>' || 'Regards' ||
                                '<br/><br/>SSHR Rollout Team' ||
                                '<br/>' ||
                                'HR Dept'
                    ,p_smtp_hostname => 'localhost'
                    ,p_smtp_portnum  => '25');

    END;
  END send_email_to_existing_user;

  FUNCTION get_latest_fu(p_proposed_fu_name IN VARCHAR2
                        ,p_proposed_offset  IN INTEGER) RETURN VARCHAR2 IS
  BEGIN
    IF does_fu_exist(p_proposed_fu_name || p_proposed_offset)
    THEN
      RETURN get_latest_fu(p_proposed_fu_name
                          ,p_proposed_offset + 1);
    END IF;

    RETURN upper(p_proposed_fu_name || p_proposed_offset);
  END get_latest_fu;

  FUNCTION get_fu_name(p_nt_login IN VARCHAR2) RETURN VARCHAR2 IS
  BEGIN
    IF NOT does_fu_exist(p_nt_login)
    THEN
      RETURN upper(p_nt_login);
    END IF;

    IF NOT does_fu_exist(p_nt_login)
    THEN
      RETURN upper(p_nt_login);
    END IF;

    RETURN get_latest_fu(p_nt_login
                        ,1);
  END get_fu_name;

  FUNCTION get_user_name_from_fu_per_id(p_person_id IN VARCHAR2)
    RETURN VARCHAR2 IS
    CURSOR c_get IS
      SELECT fu.user_name
      FROM fnd_user fu
      WHERE fu.employee_id = p_person_id;
```

```
  p_get c_get%ROWTYPE;
BEGIN
  OPEN c_get;

  FETCH c_get
    INTO p_get;

  CLOSE c_get;

  RETURN p_get.user_name;
END get_user_name_from_fu_per_id;

FUNCTION get_random_password RETURN VARCHAR2 IS
BEGIN
  RETURN lower(dbms_random.STRING('X'
                                  ,8));
END get_random_password;

FUNCTION get_person_name(p_person_id IN INTEGER) RETURN VARCHAR2 IS
  CURSOR c_get IS
    SELECT full_name
    FROM per_all_people_f
    WHERE person_id = p_person_id
    ORDER BY effective_start_date DESC;
  p_get c_get%ROWTYPE;

BEGIN
  OPEN c_get;
  FETCH c_get
    INTO p_get;
  CLOSE c_get;
  RETURN p_get.full_name;
END get_person_name;

PROCEDURE create_fnd_user_for_emp_no(p_user_name         IN VARCHAR2
                                  ,p_person_id          IN INTEGER
                                  ,p_email_address      IN VARCHAR2
                                  ,p_person_description IN VARCHAR2
                                  ,p_password           OUT VARCHAR2) IS
  v_session_id VARCHAR2(200);
  v_password   VARCHAR2(100) := get_random_password;
BEGIN
  p_password := v_password;
  fnd_user_pkg.createuser(x_user_name             => p_user_name
                        ,x_owner                  => ''
                        ,x_unencrypted_password   => v_password
                        ,x_description            => p_person_description
                        ,x_password_lifespan_days => 180
                        ,x_employee_id            => p_person_id
                        ,x_email_address          => p_email_address);
```

```
END create_fnd_user_for_emp_no;

FUNCTION get_fu_id(p_fu_name IN VARCHAR2) RETURN VARCHAR2 IS
  CURSOR c_get IS
    SELECT user_id FROM fnd_user WHERE user_name = p_fu_name;
  p_get c_get%ROWTYPE;
BEGIN
  OPEN c_get;
  FETCH c_get
    INTO p_get;
  CLOSE c_get;
  RETURN p_get.user_id;
END get_fu_id;

FUNCTION create_fnd_user(p_person_id             IN INTEGER
                        ,p_xxdp                  IN xx_windows_logon_table%ROWTYPE
                        ,p_new_fnd_user_name     OUT VARCHAR2
                        ,p_new_fnd_user_password OUT VARCHAR2)
  RETURN INTEGER IS
  v_user_name fnd_user.user_name%TYPE;
  v_password  VARCHAR2(200);
  v_err       VARCHAR2(2000);
BEGIN
  v_user_name := get_fu_name(p_nt_login => p_xxdp.nt_login);
  debug_stmt('For p_xxdp.nt_login=>' || p_xxdp.nt_login ||
             ' the username is ' || v_user_name);
  create_fnd_user_for_emp_no(p_user_name          => p_xxdp.nt_login
                        ,p_person_id          => p_person_id
                        ,p_email_address      => p_xxdp.college_email_address
                        ,p_person_description => p_xxdp.title || ' ' ||
                                                 p_xxdp.first_name || ' ' ||
                                                 p_xxdp.last_name
                        ,p_password           => v_password);
  p_new_fnd_user_name     := v_user_name;
  p_new_fnd_user_password := v_password;
  RETURN get_fu_id(p_fu_name => v_user_name);
EXCEPTION
  WHEN OTHERS THEN
    v_err := substr(SQLERRM
                   ,1
                   ,2000);
    debug_stmt(v_err);
    RETURN NULL;
END create_fnd_user;

PROCEDURE send_html_email(p_to           IN VARCHAR2
                        ,p_from          IN VARCHAR2
                        ,p_subject       IN VARCHAR2
                        ,p_text          IN VARCHAR2 DEFAULT NULL
                        ,p_html          IN VARCHAR2 DEFAULT NULL
                        ,p_smtp_hostname IN VARCHAR2
```

```
                    ,p_smtp_portnum  IN VARCHAR2) IS
  l_boundary   VARCHAR2(255) DEFAULT 'a1b2c3d4e3f2g1';
  l_connection utl_smtp.connection;
  l_body_html  CLOB := empty_clob; --This LOB will be the email message
  l_offset     NUMBER;
  l_ammount    NUMBER;
  l_temp       VARCHAR2(32767) DEFAULT NULL;
 BEGIN
  /* Usage.....
     html_email(p_to          => 'a.passi@Company.ac.uk'
           ,p_from          => 'anilpassi@gmail.com'
           ,p_subject       => 'Testing from anil'
           ,p_text          => 'ABCD'
           ,p_html          => '<b>IJKLM</b> Testing for the HTML Format of the email'
           ,p_smtp_hostname => 'localhost'
           ,p_smtp_portnum  => '25');
  */
  l_connection := utl_smtp.open_connection(p_smtp_hostname
                                  ,p_smtp_portnum);
  utl_smtp.helo(l_connection
           ,p_smtp_hostname);
  utl_smtp.mail(l_connection
           ,p_from);
  utl_smtp.rcpt(l_connection
           ,p_to);

  l_temp := l_temp || 'MIME-Version: 1.0' || chr(13) || chr(10);
  l_temp := l_temp || 'To: ' || p_to || chr(13) || chr(10);
  l_temp := l_temp || 'From: ' || p_from || chr(13) || chr(10);
  l_temp := l_temp || 'Subject: ' || p_subject || chr(13) || chr(10);
  l_temp := l_temp || 'Reply-To: ' || p_from || chr(13) || chr(10);
  l_temp := l_temp || 'Content-Type: multipart/alternative; boundary=' ||
         chr(34) || l_boundary || chr(34) || chr(13) || chr(10);

  ----------------------------------------------------
  -- Write the headers
  dbms_lob.createtemporary(l_body_html
                    ,FALSE
                    ,10);
  dbms_lob.WRITE(l_body_html
            ,length(l_temp)
            ,1
            ,l_temp);

  ----------------------------------------------------
  -- Write the text boundary
  l_offset := dbms_lob.getlength(l_body_html) + 1;
  l_temp   := '--' || l_boundary || chr(13) || chr(10);
  l_temp   := l_temp || 'content-type: text/plain; charset=us-ascii' ||
           chr(13) || chr(10) || chr(13) || chr(10);
  dbms_lob.WRITE(l_body_html
```

```
              ,length(l_temp)
              ,l_offset
              ,l_temp);

-------------------------------------------------
-- Write the plain text portion of the email
l_offset := dbms_lob.getlength(l_body_html) + 1;
dbms_lob.WRITE(l_body_html
              ,length(p_text)
              ,l_offset
              ,p_text);

-------------------------------------------------
-- Write the HTML boundary
l_temp   := chr(13) || chr(10) || chr(13) || chr(10) || '--' ||
           l_boundary || chr(13) || chr(10);
l_temp   := l_temp || 'content-type: text/html;' || chr(13) || chr(10) ||
           chr(13) || chr(10);
l_offset := dbms_lob.getlength(l_body_html) + 1;
dbms_lob.WRITE(l_body_html
              ,length(l_temp)
              ,l_offset
              ,l_temp);

-------------------------------------------------
-- Write the HTML portion of the message
l_offset := dbms_lob.getlength(l_body_html) + 1;
dbms_lob.WRITE(l_body_html
              ,length(p_html)
              ,l_offset
              ,p_html);

-------------------------------------------------
-- Write the final html boundary
l_temp   := chr(13) || chr(10) || '--' || l_boundary || '--' || chr(13);
l_offset := dbms_lob.getlength(l_body_html) + 1;
dbms_lob.WRITE(l_body_html
              ,length(l_temp)
              ,l_offset
              ,l_temp);

-------------------------------------------------
-- Send the email in 1900 byte chunks to UTL_SMTP
l_offset  := 1;
l_ammount := 1900;
utl_smtp.open_data(l_connection);
WHILE l_offset < dbms_lob.getlength(l_body_html)
LOOP
  utl_smtp.write_data(l_connection
                     ,dbms_lob.substr(l_body_html
                                     ,l_ammount
```

```
                                ,l_offset));
    l_offset  := l_offset + l_ammount;
    l_ammount := least(1900
                       ,dbms_lob.getlength(l_body_html) - l_ammount);
  END LOOP;
  utl_smtp.close_data(l_connection);
  utl_smtp.quit(l_connection);
  dbms_lob.freetemporary(l_body_html);
END send_html_email;

PROCEDURE excel_output(p_msg IN VARCHAR2) IS
BEGIN
  fnd_file.put_line(fnd_file.output
                    ,p_msg);
END excel_output;

FUNCTION get_user_name(p_user_id IN INTEGER) RETURN VARCHAR2 IS
  CURSOR c_get IS
    SELECT user_name FROM fnd_user WHERE user_id = p_user_id;
  p_get c_get%ROWTYPE;
BEGIN
  OPEN c_get;
  FETCH c_get
    INTO p_get;
  CLOSE c_get;
  RETURN p_get.user_name;
END get_user_name;

FUNCTION get_emp_no(p_person_id IN INTEGER) RETURN VARCHAR2 IS
  CURSOR c_get IS
    SELECT employee_number
    FROM xx_per_all_people_x
    WHERE person_id = p_person_id;
  p_get c_get%ROWTYPE;
BEGIN
  OPEN c_get;
  FETCH c_get
    INTO p_get;
  CLOSE c_get;
  RETURN p_get.employee_number;
END get_emp_no;

FUNCTION get_cost_centre_group(p_person_id IN INTEGER) RETURN VARCHAR2 IS
  CURSOR c_get IS
    SELECT hrou1.attribute5
    FROM hr_all_organization_units hrou1
       ,hr_all_organization_units hrou
       ,xx_per_all_asg_x         ass
       ,xx_per_all_people_x      ppx
    WHERE ppx.person_id = p_person_id
    AND ass.person_id = ppx.person_id
```

```
    AND ass.assignment_number IS NOT NULL
    AND ass.primary_flag = 'Y'
    AND hrou.organization_id = ass.organization_id
    AND hrou1.NAME = primaryhro_pkg.fn_get_primaryhro(ass.organization_id);
  p_get c_get%ROWTYPE;
BEGIN
  OPEN c_get;
  FETCH c_get
    INTO p_get;
  CLOSE c_get;
  RETURN p_get.attribute5;
END get_cost_centre_group;

FUNCTION get_parent_org(p_person_id IN INTEGER) RETURN VARCHAR2 IS
  CURSOR c_get IS
    SELECT primaryhro_pkg.fn_get_primaryhro(ass.organization_id) parent_org
    FROM hr_all_organization_units hrou
       ,xx_per_all_asg_x         ass
       ,xx_per_all_people_x      ppx
    WHERE ppx.person_id = p_person_id
    AND ass.person_id = ppx.person_id
    AND ass.assignment_number IS NOT NULL
    AND ass.primary_flag = 'Y'
    AND hrou.organization_id = ass.organization_id;
  p_get c_get%ROWTYPE;
BEGIN
  OPEN c_get;
  FETCH c_get
    INTO p_get;
  CLOSE c_get;
  RETURN p_get.parent_org;
END get_parent_org;

FUNCTION get_grade(p_person_id IN INTEGER) RETURN VARCHAR2 IS
  CURSOR c_get IS
    SELECT pg.NAME
    FROM per_grade_definitions pgd
       ,per_grades           pg
       ,xx_per_all_asg_x     ass
       ,xx_per_all_people_x  ppx
    WHERE ppx.person_id = p_person_id
    AND ass.person_id = ppx.person_id
    AND ass.assignment_number IS NOT NULL
    AND ass.primary_flag = 'Y'
    AND pg.grade_id = ass.grade_id
    AND pgd.grade_definition_id = pg.grade_definition_id;
  p_get c_get%ROWTYPE;
BEGIN
  OPEN c_get;
  FETCH c_get
    INTO p_get;
```

```
    CLOSE c_get;
    RETURN p_get.NAME;
  END get_grade;

  PROCEDURE run(errbuf                    OUT VARCHAR2
              ,retcode                    OUT VARCHAR2
              ,p_responsibility_name      IN VARCHAR2
              ,p_person_type              IN VARCHAR2
              ,p_cost_centre_group_1      IN VARCHAR2
              ,p_cost_centre_group_2      IN VARCHAR2
              ,p_parent_org_1             IN VARCHAR2
              ,p_parent_org_2             IN VARCHAR2
              ,p_emp_no                   IN VARCHAR2
              ,p_read_only_flag           IN VARCHAR2
              ,p_test_ceration_email_address IN VARCHAR2) IS
    n_count   INTEGER;
    v_sqlerrm VARCHAR2(2000);
    can_not_fnd_create_user EXCEPTION;
    error_in_fnd_user_pkg EXCEPTION;
    v_fnd_user_name VARCHAR2(100);
    CURSOR c_get IS
      SELECT *
      FROM fnd_responsibility_vl
      WHERE responsibility_name =
          nvl('XX HR Employee Self Service'
            ,p_responsibility_name);

    p_get c_get%ROWTYPE;

    duplicate_responsibility EXCEPTION;
    PRAGMA EXCEPTION_INIT(duplicate_responsibility
                        ,-20001);
    v_hard_password     VARCHAR2(1) :=
  fnd_profile.VALUE('SIGNON_PASSWORD_HARD_TO_GUESS');
    l_xxdp              xx_windows_logon_table%ROWTYPE;
    b_new_user_created  BOOLEAN;
    v_fnd_user_password VARCHAR2(100);
  BEGIN
    set_debug_context('run');

    SELECT instance_name INTO g_instance_name FROM v$instance;
    debug_stmt('g_instance_name=>' || g_instance_name);
    fnd_profile.put(NAME => 'SIGNON_PASSWORD_HARD_TO_GUESS'
                  ,val  => 'N');

    OPEN c_get;
    FETCH c_get
      INTO p_get;
    CLOSE c_get;

    --lets dump the records first into the temp table,
```

```
        --this will be followed by
        --a. see which people do not have Sign On
        --b. Which people already have Responsibility
        INSERT INTO xx_sshr_allocate_resp
          (sshr_allocate_resp_id
          ,person_id
          ,future_dated_employee_flag
          ,responsibillity_name
          ,error_during_resp_allocation
          ,fnd_user_id
          ,fnd_request_id
          ,email_address)
          (SELECT xx_sshr_allocate_resp_s.NEXTVAL
                ,ppx.person_id --PERSON_ID
                ,'N' --FUTURE_DATED_EMPLOYEE_FLAG
                ,p_responsibility_name --responsibillity_name
                ,NULL --ERROR_DURING_RESP_ALLOCATION
                ,NULL --FND_USER_ID
                ,fnd_global.conc_request_id --FND_REQUEST_ID
                ,ppx.email_address
           FROM per_person_types         ppt
              ,per_person_type_usages_x pptux
              ,xx_per_all_people_x      ppx
           WHERE ppx.person_id = pptux.person_id
           AND ppt.person_type_id = pptux.person_type_id
           AND ppx.employee_number = nvl(p_emp_no
                                 ,ppx.employee_number)
           AND ppt.system_person_type = 'EMP'
           AND ppt.user_person_type = p_person_type
           AND ppt.business_group_id =
               fnd_profile.VALUE('PER_BUSINESS_GROUP_ID')
           AND EXISTS (SELECT 'x'
                 FROM hr_all_organization_units hrou1
                    ,hr_all_organization_units hrou
                    ,xx_per_all_asg_x          pax
                 WHERE p_cost_centre_group_1 IS NOT NULL
                 AND pax.person_id = ppx.person_id
                 AND pax.primary_flag = 'Y'
                 AND pax.assignment_number IS NOT NULL
                 AND hrou.organization_id = pax.organization_id
                 AND hrou1.NAME =
                     primaryhro_pkg.fn_get_primaryhro(pax.organization_id)
                 AND hrou1.attribute5 IN
                     (nvl(p_cost_centre_group_1
                      ,'XXXX'), nvl(p_cost_centre_group_2
                               ,'XXXX'))
                 UNION ALL
                 SELECT 'x'
                 FROM dual
                 WHERE (p_cost_centre_group_1 IS NULL AND p_cost_centre_group_2 IS
    NULL))
```

```
   AND EXISTS
    (SELECT 'x'
         FROM hr_all_organization_units hrou, xx_per_all_asg_x pax
         WHERE p_parent_org_1 IS NOT NULL
         AND pax.person_id = ppx.person_id
         AND pax.primary_flag = 'Y'
         AND pax.assignment_number IS NOT NULL
         AND hrou.organization_id = pax.organization_id
         AND primaryhro_pkg.fn_get_primaryhro(pax.organization_id) IN
              (nvl(p_parent_org_1
               ,'XXXX'), nvl(p_parent_org_2
                       ,'XXXX'))
         UNION ALL
         SELECT 'x'
         FROM dual
         WHERE (p_parent_org_1 IS NULL AND p_parent_org_2 IS NULL)));
n_count := SQL%ROWCOUNT;

debug_stmt(n_count ||
          ' Records inserted into Temp Table based on Eligibility Criteria');

INSERT INTO xx_sshr_allocate_resp
  (sshr_allocate_resp_id
  ,person_id
  ,future_dated_employee_flag
  ,responsibillity_name
  ,error_during_resp_allocation
  ,fnd_user_id
  ,fnd_request_id
  ,email_address)
  (SELECT xx_sshr_allocate_resp_s.NEXTVAL
        ,ppx.person_id --PERSON_ID
        ,'Y' --FUTURE_DATED_EMPLOYEE_FLAG
        ,p_responsibility_name --responsibillity_name
        ,'Employee Is a Future Starter' --ERROR_DURING_RESP_ALLOCATION
        ,NULL --FND_USER_ID
        ,fnd_global.conc_request_id --FND_REQUEST_ID
        ,ppx.email_address
   FROM per_person_types              ppt
      ,xx_per_person_type_usages_eot pptux
      ,xx_per_all_people_eot         ppx
   WHERE NOT EXISTS
    (SELECT 'x'
         FROM xx_sshr_allocate_resp iar
         WHERE iar.person_id = ppx.person_id
         AND fnd_request_id = fnd_global.conc_request_id)
   AND ppx.person_id = pptux.person_id
   AND ppt.person_type_id = pptux.person_type_id
   AND ppx.employee_number = nvl(p_emp_no
                        ,ppx.employee_number)
   AND ppt.system_person_type = 'EMP'
```

```
            AND ppt.user_person_type = p_person_type
            AND ppt.business_group_id =
                fnd_profile.VALUE('PER_BUSINESS_GROUP_ID')
            AND EXISTS (SELECT 'x'
                  FROM hr_all_organization_units hrou1
                     ,hr_all_organization_units hrou
                     ,xx_per_all_asg_x          pax
                  WHERE p_cost_centre_group_1 IS NOT NULL
                  AND pax.person_id = ppx.person_id
                  AND pax.primary_flag = 'Y'
                  AND pax.assignment_number IS NOT NULL
                  AND hrou.organization_id = pax.organization_id
                  AND hrou1.NAME =
                      primaryhro_pkg.fn_get_primaryhro(pax.organization_id)
                  AND hrou1.attribute5 IN
                      (nvl(p_cost_centre_group_1
                       ,'XXXX'), nvl(p_cost_centre_group_2
                              ,'XXXX'))
                  UNION ALL
                  SELECT 'x'
                  FROM dual
                  WHERE (p_cost_centre_group_1 IS NULL AND p_cost_centre_group_2 IS
NULL))
            AND EXISTS
             (SELECT 'x'
                  FROM hr_all_organization_units hrou, xx_per_all_asg_x pax
                  WHERE p_parent_org_1 IS NOT NULL
                  AND pax.person_id = ppx.person_id
                  AND pax.primary_flag = 'Y'
                  AND pax.assignment_number IS NOT NULL
                  AND hrou.organization_id = pax.organization_id
                  AND primaryhro_pkg.fn_get_primaryhro(pax.organization_id) IN
                      (nvl(p_parent_org_1
                       ,'XXXX'), nvl(p_parent_org_2
                              ,'XXXX'))
                  UNION ALL
                  SELECT 'x'
                  FROM dual
                  WHERE (p_parent_org_1 IS NULL AND p_parent_org_2 IS NULL)));
         n_count := SQL%ROWCOUNT;
         debug_stmt(n_count ||
                    ' Records inserted into Temp Table that aer eligible but Future Dated');
         --Commenting the below, as we need to create User Accounts for these folks
         /*  UPDATE xx_sshr_allocate_resp isar
            SET error_during_resp_allocation = 'Employee Is Not a User'
            WHERE isar.fnd_request_id = fnd_global.conc_request_id
            AND error_during_resp_allocation IS NULL
            AND NOT EXISTS
             (SELECT 'x' FROM fnd_user fu WHERE fu.employee_id = isar.person_id);
            n_count := SQL%ROWCOUNT;
            put_log(n_count || ' Records errored due to them not being Employee');
```

```
*/
UPDATE xx_sshr_allocate_resp isar
SET fnd_user_id = (SELECT user_id
                   FROM fnd_user
                   WHERE employee_id = isar.person_id
                   AND rownum < 2)
WHERE isar.fnd_request_id = fnd_global.conc_request_id
AND error_during_resp_allocation IS NULL;

UPDATE xx_sshr_allocate_resp isar
SET responsibility_alloc_date = (SELECT start_date
                                 FROM fnd_user_resp_groups_direct
                                 WHERE user_id = isar.fnd_user_id
                                 AND responsibility_id =
                                     p_get.responsibility_id
                                 AND rownum < 2)
WHERE isar.fnd_request_id = fnd_global.conc_request_id;

n_count := SQL%ROWCOUNT;
debug_stmt(n_count ||
        ' Records were attempted to be assigned existing responsibility_alloc_date');

UPDATE xx_sshr_allocate_resp isar
SET error_during_resp_allocation = 'Responsibility Already Allocated on ' ||
                                   to_char(responsibility_alloc_date
                                       ,'DD-MON-YYYY')
WHERE isar.fnd_request_id = fnd_global.conc_request_id
AND responsibility_alloc_date IS NOT NULL;
n_count := SQL%ROWCOUNT;
debug_stmt(n_count ||
        ' Records errored as they already have the responsibility');

/*   UPDATE xx_sshr_allocate_resp isar
   SET error_during_resp_allocation = 'Employees User Record is Terminated'
   WHERE isar.fnd_request_id = fnd_global.conc_request_id
   AND error_during_resp_allocation IS NULL
   AND EXISTS (SELECT 'x'
         FROM fnd_user fu
         WHERE fu.employee_id = isar.person_id
         AND NOT (trunc(SYSDATE) BETWEEN
               nvl(fu.start_date
                  ,trunc(SYSDATE)) AND
               nvl(fu.start_date
                  ,trunc(SYSDATE))));
   n_count := SQL%ROWCOUNT;
   put_log(n_count || ' Records errored as their FND_USER is end dated');
*/

UPDATE xx_sshr_allocate_resp isar
SET error_during_resp_allocation = 'No Email Address'
WHERE isar.fnd_request_id = fnd_global.conc_request_id
```

```
AND isar.email_address IS NULL
AND error_during_resp_allocation IS NULL;
n_count := SQL%ROWCOUNT;
debug_stmt(n_count ||
        ' Records errored as they have no email address in HRMS');

UPDATE xx_sshr_allocate_resp isar
SET fnd_user_id = (SELECT user_id
                FROM fnd_user
                WHERE employee_id = isar.person_id
                AND rownum < 2)
WHERE isar.fnd_request_id = fnd_global.conc_request_id
AND error_during_resp_allocation IS NULL;

n_count := SQL%ROWCOUNT;
debug_stmt(n_count ||
        ' Records aer unerrored, and hence will be processed further');

excel_output('Action' || chr(9) || 'UserName' || chr(9) || 'emp_no' ||
           chr(9) || 'Person Full Name' || chr(9) ||
           'Allocation Date' || chr(9) || 'Error' || chr(9) ||
           'cost_centre_group' || chr(9) || 'parent_org' || chr(9) || 'Grade');

FOR p_rec IN (SELECT *
            FROM xx_sshr_allocate_resp isar
            WHERE isar.fnd_request_id = fnd_global.conc_request_id
            AND error_during_resp_allocation IS NULL)
LOOP
 BEGIN
   l_xxdp             := NULL;
   v_fnd_user_password := NULL;
   b_new_user_created  := FALSE;
   v_fnd_user_name     := NULL;
   v_sqlerrm          := is_user_creation_possible(p_person_id => p_rec.person_id
                                        ,p_xxdp      => l_xxdp);
   debug_stmt('p_rec.fnd_user_id =>' || p_rec.fnd_user_id);
   debug_stmt('Is user creation possible returned => ' || v_sqlerrm);
   IF p_rec.fnd_user_id IS NULL AND v_sqlerrm IS NOT NULL
   THEN

    RAISE can_not_fnd_create_user;
   END IF;

   IF NOT (p_read_only_flag = 'Y')
   THEN
    debug_stmt('Not read only');
    IF p_rec.fnd_user_id IS NULL
    THEN
     debug_stmt('Looks like new user is needed');

     p_rec.fnd_user_id := create_fnd_user(p_person_id            => p_rec.person_id
```

```
                                    ,p_xxdp                => l_xxdp
                                    ,p_new_fnd_user_name     => v_fnd_user_name
                                    ,p_new_fnd_user_password => v_fnd_user_password);
              IF p_rec.fnd_user_id IS NULL
              THEN
                RAISE error_in_fnd_user_pkg;
              ELSE
                UPDATE xx_sshr_allocate_resp ir
                SET ir.fnd_user_id    = p_rec.fnd_user_id
                  ,new_fnd_user_flag = 'Y'
                  ,messsage_code     = v_fnd_user_password
                WHERE ir.sshr_allocate_resp_id = p_rec.sshr_allocate_resp_id;
                b_new_user_created := TRUE;
              END IF;
            END IF;
            fnd_user_resp_groups_api.insert_assignment(user_id                    =>
p_rec.fnd_user_id
                                    ,responsibility_id            => p_get.responsibility_id
                                    ,responsibility_application_id => p_get.application_id
                                    ,security_group_id            => 0
                                    ,start_date                   => trunc(SYSDATE)
                                    ,end_date                     => NULL
                                    ,description                  => 'Auto Allocation for SSHR');
            UPDATE xx_sshr_allocate_resp
            SET responsibility_alloc_date = SYSDATE
            WHERE sshr_allocate_resp_id = p_rec.sshr_allocate_resp_id;
            IF b_new_user_created
            THEN
              excel_output('Allocated[With New User]' || chr(9) ||
                        get_user_name(p_rec.fnd_user_id) || chr(9) ||
                        get_emp_no(p_rec.person_id) || chr(9) ||
                        get_person_name(p_rec.person_id) || chr(9) ||
                        to_char(trunc(SYSDATE)
                              ,'DD-MON-YYYY') || chr(9) || '' || chr(9) ||
                        get_cost_centre_group(p_rec.person_id) || chr(9) ||
                        get_parent_org(p_rec.person_id) || chr(9) ||
                        get_grade(p_rec.person_id));
              send_email_to_new_user(p_xxdp       => l_xxdp
                                ,p_user_name  => v_fnd_user_name
                                ,p_password   => v_fnd_user_password
                                ,p_test_email => p_test_ceration_email_address);
            ELSE
              excel_output('Allocated' || chr(9) ||
                        get_user_name(p_rec.fnd_user_id) || chr(9) ||
                        get_emp_no(p_rec.person_id) || chr(9) ||
                        get_person_name(p_rec.person_id) || chr(9) ||
                        to_char(trunc(SYSDATE)
                              ,'DD-MON-YYYY') || chr(9) || '' || chr(9) ||
                        get_cost_centre_group(p_rec.person_id) || chr(9) ||
                        get_parent_org(p_rec.person_id) || chr(9) ||
                        get_grade(p_rec.person_id));
```

```
            send_email_to_existing_user(p_xxdp       => l_xxdp
                                       ,p_test_email => p_test_ceration_email_address);
          END IF;
          COMMIT;
        ELSE
          IF p_rec.fnd_user_id IS NULL
          THEN
            excel_output('Eligible [New User Will Be Created]' || chr(9) ||
                         nvl(get_user_name(p_rec.fnd_user_id)
                           ,get_fu_name(l_xxdp.nt_login)) || chr(9) ||
                         get_emp_no(p_rec.person_id) || chr(9) ||
                         get_person_name(p_rec.person_id) || chr(9) ||
                         chr(9) || chr(9) || get_cost_centre_group(p_rec.person_id) ||
                         chr(9) || get_parent_org(p_rec.person_id) || chr(9) ||
                         get_grade(p_rec.person_id));
          ELSE
            excel_output('Eligible' || chr(9) ||
                         get_user_name(p_rec.fnd_user_id) || chr(9) ||
                         get_emp_no(p_rec.person_id) || chr(9) ||
                         get_person_name(p_rec.person_id) || chr(9) ||
                         chr(9) || chr(9) || get_cost_centre_group(p_rec.person_id) ||
                         chr(9) || get_parent_org(p_rec.person_id) || chr(9) ||
                         get_grade(p_rec.person_id));
          END IF;
        END IF;
      EXCEPTION
        WHEN can_not_fnd_create_user THEN
          UPDATE xx_sshr_allocate_resp ir
          SET ir.error_during_resp_allocation = v_sqlerrm
          WHERE sshr_allocate_resp_id = p_rec.sshr_allocate_resp_id;
        WHEN error_in_fnd_user_pkg THEN
          UPDATE xx_sshr_allocate_resp ir
          SET ir.error_during_resp_allocation = 'Error while creating FND User. Please see
Concurrent Log file for details'
          WHERE sshr_allocate_resp_id = p_rec.sshr_allocate_resp_id;
        WHEN OTHERS THEN
          v_sqlerrm := SQLERRM;
          UPDATE xx_sshr_allocate_resp
          SET error_during_resp_allocation = substr(v_sqlerrm
                                                    ,1
                                                    ,2000)
          WHERE sshr_allocate_resp_id = p_rec.sshr_allocate_resp_id;

      END;
    END LOOP;

    FOR p_recx IN (SELECT *
                   FROM xx_sshr_allocate_resp isar
                   WHERE isar.fnd_request_id = fnd_global.conc_request_id
                   AND error_during_resp_allocation IS NOT NULL)
    LOOP
```

```
    excel_output('Error' || chr(9) || get_user_name(p_recx.fnd_user_id) ||
                 chr(9) || get_emp_no(p_recx.person_id) || chr(9) ||
                 get_person_name(p_recx.person_id) || chr(9) ||
                 to_char(p_recx.responsibility_alloc_date
                       ,'DD-MON-YYYY') || chr(9) ||
                 p_recx.error_during_resp_allocation || chr(9) ||
                 get_cost_centre_group(p_recx.person_id) || chr(9) ||
                 get_parent_org(p_recx.person_id) || chr(9) ||
                 get_grade(p_recx.person_id));
   END LOOP;
   fnd_profile.put(NAME => 'SIGNON_PASSWORD_HARD_TO_GUESS'
                 ,val  => v_hard_password);
   debug_end_procedure;
  EXCEPTION
   WHEN OTHERS THEN
    fnd_profile.put(NAME => 'SIGNON_PASSWORD_HARD_TO_GUESS'
                  ,val  => v_hard_password);
    RAISE;
  END run;

END xx_sshr_allocate_resp_pkg;
```

Posted 29th August 2012 by Krishnareddy

1 View comments

29th August 2012

AP JOINS IN R12

AP JOINS IN R12

```
 CE_BANK_ACCOUNTS CBA,
   CE_BANK_ACCT_USES_ALL CBAU,
   CE_BANK_BRANCHES_V CBB,
   AP_SYSTEM_PARAMETERS_ALL ASPA,
   ce_payment_documents PD,
   AP_LOOKUP_CODES ALC1,
   iby_payment_methods_vl iby1,
   iby_payment_profiles iby2,
   fnd_lookups iby3,
   fnd_lookups iby5,
   AP_LOOKUP_CODES ALC3,
   FND_DOCUMENT_SEQUENCES FDS,
   FND_DOC_SEQUENCE_CATEGORIES FDSC,
   FND_TERRITORIES_VL FT,
   GL_DAILY_CONVERSION_TYPES GDCT,
   AP_SUPPLIERS PV,
   AP_SUPPLIER_SITES_ALL PVS,
```

```
  CE_STATEMENT_RECONCILS_ALL CSR,
  CE_STATEMENT_HEADERS CSH,
  CE_STATEMENT_LINES CSL,
  AP_CHECKS AC,
  GL_DAILY_CONVERSION_TYPES GDCT1,
  HZ_PARTIES HZP,
  HZ_PARTY_SITES HPS,
  HZ_LOCATIONS HZL,
  /* Bug 8345877 */
  AP_SUPPLIERS PV1,
  AP_SUPPLIER_SITES_ALL PVS1,
  HZ_PARTY_SITES HPS1,
  HZ_LOCATIONS HZL1,
  /* Bug 8345877 */
  HZ_PARTIES HZP1
  /*Bug 8579660*/
WHERE AC.CE_BANK_ACCT_USE_ID       = CBAU.bank_acct_use_id(+)
AND CBAU.bank_account_id           = CBA.bank_account_id(+)
AND CBAU.ORG_ID                    = ASPA.ORG_ID(+)
AND AC.MATURITY_EXCHANGE_RATE_TYPE = GDCT1.CONVERSION_TYPE(+)
AND CBB.BRANCH_PARTY_ID(+)         = CBA.BANK_BRANCH_ID
AND AC.PAYMENT_DOCUMENT_ID         = PD.payment_document_id(+)
AND ALC1.LOOKUP_TYPE               = 'PAYMENT TYPE'
AND ALC1.LOOKUP_CODE               = AC.PAYMENT_TYPE_FLAG
AND IBY1.PAYMENT_METHOD_CODE (+)   = AC.PAYMENT_METHOD_CODE
AND iby2.payment_profile_id (+)    = AC.payment_profile_id
AND ALC3.LOOKUP_TYPE (+)           = 'CHECK STATE'
AND ALC3.LOOKUP_CODE (+)           = AC.STATUS_LOOKUP_CODE
AND AC.BANK_CHARGE_BEARER          = IBY3.LOOKUP_CODE(+)
AND IBY3.LOOKUP_TYPE (+)           = 'IBY_BANK_CHARGE_BEARER'
AND AC.SETTLEMENT_PRIORITY         = IBY5.LOOKUP_CODE (+)
AND IBY5.LOOKUP_TYPE (+)           = 'IBY_SETTLEMENT_PRIORITY'
AND AC.DOC_SEQUENCE_ID             = FDS.DOC_SEQUENCE_ID (+)
AND FDSC.APPLICATION_ID(+)         = 200
AND AC.DOC_CATEGORY_CODE           = FDSC.CODE (+)
AND AC.COUNTRY                     = FT.TERRITORY_CODE (+)
AND AC.EXCHANGE_RATE_TYPE          = GDCT.CONVERSION_TYPE (+)
AND AC.VENDOR_ID                   = PV.VENDOR_ID (+)
AND AC.PARTY_ID                    = HZP.PARTY_ID (+)
AND AC.VENDOR_SITE_ID              = PVS.VENDOR_SITE_ID (+)
AND AC.PARTY_SITE_ID               = HPS.PARTY_SITE_ID (+)
AND HPS.LOCATION_ID                = HZL.LOCATION_ID (+)
AND CSR.REFERENCE_TYPE (+)         = 'PAYMENT'
AND CSR.REFERENCE_ID (+)           = AC.CHECK_ID
AND CSR.CURRENT_RECORD_FLAG (+)    = 'Y'
AND CSR.STATEMENT_LINE_ID          = CSL.STATEMENT_LINE_ID (+)
AND CSL.STATEMENT_HEADER_ID        = CSH.STATEMENT_HEADER_ID (+)
AND CSR.STATUS_FLAG (+)            = 'M'
  /* Bug 8345877 */
AND AC.REMIT_TO_SUPPLIER_ID      = PV1.VENDOR_ID (+)
AND AC.REMIT_TO_SUPPLIER_SITE_ID = PVS1.VENDOR_SITE_ID (+)
```

```
AND PVS1.PARTY_SITE_ID          = HPS1.PARTY_SITE_ID (+)
AND HPS1.LOCATION_ID            = HZL1.LOCATION_ID (+)
  /* Bug 8345877 */
AND PV1.party_id = HZP1.PARTY_ID (+)
```

Posted 29th August 2012 by Krishnareddy

1 View comments

29th August 2012 Alerts

Alerts

[http://www.blogger.com/blogger.g?blogID=955765254185387334]
Introduction:
Oracle Alerts is something that can be used to Notify/Alert to one or multiple persons about an activity or change that occurs in the system. The alerts can also be used to call a procedure, run some sql script etc.
There are 2 types of alert
1) Periodic Alert
2) Event Alert
Periodic Alerts:
These alerts are trigger periodically, hourly, daily, weekly, monthly etc based upon how it is setup to be triggered. When alert runs and the condition(SQL Query etc.) in the alerts fetches record, then the events specified in the alert are triggered.
Ex. 1) Daily alert to send notification on the sales order on which credit check hold is applied for a day
2) Hourly alert to send notification on all the concurrent request that completed with error
3/If you want to know list of items created on that day at the end of day you can use periodic alerts repeating periodically by single day.This alert is not based on any chages to database.this alert will notify you everyday regardless of data exists or not that means even if no items are created you wil get a blank notification.
Event Alerts:
These Alerts are fired/triggered based on some change in data in the database. This is very similar to the triggers written on the table. Unlikely, event alerts can only fire on After Insert or After Update.
Ex. 1) An alert that sends notification when new item is created.
Ex: If u want to notify your manager when you create an item in the inventory you can use event based alerts. When you create an item in the inventory it will cretae a new record in mtl_system_items_b, here inserting a record in the table is an event so when ever a new record is inserted it will send the alert.In same alert you can also send the information related to that particular item.
What can be done with Alerts :
1.You can send notifications

2.You can send log files as attachments to notifications
3.You can call PL/SQL stores procedures
4.You can send approval emails and get the results
5.Print some content dynamically
How to create an Alert?
1.Study your Business requirement and decide what type of alert you need either periodic alert or event based alert.

2. If you are going for periodic alert decide the frequency.
3. If you have chosen event based alert then find out on whst event(insert,update,delete) you want to fire the alert.
4. Decide what data need to be included in the alert.
5. Based on the data you want in the alert write a SELECT SQL statement to pull the data.
6. Create a distribution list grouping all the people to whom you want to send the alert.
Navigation :
1. Go to "Alert Manager" Responsibility.
2. Alert >> Define
Business Requirement

Notify when sales order is booked or new line is entered on booked order
We can do this through triggers. Alternative way is Alerts
Query

```
SELECT ooh.order_number
, ool.line_number||'.'||ool.shipment_number line_number
, ordered_item, ordered_quantity, ool.flow_Status_code
INTO &order_num, &line_num,&Item_num, &Quantity, &line_Status
FROM oe_order_headers_all ooh, oe_order_lines_all ool
WHERE ooh.header_id = ool.header_id
AND
( ooh.booked_date >= to_date(Sysdate,'DD-MON-RRRR HH24:MI:SS')
OR (ool.creation_Date >= to_date(Sysdate,'DD-MON-RRRR HH24:MI:SS')
AND ool.creation_date > ooh.booked_date)
)
```

2/Define Actions
Click on the actions button and then actions Detail button and define message as shown in screenshot. Note that the message type is summary.

3) Define Action Sets
Click on action sets and then action set details and in the members tab enter the action

4) Schedule the Request
Navigate to Request --> Check and submit the alert. Based on the definition of alert it will be scheduled to run.
Second Example

We have to design periodic alert “Items with zero weight” for our Client.Basic Business need of this alerts is for Email to MIS User for List of item having Zero Weight saved in Oracle Inventory.

Alert Manager=>Alerts=>Define

Application Field: Name of Application e.g Oracle Payable
Name : User defined Name of Alerts.
Choose Period tab
Frequency :Choose Frequency Accordingly.
Days: Choose days according to Frequency. Suppose you Chosse Frequency “Every N Business Days” Then Enter Value “1” in Day Field.
Start Time: Time You want to Fire periodic alerts suppose you want to fire at 6:30 A.M Write “06:30:00” in Start Time Field.
Keep: How many days You Want to Mainitain History.
Select Statement: Write Query in select Statement . Here is an Exapmle of Select Statement for Test Periodic Alert “Items with zero weight”. Select statement must include an INTO clause that contains one output for each column selected by your Select statement.In Example all input Column like Orgaization_code,tem_number(segment1),Description,Creation_date have Output Variable ORG,ITEM,DESCR,Create_date preceded by Amperstand(&).
Query is: Test Query for “Items with zero weight” Alert is

```
SELECT
distinct p.organization_code ,
substr(i.segment1,1,20) ,
substr(i.description,1,50),
i.creation_date,
INTO
&org
, &item
, &descr
, &create_date
FROM
mtl_system_items i,
mtl_parameters p
where i.organization_id = p.organization_id
and p.organization_code in ('INF','ENF')
and i.INVENTORY_ITEM_STATUS_CODE||'' = 'Active'
and i.unit_weight is null
and I.ITEM_TYPE = 'P'
order by 1,2
```

Verify: Click on Verify Button to Verify your Query. This message will populate if Query is Right.

Run: To Check Record Count of Your Query Click on Run Button. Suppose Your Query Written

Zero Rows This Message will populate.
STEP2 : Define Action:
Click on Action Button(Marked With Circle). This Form (Action Window 1.0) Will Poulate..
Explaination of Field on this Form :
Action Name:Specify Name of your Action. For Example Our Alert ""Items with zero weight". is for
Email.we specify name "EMAIL" in Action Name.
Action Level: Action level should be Summary.
Three are three Action Level
Detail - action is performed for each exception returned.
Summary - action performed once for each exception.
No Exception - action performed when no exceptions returned.
Click on Action Details

Action Type: Four Action type.
Message - send message
Concurrent Program Request - submit concurrent program
SQL script - execute a SQL script
OS script - execute an OS script
We Choose Action Type "Message" because "Items with zero weight" Alert is for Email Purpose.
Enter Email Addrees in To Field.
Text : In Text field design Layout of your Periodic Alerts. "Items with zero weight" Alert Layout is this:
Important thing is that Output Variable &org,&item etc should be placed with in template like this...

=**= Enter summary template below this line =**=
**&org &item &descr &create_date
=**= Enter summary template above this line =**=
Layout Sample of Test Periodic Alert "Items with zero weight":
The following items currently have no weight maintained against them in the system.
Org Item Description Creation Date
=== ==========
==========
=**= Enter summary template below this line =**=
**&org &item &descr &create_date
=**= Enter summary template above this line =**=
Column OverFlow: 'Wrap'
Max Width: Choose According to Requirments.
80 Characters
132 Characters
180 Characters
STEP3 : Define Action Sets:

Enter the groups of action to be run. Click on Action Sets Button.
Action Set Window (Shown in Pictiure “Action set Window 1.2”) will populate.
Action set Name: Enter Action Set Name. In our Test Alert we enter Action set Name “EMAIL”.
Note : Action set Name should be same as in Action Name. Otherwise Periodic Alert will not Fire.

Action set Name: Enter Action Set Name. In our Test Alert we enter Action set Name “EMAIl”.
Output Tab: Output tab Contain List of Output Variable defoen in select Statement as shown in Window 1.4.Note that if you Output tab is Blank Requery the Alert then it will show output Variable.
This problem we generally faced when designing periodic Alerts.
Member Tab: Enter Name of Action .

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

HZ_CUST_ACCOUNTS and HZ_PARTIES

HZ_CUST_ACCOUNTS and HZ_PARTIES

In R12 ra_customers is absoleted So we can use AR_CUSTOMERS it is view base on HZ_CUST_ACCOUNTS and HZ_PARTIES

CREATE OR REPLACE FORCE VIEW "APPS"."AR_CUSTOMERS" ("CUSTOMER_ID", "CUSTOMER_NAME", "CUSTOMER_NUMBER", "CUSTOMER_KEY", "STATUS", "ORIG_SYSTEM_REFERENCE", "CUSTOMER_PROSPECT_CODE", "CUSTOMER_CATEGORY_CODE", "CUSTOMER_CLASS_CODE", "CUSTOMER_TYPE", "PRIMARY_SALESREP_ID", "SIC_CODE", "TAX_REFERENCE", "TAX_CODE", "FOB_POINT", "SHIP_VIA", "GSA_INDICATOR", "SHIP_PARTIAL", "TAXPAYER_ID", "PRICE_LIST_ID", "FREIGHT_TERM", "ORDER_TYPE_ID", "SALES_CHANNEL_CODE", "WAREHOUSE_ID", "MISSION_STATEMENT", "NUM_OF_EMPLOYEES", "POTENTIAL_REVENUE_CURR_FY", "POTENTIAL_REVENUE_NEXT_FY", "FISCAL_YEAREND_MONTH", "YEAR_ESTABLISHED", "ANALYSIS_FY", "COMPETITOR_FLAG", "REFERENCE_USE_FLAG", "THIRD_PARTY_FLAG", "ATTRIBUTE_CATEGORY", "ATTRIBUTE1", "ATTRIBUTE2", "ATTRIBUTE3", "ATTRIBUTE4", "ATTRIBUTE5", "ATTRIBUTE6", "ATTRIBUTE7", "ATTRIBUTE8", "ATTRIBUTE9", "ATTRIBUTE10", "ATTRIBUTE11", "ATTRIBUTE12", "ATTRIBUTE13", "ATTRIBUTE14", "ATTRIBUTE15", "LAST_UPDATED_BY", "LAST_UPDATE_DATE",

```
"LAST_UPDATE_LOGIN",
  "CREATED_BY", "CREATION_DATE", "CUSTOMER_NAME_PHONETIC",
"TAX_HEADER_LEVEL_FLAG", "TAX_ROUNDING_RULE",
"GLOBAL_ATTRIBUTE_CATEGORY", "GLOBAL_ATTRIBUTE1",
"GLOBAL_ATTRIBUTE2", "GLOBAL_ATTRIBUTE3", "GLOBAL_ATTRIBUTE4",
"GLOBAL_ATTRIBUTE5", "GLOBAL_ATTRIBUTE6", "GLOBAL_ATTRIBUTE7",
"GLOBAL_ATTRIBUTE8", "GLOBAL_ATTRIBUTE9", "GLOBAL_ATTRIBUTE10",
"GLOBAL_ATTRIBUTE11", "GLOBAL_ATTRIBUTE12", "GLOBAL_ATTRIBUTE13",
"GLOBAL_ATTRIBUTE14", "GLOBAL_ATTRIBUTE15", "GLOBAL_ATTRIBUTE16",
"GLOBAL_ATTRIBUTE17", "GLOBAL_ATTRIBUTE18", "GLOBAL_ATTRIBUTE19",
"GLOBAL_ATTRIBUTE20")
AS
  SELECT CUST.CUST_ACCOUNT_ID CUSTOMER_ID ,
    substrb(PARTY.PARTY_NAME,1,50) CUSTOMER_NAME ,
    CUST.ACCOUNT_NUMBER CUSTOMER_NUMBER ,
    PARTY.CUSTOMER_KEY CUSTOMER_KEY ,
    CUST.STATUS STATUS ,
    CUST.ORIG_SYSTEM_REFERENCE ORIG_SYSTEM_REFERENCE ,
    'CUSTOMER' CUSTOMER_PROSPECT_CODE ,
    PARTY.CATEGORY_CODE CUSTOMER_CATEGORY_CODE ,
    CUST.CUSTOMER_CLASS_CODE CUSTOMER_CLASS_CODE ,
    CUST.CUSTOMER_TYPE CUSTOMER_TYPE ,
    CUST.PRIMARY_SALESREP_ID PRIMARY_SALESREP_ID ,
    DECODE(PARTY.PARTY_TYPE, 'ORGANIZATION', PARTY.SIC_CODE, NULL)
SIC_CODE ,
    PARTY.TAX_REFERENCE TAX_REFERENCE ,
    CUST.TAX_CODE TAX_CODE ,
    CUST.FOB_POINT FOB_POINT ,
    CUST.SHIP_VIA SHIP_VIA ,
    DECODE(PARTY.PARTY_TYPE, 'ORGANIZATION', PARTY.GSA_INDICATOR_FLAG,
'N') GSA_INDICATOR ,
    CUST.SHIP_PARTIAL SHIP_PARTIAL ,
    PARTY.JGZZ_FISCAL_CODE TAXPAYER_ID ,
    CUST.PRICE_LIST_ID PRICE_LIST_ID ,
    CUST.FREIGHT_TERM FREIGHT_TERM ,
    CUST.ORDER_TYPE_ID ORDER_TYPE_ID ,
    CUST.SALES_CHANNEL_CODE SALES_CHANNEL_CODE ,
    CUST.WAREHOUSE_ID WAREHOUSE_ID ,
    DECODE(PARTY.PARTY_TYPE, 'ORGANIZATION', PARTY.MISSION_STATEMENT,
NULL) MISSION_STATEMENT ,
    DECODE(PARTY.PARTY_TYPE, 'ORGANIZATION', PARTY.EMPLOYEES_TOTAL,
TO_NUMBER(NULL)) NUM_OF_EMPLOYEES ,
    DECODE(PARTY.PARTY_TYPE, 'ORGANIZATION',
PARTY.CURR_FY_POTENTIAL_REVENUE, TO_NUMBER(NULL))
POTENTIAL_REVENUE_CURR_FY ,
    DECODE(PARTY.PARTY_TYPE, 'ORGANIZATION',
PARTY.NEXT_FY_POTENTIAL_REVENUE, TO_NUMBER(NULL))
POTENTIAL_REVENUE_NEXT_FY ,
    DECODE(PARTY.PARTY_TYPE, 'ORGANIZATION',
PARTY.FISCAL_YEAREND_MONTH, NULL) FISCAL_YEAREND_MONTH ,
    DECODE(PARTY.PARTY_TYPE, 'ORGANIZATION', PARTY.YEAR_ESTABLISHED,
```

```
TO_NUMBER(NULL)) YEAR_ESTABLISHED ,
  DECODE(PARTY.PARTY_TYPE, 'ORGANIZATION', PARTY.ANALYSIS_FY, NULL)
ANALYSIS_FY ,
  PARTY.COMPETITOR_FLAG COMPETITOR_FLAG ,
  PARTY.REFERENCE_USE_FLAG REFERENCE_USE_FLAG ,
  PARTY.THIRD_PARTY_FLAG THIRD_PARTY_FLAG ,
  CUST.ATTRIBUTE_CATEGORY ATTRIBUTE_CATEGORY ,
  CUST.ATTRIBUTE1 ATTRIBUTE1 ,
  CUST.ATTRIBUTE2 ATTRIBUTE2 ,
  CUST.ATTRIBUTE3 ATTRIBUTE3 ,
  CUST.ATTRIBUTE4 ATTRIBUTE4 ,
  CUST.ATTRIBUTE5 ATTRIBUTE5 ,
  CUST.ATTRIBUTE6 ATTRIBUTE6 ,
  CUST.ATTRIBUTE7 ATTRIBUTE7 ,
  CUST.ATTRIBUTE8 ATTRIBUTE8 ,
  CUST.ATTRIBUTE9 ATTRIBUTE9 ,
  CUST.ATTRIBUTE10 ATTRIBUTE10 ,
  CUST.ATTRIBUTE11 ATTRIBUTE11 ,
  CUST.ATTRIBUTE12 ATTRIBUTE12 ,
  CUST.ATTRIBUTE13 ATTRIBUTE13 ,
  CUST.ATTRIBUTE14 ATTRIBUTE14 ,
  CUST.ATTRIBUTE15 ATTRIBUTE15 ,
  CUST.LAST_UPDATED_BY LAST_UPDATED_BY ,
  CUST.LAST_UPDATE_DATE LAST_UPDATE_DATE ,
  CUST.LAST_UPDATE_LOGIN LAST_UPDATE_LOGIN ,
  CUST.CREATED_BY CREATED_BY ,
  CUST.CREATION_DATE CREATION_DATE ,
  DECODE(PARTY.PARTY_TYPE, 'ORGANIZATION',
PARTY.ORGANIZATION_NAME_PHONETIC,NULL) CUSTOMER_NAME_PHONETIC ,
  CUST.TAX_HEADER_LEVEL_FLAG TAX_HEADER_LEVEL_FLAG ,
  CUST.TAX_ROUNDING_RULE TAX_ROUNDING_RULE ,
  CUST.GLOBAL_ATTRIBUTE_CATEGORY GLOBAL_ATTRIBUTE_CATEGORY ,
  CUST.GLOBAL_ATTRIBUTE1 GLOBAL_ATTRIBUTE1 ,
  CUST.GLOBAL_ATTRIBUTE2 GLOBAL_ATTRIBUTE2 ,
  CUST.GLOBAL_ATTRIBUTE3 GLOBAL_ATTRIBUTE3 ,
  CUST.GLOBAL_ATTRIBUTE4 GLOBAL_ATTRIBUTE4 ,
  CUST.GLOBAL_ATTRIBUTE5 GLOBAL_ATTRIBUTE5 ,
  CUST.GLOBAL_ATTRIBUTE6 GLOBAL_ATTRIBUTE6 ,
  CUST.GLOBAL_ATTRIBUTE7 GLOBAL_ATTRIBUTE7 ,
  CUST.GLOBAL_ATTRIBUTE8 GLOBAL_ATTRIBUTE8 ,
  CUST.GLOBAL_ATTRIBUTE9 GLOBAL_ATTRIBUTE9 ,
  CUST.GLOBAL_ATTRIBUTE10 GLOBAL_ATTRIBUTE10 ,
  CUST.GLOBAL_ATTRIBUTE11 GLOBAL_ATTRIBUTE11 ,
  CUST.GLOBAL_ATTRIBUTE12 GLOBAL_ATTRIBUTE12 ,
  CUST.GLOBAL_ATTRIBUTE13 GLOBAL_ATTRIBUTE13 ,
  CUST.GLOBAL_ATTRIBUTE14 GLOBAL_ATTRIBUTE14 ,
  CUST.GLOBAL_ATTRIBUTE15 GLOBAL_ATTRIBUTE15 ,
  CUST.GLOBAL_ATTRIBUTE16 GLOBAL_ATTRIBUTE16 ,
  CUST.GLOBAL_ATTRIBUTE17 GLOBAL_ATTRIBUTE17 ,
  CUST.GLOBAL_ATTRIBUTE18 GLOBAL_ATTRIBUTE18 ,
  CUST.GLOBAL_ATTRIBUTE19 GLOBAL_ATTRIBUTE19 ,
```

```
  CUST.GLOBAL_ATTRIBUTE20 GLOBAL_ATTRIBUTE20
FROM HZ_CUST_ACCOUNTS CUST,
  HZ_PARTIES PARTY
WHERE cust.party_id = party.party_id;
```

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

Purchase Order Types

Purchase Order Types

Oracle Purchasing provides the following purchase order types: Standard Purchase Order, Planned Purchase Order, Blanket Purchase Agreement and Contract Purchase Agreement. You can use the Document Name field in the Document Types window to change the names of these documents. For example, if you enter Regular Purchase Order in the Document Name field for the Standard Purchase Order type, your choices in the Type field in the Purchase Orders window will be Regular Purchase Order, Planned Purchase Order, Blanket Purchase Agreement and Contract Purchase Agreement.

Standard Purchase Orders:-

You generally create standard purchase orders for one–time purchase of various items. You create standard purchase orders when you know the details of the goods or services you require, estimated costs, quantities, delivery schedules, and accounting distributions. If you use encumbrance accounting, the purchase order may be encumbered since the required information is known.

Blanket Purchase Agreements (BPA):-

You create blanket purchase agreements when you know the detail of the goods or services you plan to buy from a specific supplier in a period, but you do not yet know the detail of your delivery schedules. You can use blanket purchase agreements to specify negotiated prices for your items before actually purchasing them. BPA are widely used in product manufacturing companies.

You can issue a **blanket release** against a blanket purchase agreement to place the actual order (as long as the release is within the blanket agreement effectivety dates). If you use encumbrance accounting, you can encumber each release.

Contract Purchase Agreements:-

You create contract purchase agreements with your suppliers to agree on specific terms and conditions without indicating the goods and services that you will be purchasing. You can later issue standard purchase orders referencing your contracts, and you can encumber these purchase orders if you use encumbrance accounting.

Planned Purchase Order:-

You create a planned purchase order when you want to establish a long term agreement with a single source supplier with a commitment to buy goods or services. Planned purchase orders include tentative delivery schedules and accounting distributions. You then create scheduled releases against the planned purchase order to actually order the goods or services.

A planned purchase order is a type of purchase order you issue before you order actual delivery of goods and services for specific dates and locations. A **scheduled release** is issued against a planned purchase order to place the actual order. You can also change the accounting distributions on each release and the system will reverse the encumbrance for the planned purchase order and create a new encumbrance for the release.

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

FNDLOAD

The Generic Loader (FNDLOAD) is a concurrent program that can transfer Oracle Application entity data between database and text file. The loader reads a configuration file to determine which entity to access. In simple words FNDLOAD is used to transfer entity data from one instance/database to other. For example if you want to move a concurrent program/menu/value sets developed in DEVELOPMENT instance to PRODUCTION instance you can use this command.

Steps to Move a Concurrent program from one instance(Database) to other

- Define your concurrent program and save it in first instance(for how to register a concurrent program click here)
- Connect to your UNIX box on first instance and run the following command to download the .ldt file
  FNDLOAD apps/apps O Y DOWNLOAD $FND_TOP/patch/115/import/afcpprog.lct file_name.ldt PROGRAM APPLICATION_SHORT_NAME="Concurrent program application short name" CONCURRENT_PROGRAM_NAME="concurrent program short name"
- Move the downloaded .ldf file to new instance(Use FTP)
- Connect to your UNIX box on second instance and run the following command to upload the .ldt file
  FNDLOAD apps/apps 0 Y UPLOAD $FND_TOP/patch/115/import/afcpprog.lct file_name.ldt

Note: Make sure you are giving proper .lct file in the commands and don't confuse with .lct and .ldt files

These following are the other entity data types that we can move with FNDLOAD

1 – Printer Styles
FNDLOAD apps/apps O Y DOWNLOAD $FND_TOP/patch/115/import/afcppstl.lct file_name.ldt
STYLE PRINTER_STYLE_NAME="printer style name"

2 – Lookups
FNDLOAD apps/apps O Y DOWNLOAD $FND_TOP/patch/115/import/aflvmlu.lct file_name.ldt
FND_LOOKUP_TYPE APPLICATION_SHORT_NAME="FND"
LOOKUP_TYPE="lookup name"

3 – Descriptive Flexfield with all of specific Contexts
FNDLOAD apps/apps O Y DOWNLOAD $FND_TOP/patch/115/import/afffload.lct file_name.ldt
DESC_FLEX P_LEVEL=COL_ALL:REF_ALL:CTX_ONE:SEG_ALL
APPLICATION_SHORT_NAME="FND" DESCRIPTIVE_FLEXFIELD_NAME="desc flex name"
P_CONTEXT_CODE="context name"

4 – Key Flexfield Structures
FNDLOAD apps/apps O Y DOWNLOAD $FND_TOP/patch/115/import/afffload.lct file_name.ldt
KEY_FLEX
P_LEVEL=COL_ALL:FQL_ALL:SQL_ALL:STR_ONE:WFP_ALL:SHA_ALL:CVR_ALL:SEG_ALL
APPLICATION_SHORT_NAME="FND" ID_FLEX_CODE="key flex code"
P_STRUCTURE_CODE="structure name"

5 – Concurrent Programs
FNDLOAD apps/apps O Y DOWNLOAD $FND_TOP/patch/115/import/afcpprog.lct file_name.ldt
PROGRAM APPLICATION_SHORT_NAME="FND"
CONCURRENT_PROGRAM_NAME="concurrent name"

6 – Value Sets
FNDLOAD apps/apps O Y DOWNLOAD $FND_TOP/patch/115/import/afffload.lct file_name.ldt
VALUE_SET_VALUE FLEX_VALUE_SET_NAME="value set name"

7 – Value Sets with values
FNDLOAD apps/apps O Y DOWNLOAD $FND_TOP/patch/115/import/afffload.lct file_name.ldt
VALUE_SET FLEX_VALUE_SET_NAME="value set name"

8 – Profile Options
FNDLOAD apps/apps O Y DOWNLOAD $FND_TOP/patch/115/import/afscprof.lct file_name.ldt
PROFILE PROFILE_NAME="profile option" APPLICATION_SHORT_NAME="FND"

9 – Request Groups
FNDLOAD apps/apps O Y DOWNLOAD $FND_TOP/patch/115/import/afcpreqg.lct file_name.ldt
REQUEST_GROUP REQUEST_GROUP_NAME="request group"
APPLICATION_SHORT_NAME="FND"

10 – Request Sets
FNDLOAD apps/apps O Y DOWNLOAD $FND_TOP/patch/115/import/afcprset.lct file_name.ldt
REQ_SET
APPLICATION_SHORT_NAME="FND" REQUEST_SET_NAME="request set"

11 – Responsibilities

FNDLOAD apps/apps O Y DOWNLOAD $FND_TOP/patch/115/import/afscursp.lct file_name.ldt FND_RESPONSIBILITY RESP_KEY="responsibility"

12 – Menus
FNDLOAD apps/apps O Y DOWNLOAD $FND_TOP/patch/115/import/afsload.lct file_name.ldt MENU MENU_NAME="menu_name"

13 – Forms Personalization
FNDLOAD apps/apps 0 Y DOWNLOAD $FND_TOP/patch/115/import/affrmcus.lct file_name.ldt FND_FORM_CUSTOM_RULES function_name=FUNCTION_NAME

Note: UPLOAD command is same for all except replacing the .lct and passing any extra parameters if you want to pass

FNDLOAD apps/apps 0 Y UPLOAD $FND_TOP/patch/115/import/corresponding.lct upload_file.ldt

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

Trace file

The main use of enabling trace for a concurrent program comes during performance tuning.
By examining a trace file, we come to know which query/queries is/are taking the longest time to execute, there by letting us to concentrate on tuning them in order to improve the overall performance of the program.
The following is an illustration of how to Enable and View a trace file for a Concurrent Program.

- Navigation: Application Developer–>Concurrent–>Program
  - Check the Enable Trace Check box. After that go to that particular Responsibility and run the Concurrent Program.
  - Check that the Concurrent Program has been completed successfully.
  - The trace file by default is post fixed with oracle Process_id which helps us to identify which trace file belongs to which concurrent

request. The below SQL Query returns the process_id of the concurrent request:

Select oracle_process_id from fnd_concurrent_requests where request_id='2768335′
(This query displays Process Id)

- The path to the trace file can be found by using the below query:

SELECT * FROM V$PARAMETER WHERE NAME='user_dump_dest'
(This Query displays the path of trace file)

- The Trace File generated will not be in the readable format. We have to use TKPROF utility to convert the file into a readable format.

- Run the below tkprof command at the command prompt.

TKPROF < Trace File_Name.trc> <Output_File_Name.out> SORT=fchela

A readable file will be generated from the original trace file which can be further analyzed to improve the performance. This file has the information about the parsing, execution and fetch times of various queries used in the program.

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012 Shell scripting

Objectives:

- Steps to Register Shell Script as a concurrent program
- Sample Shell Script to copy the file from source to destination
- Basic Shell Script Commands

Steps to Register Shell Script as a concurrent program

step 1:

=======
Place the <name>.prog script under the bin directory for your
applications top directory.

For example, call the script ERPS_DEMO.prog and place it under
$CUSTOM_TOP/bin

step 2:
=======
Make a symbolic link from your script to $FND_TOP/bin/fndcpesr
For example, if the script is called ERPS_DEMO.prog use this:

ln -s $FND_TOP/bin/fndcpesr ERPS_DEMO

This link should be named the same as your script without the
.prog extension.

Put the link for your script in the same directory where the
script is located.

step 3:
=======
Register the concurrent program, using an execution method of
'Host'. Use the name of your script without the .prog extension
as the name of the executable.

For the example above:
Use ERPS_DEMO

step 4:
=======
Your script will be passed at least 4 parameters, from $1 to $4.
$1 = orauser/pwd
$2 = userid(apps)
$3 = username,
$4 = request_id

Any other parameters you define will be passed in as $5 and higher.
Make sure your script returns an exit status also.

Sample Shell Script to copy the file from source to destination

#Note: If you see # in front of any line it means that it's a comment line not the actual code
#** **********
Created By : Prudhvi A
Creation Date : 19-FEB-2008
Script Name : ERPSCHOOLS.prog
Description : This Script accepts three parameters
1)Data File Name 2)Source Directory Path 3)Target Directory Path
Then copy the file from source location to target location.

```
# If copy fails send the error status/message to concurrent program so that user can see status.
#
#
# ========
# History
# ========
# Version 1 Prudhvi A 19-FEB-2008 Created for erpschools.com users
#
#** *********************************************************************
#Parameters from 1 to 4 i.e $1 $2 $3 $4 are standard parameters
# $1 : username/password of the database
# $2 : userid
# $3 : USERNAME
# $4 : Concurrent Request ID
DataFileName=$5
SourceDirectory=$6
TargetDirectory=$7
echo "———————————————————————————————"
echo "Parameters received from concurrent program .."
echo " Time : "`date`
echo "———————————————————————————————"
echo "Arguments : "
echo " Data File Name : "${DataFileName}
echo " SourceDirectory : "${SourceDirectory}
echo " TargetDirectory : "${TargetDirectory}
echo "———————————————————————————————"
echo " Copying the file from source directory to target directory…"
cp ${SourceDirectory}/${DataFileName} ${TargetDirectory}
if [ $? -ne 0 ]
# the $? will contain the result of previously executed statement.
#It will be 0 if success and 1 if fail in many cases
# -ne represents not "equal to"
then
echo "Entered Exception"
exit 1
# exit 1 represents concurrent program status. 1 for error, 2 for warning 0 for success
else
echo "File Successfully copied from source to destination"
exit 0
fi
echo "*********************************************************************"
```

Basic Shell Script Commands

```
# Create Directory
mkdir <dirname>
# Remove Directory
rmdir <dirname>
#remove folder with files
rm -r -f <dirname>
# Change Directory
```

cd **<newpath>**
Create new file
vi <newfile.ext>
#insert data into file
vi <openfilename.ext>
esc i <make changes>
#Save file
esc :wq enter
exit without saving changes
esc :q! enter
open existing file
vi <existingfilename.ext>
#remove file
rm <filename.ext>
copy file with same name
cp <sourcedir>/<sourcefilename.ext> <destinationdir>
copy file with new name
cp <sourcedir>/<sourcefilename.ext> <destinationdir>/<newfilename.ext>
Move file with same name
mv <sourcedir>/<sourcefilename.ext> <destinationdir>
move file with data appended to filename in the front
mv <sourcedir>/<sourcefilename.ext>
<destinationdir>/`date+%H%M%d%m%y`<filename.ext>
#print line
echo "your text here to print"
#print date
echo `date`

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

Return Material Authorization (RMA) in Order Management

The following topics will be discussed in this article:

1. Overview of RMA
2. Creating a New Return (RMA with Credit)
3. Creating a Return using the Copy Function (RMA with Credit)
4. Creating a New Return (RMA no Credit)
5. Creating a New Return (RMA Credit Only)
6. RMA Receipt
7. Viewing the status of RMA
8. Close RMA

Pre-requisites:

1. Return Order Categories and Transaction Types have been defined.

2. Items exist in the Item Master with the attribute *Returnable* enabled.
3. Those items must exist on a Price List.
4. Customers exist in the Customer Master.
5. Return reasons have been defined.
6. Discounts have been defined.
7. Salespersons have been defined.

1. Overview

Order Management allows the customers to return the goods to you. OM also enables you to authorize the return of your sales orders as well as sales made by other dealers or suppliers, as long as the items are part of your item master and price list. You can authorize returns for replacement that returns with or without credit.

RMA can be created in two different ways in Oracle:

- Create a **New Return** which will create the RMA from scratch
- **Copy** an original order into an order with an RMA Order Type

R2i supports four different RMA Order Types, each with a different Order Cycle:

- **RMA with Credit** is used when the customer returns the physical product and also receives credit as a result of the return.
  This type applies for:
    - Defective Product
    - Customer does not like the product
    - Product does not meet the customer's expectations

- **RMA no Credit** is used when the customer will return the product but will not be receiving a credit as a result of the return.
  These returns would be for:
    - Evaluation Orders
    - Samples
    - Other orders where the customer was not originally charged for the product.

- **RMA Credit Only** is used when the customer will receive a credit, but the physical return of the product is not required.
    - These credits are generally used by software companies when the customer destroys the CD or disk and erases the software from their machine, but no physical thing to return.

- **RMA with Credit and Approval** is used in the same manner as an RMA with Credit but this order cycle includes an approval process that requires someone to approve the RMA before it

is booked. In order for an order/return or order/return line approval workflow to work correctly the profile option **OM: Notification Approver** must have a value.

2. Creating a New Return (RMA with Credit)

Select:	**Order Management** responsibility
Navigate to:	Orders, Returns > Order Organizer or Orders, Returns > Sales Orders
Select:	**New Order** button if using the Order Organizer

Customer:	Enter Customer Name (Customer Number will default in).
Customer Number:	Alternatively, enter Customer Number and the Customer Name will default in.
Order Type:	RMA Order with Credit
Customer PO:	Enter a Customer PO number if the order type you selected requires it.
Date Ordered:	The Current Date will default in as the Order Date. You may change this date or accept the current date.
Customer Contact:	Enter a Customer contact (optional).
Order Number:	Oracle will assign the RMA number as soon as the information is saved if automatic numbering is enabled.
Price List:	Select a price list from the list of values
Ship To:	Enter the customer's ship-to address from the list of values or accept the default. (Not required for Returns)
Salesperson:	Enter the Salesperson
Status:	The initial entry status for an RMA is *Entered*. After Booking the RMA status will changed to *Booked*.
Currency:	Select the currency for the RMA from the list of values or accept the default.
Bill To:	Enter the customer's bill-to address from the list of values or accept the default.

Order Information – Others

Payment Terms:	Payment Terms are user-definable and must be setup in advance. (Setup>Orders>Payment Terms). Select from the list of values or accept the default. (Not required for Returns)
	Sales Channels are user-definable and must be setup in advance

Sales Channel:	(Order Management Quick Codes). Select a Sales Channel from the list of values or accept the default.
Warehouse:	Select a Warehouse (inventory organization) from which Returns will be received.
Shipping Method:	Not used for Returns
Line Set:	Not used for Returns
Freight Terms:	
FOB:	Defaults (In order, from Ship-To, Bill-To, Customer, Order Type, or Price List)
Shipment Priority:	Defaults from Order Type
Shipping Instruct:	Shipping instructions are printed on the pick slip. Since this is a return do not use this field.
Packing Instructions:	Packing instructions are printed on the pack slip. Since this is a return do not use this field.
Tax Handling:	Select from the following: Exempt – Indicates that this order is exempt for a normally taxable customer site and/or item. If you select exempt you must enter a reason for exemption. Require – Indicates that this order is taxable for a normally non- taxable customer and/or item. Standard – Indicates that taxation should be based on existing exemption rules. If the customer has a tax exemption defined, Order Management displays any certificate number and reason for the exemption in the corresponding fields.
Tax Exempt Number:	If you choose *Exempt* in the Tax Handling field then select an existing certificate number for the ship-to customer, or enter a new, unapproved exemption certificate number. If you chose *Standard* in the Tax Handling field, an existing exemption rule may display a certificate number in this field. Unapproved exemption certificate numbers can be approved using the Tax Exemptions window.
Exempt Reason:	If you chose *Exempt* in the Tax Handling field then select an exemption reason. If you chose *Standard* in the Tax Handling field, an existing exemption rule may display an exemption reason in this field.

	You can define tax exemption reasons using the Receivables Quickcodes window.
Payment Type:	
Amount:	Optional unless the payment type selected requires it.
Check Number:	Optional unless the payment type selected requires it.
Credit Card Type:	Optional unless the payment type selected requires it.
Credit Card Num:	Optional unless the payment type selected requires it.
Card Holder:	Optional unless the payment type selected requires it.
Card Expiration Date:	Optional unless the payment type selected requires it.
Approval Code:	Optional unless the payment type selected requires it.
Order Source:	If the RMA is copied from an existing order/return 'Copy' will appear in this field.
Order Source Rule:	If the RMA is copied from an existing order/return the original order/return number will appear in this field.

Line Items Tab – Returns

Line:	This field will automatically be populated.
Ordered Item:	Enter the Ordered Item from the list of values.
Qty:	Enter the quantity to be returned.
Return Reason:	Select a defined reason from the list of values.
Line Type:	Select a line type. A line type may default depending on the transaction type setup. Select a line type from the list of values if you wish to change the defaulted value.
Reference:	Select the appropriate Reference Type. Use the Reference Type if you want to refer to a specific Invoice, Purchase Order, or Sales Order. These references must be for transactions originally placed in Oracle. You have the option of leaving them blank, in which case the customer's credit will be placed On Account when it interfaces to Accounts Receivable. On Account credit memos may be applied to invoices at a future time.
Order:	If referencing a Sales Order then enter the Sales Order number.
Line:	If referencing a Sales Order enter the appropriate line number from the Sales Order referenced. **Note:** If creating the RMA using the copy function the information in the copied Sales Order will automatically populate in this field. If you enter *Sales Order* or *Invoice* in the Reference field, then you have the option of selecting a specific invoice in the Invoice field.

Invoice:	This would allow for a Credit Memo to be created and directly applied to this invoice. Leaving this field blank will yield an On Account credit memo in Receivables.
Invoice Line:	If referencing an Invoice, enter the appropriate line number from the Invoice referenced.
Credit Invoice:	
Item Revision:	

Line Items Tab – Main

UOM:	The UOM will default in based on the item selected to be returned.
Unit Selling Price:	The price defaults in from the invoice, purchase order, sales order, or invoice if selected in the reference field, otherwise, it will default from the price list selected on the Return.

3. Creating a Return using the Copy Function (RMA with Credit)

Select:	**Order Management** responsibility
Navigate to:	Orders, Returns > Order Organizer
Query:	Query an existing order or return to copy from.
Select:	**Actions** button in the Order Organizer window
Select:	Copy

Quick Copy Tab:

Select:	Create New Order
Change Order Type To:	Select RMA Order with Credit
New Order Number:	Enter a new RMA number for RMA order types that require manual numbering.

Copy Header Tab

To exclude child entities (lines, sales credits, notes, descriptive flex, and holds) or to re-price, navigate to the **Copy Header**, **Copy Line**, and **Pricing Options** tabs and deselect options as desired.

Note:	The **OM: Credit Card Privileges** profile option determines whether you are able to copy customer credit card information.

Copy Lines Tab

Change Line Type To:	Select RMA Line with Credit
Return Reason Code:	Select a return reason from the list of values.
Include Lines:	Includes the lines from the original order/return.
Include Descriptive Flex:	Includes the descriptive flexfield values from the original order/return.
Include Attachments:	Includes the attachments from the original order/return.
Include Fully Cancelled Lines:	Determine whether to include/exclude fully cancelled lines when using the copy feature. If fully cancelled lines are included, the lines are copied over with the original ordered quantity.

A common use of the **Copy** function is in the case where a customer wants to return all or part of a previous sales order. You may use the **Copy** function to create the return based directly on the information contained in the original sales order.

Another advantage of using the Copy function to create your RMAs is in the case where the customer will be receiving a credit for the return, Oracle can use the original sales order number to identify the original invoice in Accounts Receivable, and apply the credit directly against the original invoice.

When creating returns for configurations, copy the model line. Select the specific order lines and copy them as return lines to return individual components of a PTO configuration.

Pricing Tab

At Original Selling Price:	Select this option if you want the return to contain the original selling price in the originating order or return. Retaining the original pricing will retain all discounts and charges and the Calculate Price Flag is set to 'Partial' for return lines.
Re-price as of this date:	If you choose to re-price, specify the pricing date. Manual discounts and charges are removed and automatic discounts and charges are recalculated.
Select:	**OK** button. This will perform the copy and close the window.

If any validation errors occur, message(s) in the Messages window are displayed and indicates that an order was successfully created.

Select:	**Continue** button. The newly copied order is available through Order Organizer.

To update and book the RMA, select the RMA from *Today's Orders* in the Order Organizer window.

Select:	**Open** button.

- The original sales order from which this RMA was created is identified both at the header level (in the *Order Source* field of the Others tab) and at the line level (in the Order Source field of the Main tab).

You have the option to manually make changes to this RMA before booking it. For example, the customer may only want to return part of one line or not return another line at all.

You may optionally update the Receive From and Credit To Addresses using the Addresses Tab in the Line Items Tab.

Under the **Actions** button, there are several other options:

Promotions/Pricing Attributes – You may optionally apply Discounts to each lines at this time (assuming that Discounts have been defined and you have the appropriate discounting privileges). A Discount will decrease the amount of the credit the customer will receive.

Return Lot/Serial Numbers – You can enter lot and serial numbers for the return.

Sales Credits – If the Sales Credits button was checked in preparing the Copy then Sales Credits for the return will be derived from the original order. You may change the Sales Credits for the return if you wish by using this option.

To book the order, select the **Book Order** button.

4. Creating a New Return (RMA no Credit)

Select:	**Order Management** responsibility
Navigate to:	Orders, Returns > Order Organizer or Orders, Returns > Sales Orders
Select:	**New Order** button if using the Order Organizer

The process for creating an RMA no Credit is identical to creating an RMA with Credit. You have the option to create the RMA using the New Return option or the Copy option. The only difference between the two processes is that the Invoice Interface does not exist in the workflow for an Order Type of RMA no Credit. As a result, no credit memo will be created for this RMA.

Note:	Oracle does not provide a seeded workflow process to handle RMAs with Receipt no Credit; therefore, the R2i control environment provides a custom process to fulfill this need. For further information on this custom process refer to **OM Transaction Types Setup** and **R2i OM Order Line Workflow Package**.

5. Creating a New Return (RMA Credit Only)

Select:	**Order Management** responsibility
Navigate to:	Orders, Returns > Order Organizer or Orders, Returns > Sales Orders
Select:	**New Order** button if using the Order Organizer

The process for creating an RMA Credit Only is identical to creating an RMA with Credit. You have the option to create the RMA using the New Return option or the Copy option. The only difference between the two processes is that the Fulfillment activity does not exist in the workflow for an Order Type of RMA no Credit. As a result, no physical return of product is required.

6. RMA Receipt

Select:	**Purchasing Super User R2i** responsibility
Navigate to:	Receiving > Receipts
Select:	The Inventory Organization for the Receipt (not Global).
Select:	**OK** button.

Select:	The **Customer** tab in the *Find Expected Receipts* window.
RMA Num:	Optionally enter a specific RMA number.
Line Num:	Optionally enter a specific line number on a specific RMA. Note: Can only enter a line number if you have enter a number in the RMA Num field.
Line Type:	This field will populate automatically if you enter a value in RMA Num. If you do not enter a value in RMA Num you can optionally select a line type.
Customer:	Optionally select a customer from the LOV. If you enter a value in RMA Num, this field will populate automatically.
Customer Num:	Optionally select a customer number from the LOV. If you enter a value in RMA Num, this field will populate automatically.
Customer Item Num:	Optionally select a customer item number from the LOV.

- You can further search for expected receipts using the Item, Date Ranges, and Shipments tabs.

Select:	**Find** button.

All the receipt lines that meet the search criteria are displayed in the Receiving Transaction form.

Only lines with a “Destination Type” of “Inventory” can be delivered to Inventory.

Select:	Checkbox next to receipt line to be delivered to inventory.
Quantity:	Enter the “Quantity” to be delivered.
Subinventory:	Enter the subinventory where the items will be delivered to.

Save. Once the transaction is saved a receipt number is assigned.

For more information on Receiving Transactions in Purchasing refer to related R2i Purchasing Training documentation.

7. Viewing the Status of an RMA

The Sales Orders window displays the RMA header status in the Main tab of the Order Information tabbed region. The RMA line status is displayed in the Main tab of the Line Items tabbed region.

The *Workflow Status* option on the Sales Order window Tools menu launches the workflow status page. The window shows all the activities an RMA header or line has completed and the corresponding results in tabular format.

Note:	In order to view workflow status from the Order Organizer the menu attached to the responsibility in use must have two functions assigned to it: Monitor Activities List and Workflow Status. For more information see the appropriate AOL documentation.

8. Close the RMA

Closing RMAs that are complete enhances performance, since many programs, windows and report queries retrieve open RMAs only.

An RMA line is closed automatically once it has completed its corresponding workflow successfully. An RMA header is closed at the end of the month once all the lines have been closed.

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

Drop Ship Cycle in Order Management

The below are the steps involved in the Drop Ship Cycle:

Before you create an order, you have to perform some setups in order to drop ship, which are listed in the below mentioned article "Drop Ship Setups":

Drop Ship Setups [http://www.erpschools.com/Apps/oracle-applications/articles/Manufacturing/Order-Management/Drop-Ship-Setups/index.aspx]
==========SETUPS OF DROP SHIPMENT==========

To create a drop ship order the item should be setup as below.

Navigation: Inventory >> Items >> Organization Items

Select the organization.Click OK.

OK.

Enter Item Number click Find button

Main:

Item Status should be active.

Inventory

Purchasing

Receiving

General Planning

Order Management

==========END OF SETUPS OF DROP SHIPMENT==========

1. Create Sale Order

2. Book Sales Order

3. Check Status

4. Progress Sales Order

5. Check Status

6. Release Purchase Order

7. Import Requisition / Purchase Order

8. Link between sales order and purchase order

9. Receive the material against purchase order

10. Check Order status.

Create Sale Order

Navigation: Order Management Super User >> Order Organizer

Click on 'New Order'

Enter Order Header Information

Click on 'Line Items' tab

Click the 'Shipping' tab and enter Item Number , Quantity and Receiving Organization

Click 'Book Order' button.

If order is booked successfully then a confirmation message will be displayed as shown in the below picture.

Click on 'Actions' Button under 'Line Items' tab

Select 'Additional Line Information' from the List of values and click OK

Select 'Drop Ship' tab.

At this stage we do not have any purchase orders created related to this drop ship order.

Close the 'Additional Line Information' form.

Make a note that the line status is 'Booked' at this stage.

Let's see the workflow status.

Click on Tools >> workflow status

Current activity will be ' Purchase Release Eligible' with a status of 'Notified'.

Close the workflow status page.

Go back to 'Line Items' and select the line. Right click on it and select 'Progress Order' option as shown below.

Select 'Purchase Release – Eligible' option from List of Eligible Activities.

Click OK.

The 'Line Status' changes from 'Booked' to 'Awaiting Receipt'.

Click on Actions button

Select 'Additional Line Information'.

Make a note that we still do not have 'Purchase Order' created for this drop ship order.

Close the order form.

Navigation: Order Management Super User >> Purchase Release

A Concurrent request submit screen will pop up.

Click on Parameters and enter the sales order number that we created above. By doing this concurrent request will just process this particular order instead of releasing all pending drop ship order requests.

Click Submit

Close all open forms.

Navigation: Purchasing Super User >> Reports >> Run

Select 'Requisition Import' program and click on parameters text box.

Enter parameters as follows

Import Source: ORDER ENTRY

Import Batch ID: Leave it blank

Group By: Item+

Last Requisition Number: Leave it blank

Multiple Distributions: No

Initiate Approval after ReqImport: Yes

Click OK.

Wait until two concurrent requests 'Requisition Import' and 'Create Releases' are completed.

Now go back Order Management Super user responsibility

Click on Order Organizer and enter the sales order number.

Click find to open the sales order.

Select Line Items >> Actions >> Additional Information.

At this stage a purchase order is created related to this sales order.

Purchase order is in 'Approved' status because we initiated the approval process by setting the parameter 'Initiate Approval Process after ReqImport' to yes in the above concurrent request.

If you have EDI / e-commerce gateway setup to drop the purchase order information to supplier, it should trigger now.

Now Supplier has the purchase order information.

Supplier ships the parts to customer and will send us back the confirmation.

Once the confirmation is received we need to create a receipt.

Navigation: Inventory >> Transactions >> Receiving >> Receipts.

Select the Receiving Organization that is selected on shipping tab of line items form.

Click the purchase order number which is found on the additional line information form or the sales order that is created.

Click find.

The Receipt form should open with the item information as shown below.

Check mark the Left checkbox and if the item is Lot or Serial controlled then click on ' Lot – Serial' button.

Enter the lot number. In my case the item being used is lot controlled.

Click done.

Save and close the form.

Go back to Order Management and check the line status on the order form.

Now the line is shipped. To close the line run the 'workflow background process' concurrent program.

Once the workflow background process completed the line status should change from 'shipped' to 'closed'.

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

Display and change images dynamically in XML publisher

Display and change images dynamically in XML publisher :

This article discusses about how to display images dynamically using XML Publisher.

COMPANY NAME			
ADDRESS			
REPORT TITLE			
CF_PRINTED			
1.	2.	3.	
Customer No..	**Customer Name**	**Address**	**Tax Category**
Number	Name	Address	Tax Category

Note :- We have to place Dummy Images on template like in above template.

1) In this option we placed **image path** Directly like as follows

By right clicking on Dummy image select **SIZE** option (office 2007) and click on **Alt text** Button the following screen will display.

2) In this option we **get the image file** name in CF.

a) By right click on Dummy image select **SIZE** option (office 2007) and click on **Alt text** Button the following screen will display

3) In this option we get the **Image File path** From CF.

b) By right click on Dummy image select **SIZE** option (office 2007) and click on **Alt text** Button the following screen will display

OUTPUT:-

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

Data loader

Introduction:

Data loader is an utility where you can load data into different Form Based systems especially Oracle APPS. This simple utility works by recording keystrokes that are necessary for loading data from frontend.

Software be downloaded free from
http://www.dataload.net/downloads/download.php

Advantages:

- Easy to learn and use
- Can save time for repetitive processes
- Can be copied from Excel
- Can be an option to data edits and complex interfaces if the data are simple

Disadvantages:

- Cannot track mouse movements.
- Cannot perform interactive sessions to much extent.
- Do not generate a success failure logs.

List Of Commands:

DataLoad Command	Application [http://erpschools.com/Data_Load_Tutorial_Oracle_Apps.asp] Action(s)

TAB	<Tab>
ENT	<Enter>
*UP	<Up Arrow>
*DN	<Down Arrow>
*LT	<Left Arrow>
*RT	<Right Arrow>
*SP	Save & Proceed
*FE	Field Editor
*PB	Previous Block
*NB	Next Block
*PF	Previous Field
*NF	Next Field
*PR	Previous Record
*NR	Next Record
*ER	Erase Record
*DR	Delete Record
*FR	First Record
*LR	Last Record
*SAVE	Save Record
*SB	Send a single space character
*ST	Select entire field text.
*SL*N* or *SL(*N*)	Pause for N seconds. **Note 1**
*BM	Block Menu
*A*X*	Alt + X where X is a single letter (A-Z). **Note 2**
*FI	Find **+**
*FA	Find All **+**
*QE	Query Enter **+**
*QR	Query Run **+**
*CL	Clear Field **+**
*IR	Insert record **+**
*CW(*window*)	Change to window *window*. **+**
*ML(*coordinates*)	Position the mouse at *coordinates* and press the left button. ++
*MR(*coordinates*)	Position the mouse at *coordinates* and press the right button. ++
\^{f4}	Closing a window
*CW(*window*)	Make *window* the new target window for DataLoad.
*SL*N* or *SL(*N*)	Sleep for *N* seconds.
*ML(*coordinates*)	Position the mouse at *coordinates* and press the left button.
*MR(*coordinates*)	Position the mouse at *coordinates* and press the right button.
*DL(*coordinates*)	Position the mouse at *coordinates* and double click the left button.
PROMPT(*message*)	Prompt the user with *message* and wait for a response
BACKSPACE	{BACKSPACE}
DELETE	{DELETE}
UP ARROW	{UP}
DOWN ARROW	{DOWN}

LEFT ARROW	{LEFT}
RIGHT ARROW	{RIGHT}
END	{END}
ENTER	{ENTER}
TAB	{TAB}
ESC	{ESC}
HOME	{HOME}
PAGE DOWN	{PGDN}
PAGE UP	{PGUP}
INSERT	{INSERT}
Toggle Numlock1	{NUMLOCK}
Prnt Scrn2	{PRNTSCRN}
F1	{F1}
F2	{F2}
F3	{F3}
F4	{F4}
F5	{F5}
F6	{F6}
F7	{F7}
F8	{F8}
F9	{F9}
F10	{F10}
F11	{F11}
F12	{F12}
F13	{F13}
F14	{F14}
F15	{F15}
F16	{F16}

Note 1 DataLoad can send keystrokes to applications faster than they can be processed. If this problem is encountered, delays can be added to the load which will pause DataLoad at key times. The *SLN command can be added to the spreadsheet to indicate DataLoad should 'sleep' for a given number of seconds. E.g. '*SL5′ will cause a delay in processing for 5 seconds. Decimal numbers can be used for more precise delays, E.g. *SL0.5 will result in a half second delay. A large number of predefined delays are available in DataLoad and these, along with *SL, are described in greater detail in Using delays. To reduce setup work, predefined delays should be used instead of *SL wherever possible. **Note 2** In Oracle Applications it is sometimes necessary to press a button to navigate to another block. This can be achieved by pressing <Alt-X>, where X is the letter that is underlined on the button. Any menu item can also be invoked by pressing <Alt> + the letter underlined on the menu.

To use any combination of the Shift, Control, Alt and right Alt keys one of the following codes should be used. If you want to send the +, ^, % or & keys these characters test must be enclosed in braces {}.

Key	Code
SHIFT	+

CTRL	^
ALT	%
Right Alt	&

Case Study:
Granting "Application Developer" responsibility to 3 users.
Process:

1. The following example would show how to assign 'Application Developer" responsibility to the users USER1, USER2 and USER3
2. Try to record the process of assigning the responsibility to an user through key-strokes only.
3. Record the Keystrokes in terms of Data Load commands.
4. Note them sequentially to create the Dataload file (.dlt) as shown in the screenshot below.

5. Execute the Data Load being choosing the right window and command group

6. Be absolutely sure no other window becomes active during the process of data loading.

After completion of data load the window shows the final status.

The sample data file is also attached along with.

Click here to download

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012 Technical Terms of Oracle APPS

Story

The below example explains a few of the important terms and concepts used in the Oracle E-Business Suite. This would be a good starting point for the beginners to better understand the concepts behind Oracle Applications.

Say **Harry** is the owner of a wholesale fruit shop. He buys various fruits like apples, oranges, mangos and grapes etc from farmers directly and sells them to retail shop owners and also to the direct customers.

The farmers are referred to as **VENDORS/SUPPLIERS** in Oracle Applications**.** Harry keeps track of all his vendors' information like addresses, bank account and the amount he owes to them for the fruits that he bought etc, in a book named **PAYABLES**.

Harry gets an order from a retail shop owner of Fruit Mart, for a shipment of 11 bags of apples, 25 bags of oranges and 32 kgs of grapes. In Oracle Apps, bags and kgs are referred to as **UOM** (unit of measure), Fruit Mart is called **CUSTOMER** and the order is referred to as **SALES ORDER.** Harry maintains a book called **ORDER MANAGEMENT** where he writes down all the details of the **SALES ORDERS** that he gets from his customers.

Say the fruits have been shipped to the customer Fruit Mart. Harry now sends him the details like cost of each bag/fruit, the total amount that the customer has to pay etc on a piece of paper which is called **INVOICE / TRANSACTION.** Once the INVOICE has been sent over, the customer then validates this against the actual quantity of fruits that he received and will process the payments accordingly. The invoice amount could be paid as a single amount or could be paid in installments. Harry's customer, Fruit Mart pays him in installments (partial payments). So Harry has to make a note of the details like date received, amount received, amount remaining, amount received for what goods/shipments/invoice etc, when Harry receives the payments. This detail is called **RECEIPT**, which will be compared to the invoice by Harry to find how much Fruit Mart has paid to him and how much has to be paid yet. This information is maintained in a book named **RECEIVABLES** to keep track of all the customers, their addresses (to ship the items), what and how much he has shipped to his customers and the amount his customers owe him etc**.**

Harry's fruit business has begun to improve and has attracted more and more customers. As a result, Harry decided to buy a cold storage unit where he could stock more fruits. In Apps, this cold storage unit is known as **WAREHOUSE** and all the fruits are referred to as **INVENTORY.** Due to increase in customers, Harry needs to hire more people to help him out in his business without any hiccups. These workers are called **EMPLOYEES.** At the end of every month, Harry pays the salary for all his employees through Checks. These checks are nothing but **PAYROLL** in Apps.

At the end of every month, Harry prepares a balance sheet in a book called **GENERAL LEDGER** to determine how much profit/loss he got and keeps track of the money going out and going in.

As the business grows, it becomes impossible to record everything on a paper. To make everybody's life easier, we have very good tools in the market, which help the business men to keep track of everything. One such tool is Oracle E-Business Suite.

Oracle Applications is not a single application, but is a collection of integrated applications. Each application is referred to as a module and has it own functionality trying to serve a business purpose.

Few of the modules are Purchasing, Accounts Payables, Accounts Receivables, Inventory, Order Management, Human Resources, General Ledger, Fixed Assets etc.

Here is a high level business use of various modules:

Oracle Purchasing handles all the requisitions and purchase orders to the vendors.

Oracle Accounts Payables handles all the payments to the vendors.

Oracle Inventory
deals with the items you maintain in stock, warehouse etc.

Order Management helps you collect all the information that your customers order.

Oracle Receivables help you collect the money for the orders that are delivered to the customers.

Oracle Human Resources helps maintain the Employee information, helps run paychecks etc.

Oracle General Ledger receives information from all the different transaction modules or sub ledgers and summarizes them in order to help you create profit and loss statements, reports for paying Taxes etc. For Example: when you pay your employees that payment is reported back to General Ledgers as cost i.e money going out, when you purchase inventory items and the information is transferred to GL as money going out, and so is the case when you pay your vendors. Similarly when you receive items into your inventory, it is transferred to GL as money coming in, when your customer sends payment, it is transferred to GL as money coming in. So all the different transaction modules report to GL (General Ledger) as either “money going in” or “money going out”, the net result will tell you if you are making a profit or loss.

All the equipment, shops, warehouses, computers can be termed as ASSETS and they are managed by **Oracle Fixed Assets**.

There is a lot more in Oracle applications. This is the very basic explanation just to give an idea of the flow in ERP for the beginners.

Terminology often used in Oracle Applications:

1. Invoice
2. Receipt
3. Customer
4. Vendor
5. Buyer
6. Supplier
7. Purchase Order
8. Requisition
9. ACH: Account Clearance House
10. Sales Order
11. Pack Slip
12. Pick Slip
13. Drop Ship
14. Back Order
15. ASN: Advance Shipping Notice
16. ASBN: Advance Shipping Billing Notice
17. ATP: Available to Promise
18. Lot/Serial Number

19. DFF: Descriptive Flex Fields
20. KFF: Key Flex Fields
21. Value Sets
22. Organization
23. Business Unit
24. Multi Org
25. Folders
26. WHO Columns
27. Oracle Reports
28. Oracle Form
29. Workflow Builder
30. Toad
31. SQL Developer
32. SQL Navigator
33. Discoverer Reports
34. XML/BI Publisher
35. ADI: Application Desktop Integrator
36. Winscp
37. Putty

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

Discoverer

Oracle Discoverer is a business intelligence tool to support organizational decisions and data will show in the form of excel format.

Components of discoverer:-

1.Discoverer Adminstration Edition

2. Discoverer Desktop Edition

Architecture Of Discoverer Administartion Edition:-

i.End User Layer

ii.Business Area

iii.Business Folders

Overview of Business Areas:-

-A business area is a collection of related information in the database.

-A business area is a set of related information with a common business purpose

-For example, information about Sales may be stored in one business area, while information about Cops is stored in another business area.

-In simple words it can be termed as collections of objects in a particular module

Overview of Business Folders:-

-Simple Folders – Folders that are based on a database table (e.g.: ITEM)

-Custom Folders – Folders that contain a custom SQL query.

-Complex Folders – Folders that are based on multiple simple folders.

Here are the steps for creating the 'Business area'

Open Discoverer Administrative Edition

Logon to Discoverer Administrative Edition using SYSADMIN user

Click Connect

Choose a Responsibility and Click OK

Click Create a New Business Area and Click Next

Select Any User and Click Next

Expand the Node and Select Any Table or View And Click Next

Click Next

Name the Business Area and Description Appropriately And Click Finish

The Business Area Will be created and you would view the following screen

Close the Administrative Tasklist Window

Expand the Business Area

Delete The Folder under the Business Area

Click Yes

Now the business Area is Empty

Logon to SQL Plus and Create a View according to the requirement

Relogon to Discoverer Administration Edition to have the Schema Refreshed And Open the Business Area Created Earlier.

Right Click on the Business Area Created and Select the New Folder from Database Option

Click Next

Select the Schema APPS and Click Next as shown below

Expand the APPS Schema

Select the View Created at SQL Plus and Click Next

Click Finish

The Folder is Created

Expand the Business Area and you can view the Folder

Click Tools → Security Menu

Assign the Users/Responsibilities who can access the Business Area and Click OK

Here are the screen shots for creating the workbooks in the Discoverer

Desktop:-

Logon to the Discoverer Desktop Edition to create Work Books

Login as SYSADMIN User

Select System Administrator Responsibility and Click Ok

Select Create a new workbook option

Select one of the Display Style as per the requirement and Click Next

Select the Business Area and the folder on which you would like to create the Workbook and Click Next

Check Show Page Items and Click Next

You could add the condition required by clicking New.

Select New Parameter Option from the List

You will get the following screen

Enter the Name, Prompt, Description and other Fields and Click OK

Click Ok

Click Next

You can create the Sort Condition so that the Workbook would sort the data accordingly.

Click Add

Select the Field on which you would like to sort the data and Click ok

Add as many sorts you need

Click Finish

You workbook is Created.

Go to File → Managewokbooks → Properties

Give the Identifier and Description

Click Ok

Click Yes

Go to Sheet → Rename Sheet Menu

Give an Appropriate Name and Click ok

Click Save

Select Database

Give an appropriate name and Click Save

Goto File → Manageworkbooks → sharing menu

Select the Workbook and assign it to the responsibility who can access the workbooks as shown in the screen

Click Ok

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

Delimited Report Output using Report Builder

Delimited Report Output using Report Builder

Overview:

In this tutorial, we will see how to customize the existing standard Oracle report to get a delimited output file instead of the regular layout. Most business users prefer delimited report output as it can easily be imported into Excel where they can manipulate and perform calculations on the data easily.
Report Builder is the tool used to develop/customize Oracle reports. Before getting into the details, I would like to give an overview about Report Builder.
Main Components of a Report Builder:
Below is the snapshot of the Object navigator when you open a report in Report Builder. Some important components are Data Model, Layout Model, Parameter form, Triggers. Let's discuss about each one of them in detail.

- Data Model:
  The Data Model is a work area in which you define what data to retrieve when the report is submitted. You create and define queries, groups, columns, parameters, and links which are called data model objects to determine what data needs to be extracted from the database as part of a report.

Tool Palette in Data Model view:

#	Object	Object Name	Description
1		Select	To Select objects for an operation
2		Magnify	To Magnify an area
3		SQL Query	To create a new query in the data Model
4		RefCursor Query	To Create a new Ref Cursor query
5		Summary Column	To create a summary column. A summary column performs a computation like sum, average, count, minimum, maximum, % total on another column's data.
6		Formula Column	To create a Formula column. A formula column performs a user-defined computation on another column's data
7		Cross Product	To create a Cross Product group
8		Data Link	To create relationship between two queries in the data model
9		Placeholder Column	To Create a Placeholder column. A placeholder is a column for which you set the datatype and value in PL/SQL that you define. You can set the value of a placeholder column in the following places: – Before Report Trigger, if the placeholder is a report-level column – report-level formula column, if the placeholder is a report-level column –a formula in the placeholder's group or a group below it (the value is set once for each record of the group)

- Layout Model:
  Layout Model is a work area in which you can define the format (using objects like frames, repeating frames, fields, boilerplate, anchors, and Graphics objects) of your report output. When you run a report, Report Builder uses the Layout Model as a default template for the report output.

Layout Tool Palette:

#	Object	Object Name	Description
1		Select Tool	To select one or more objects
2		Frame Select	To select frame or repeating frame and all the objects within them.
3		Rotate	To rotate Objects
4		Reshape	To change the shape of the objects
5		Magnify	To magnify the area

6	Line	To draw a line
7	Rectangle	To draw a rectangular object
8	Rounded Rectangle	To draw a rounded rectangular object
9	Ellipse	To draw a elliptic object
10	Arc	To draw arc
11	Polyline	To create a Polyline object
12	Polygon	To draw a polygon object
13	Text	To create a Text object
14	Freehand	To create a free form obk=ject
15	Frame	To create a frame. Frames are used to surround other objects and protect them from being overwritten or pushed by other objects
16	Repeating Frame	To create a repeating frame. Repeating frames surround all of the fields that are created for a group's columns meaning each repeating frame must be associated with a group created in the Data model. The repeating frame prints (is fired) once for each record of the group.
17	Link file	To create an object that is read in from file
18	Field	To create a field
19	Chart	To create a Chart
20	Button	To create a button
21	Anchor	To create an anchor between two objects. Since the size of some layout objects may change when the report runs, you need anchors to define where you want objects to appear relative to one another.
22	OLE2	To create OLE2 object

- Parameter Form:

Parameter Form enables you to define the parameters for your report.
Tool Palette:

- Report Triggers:
  Report triggers execute PL/SQL functions at specific times during the execution and formatting of your report. Report Builder has five global report triggers:
  - After Parameter Form trigger:
    This trigger fires after the Parameter form is displayed.
  - After Report trigger:
    This trigger fires after the report output is displayed. You can use this trigger to delete any temporary values or tables created during the process.
  - Before Parameter Form trigger:
    This trigger fires before the Parameter form is displayed.

- Before Report trigger:
  This trigger fires before the reports is executed but after queries are parsed and data is fetched.
- Between Pages trigger:
  This fires before each page of the report is formatted, except the very first page. This can be used to display page totals etc.

Report Customization steps:

When you have to customize a standard report, it is always advisable not to make changes to the standard report itself, instead rename it to another report and make the changes to it.
Steps:

- Download the original rdf from the file system.
- Open it in Report builder.
- Save it with a different name that meets the client's naming conventions.
- Make the necessary changes to it.
- Save and compile the report.
- Move it to the custom top/reports/US
- Register it as a concurrent program in Oracle Applications under custom application.

Steps to change the report output to a Pipe delimited output:
The requirement here is to customize the standard "Receipt Adjustment report" to get a delimited output file instead of the regular layout. The output file should have header information like shown below on the top of the output page followed by the receipt adjustment data whose fields are separated by '~'.

```
Vendor Name~PO Number~PO Line~Po Line Description~Item
Number~Category~Organization~Ship To Location~Qty Ordered~Net Qty
Received~Qty Billed~Qty Accepted~Qty Rejected~Qty Cancelled~Received
Qty Corrected~Net Qty RTV~Qty RTV Corrected~Receipt Num~Transaction
Date~Transaction Type~Parent Transaction~Transaction amount~Unit
```

Regular Layout generated by the standard report:

We need Pipe delimited Output file like the below:

To achieve this:

- We have to get rid of the current layout and create a new layout with 2 objects:
  Frame to print the header information.
  Repeating Frame to print the data.

- We need to create a Formula column in the Data model that will get the concurrent program's output filename. We will use this file to write our pipe delimited report output to.

Steps:

1. Download the original report POXRVRTN.rdf

2. Open the report in the Report Builder. File>Open
3. Rename it according Custom naming conventions followed by the client. Here we will rename it to XXERP_POXRVRTN

   Tools> Property Palette

   Give the name as: XXERP_POXRVRTN

1. To Create a Formula column to derive the output file name:

   - Double click on the Data model in the Object Navigator.
   - Click in the tool palette
   - Click and drag a rectangle.
   - Double-click the formula column created in the data model to open up its property palette where you can set its properties.

     Name: Give the name as C_OUTPUT_FILE

     Data Type: Choose Character

     Width: 300

     PL/SQL Formula: Insert the below code which gets the Concurrent program's output filename from the database.

```
function C_OUTPUT_FILEFormula return Char is
v_filename fnd_concurrent_requests.outfile_name%type;

begin
SELECT outfile_name
INTO v_filename
FROM fnd_concurrent_requests
WHERE request_id =  _CONC_REQUEST_ID;
RETURN(v_filename);
exception
when others then
RETURN(null);
end;
```

1. Double click on the Layout model in the Object Navigator.
2. Remove all the objects placed in the layout model except "No Data Found" Object.
3. Place a Frame and a repeating frame one below the other as shown below.

- To place a frame in the Layout:

  Click in the tool palette.

  Click and drag a rectangle.

  Double-click the frame object in the layout to open up its property palette where you can set its properties.

Some important properties are discussed here.

- Name: Rename it to whatever you want.
- Vertical and Horizontal Elasticity: For frames and repeating frames, elasticity defines whether the size of the frame or repeating frame should vary with the objects inside of it.

Possible Values that you can enter are Contract, Expand, Fixed, and Variable.

- Contract means the vertical (for vertical elasticity) or horizontal (for horizontal elasticity) size of the object decreases, if the formatted objects or data within it are short (for vertical elasticity) or less wide (for horizontal elasticity) enough, but it cannot increase to a height (for vertical elasticity) or width (for horizontal elasticity) greater than that shown in the Report Editor.
- Expand Means the vertical (for vertical elasticity) or horizontal (for horizontal elasticity) size of the object increases, if the formatted objects or data within it are tall or more wide enough, but it cannot decrease to a height or width less than that shown in the Report Editor.
- Fixed Means the height or width of the object is the same on each logical page, regardless of the size of the objects or data within it. Truncation of data may occur.
- Variable Means the object may expand or contract vertically to accommodate the objects or data within it (with no extra space), which means the height or width shown in the Report Editor has no effect on the object's height or width at runtime.

- To place a repeating frame in the Layout:

  Click in the tool palette.
  Click and drag a rectangle.
  Double Click on Repeating Frame to open up the property palette and rename it. Every repeating frame must be associated with a group defined in the Data model.
  Here give the Source as "G_shipment_lines".
  Set the Vertical and horizontal elasticity to the required.

1. To print a pipe delimited text in the output file, we will use a format trigger on the frame and repeating frame.
   A **format trigger** is a PL/SQL function executed before an object is formatted. This function must return a Boolean value (TRUE or FALSE). Depending on whether the function returns TRUE or FALSE, the current instance of the object is included or excluded from the report output. Format trigger can be used to highlight a value, for suppressing values and labels.

In the property palette of the Frame, under Advanced Layout section:

Double Click on the Format Trigger. This opens up a SQL Editor, where you can place the below code to print the header information to the output file.

```
function M_SHIPMENT_LINE_HDRFormatTrigg return boolean is
–Variable declaration
cmd_line VARCHAR2(3000);
v_file_name text_io.file_type;
begin
–Setting cmd_line variable to the header info
        cmd_line := 'Vendor Name'||'~'||'PO Number'||'~'||'PO Line'||'~'||'Po Line
        Description'||'~'||'Item Number'||'~'||'Category'||'~'||'Organization'||'~'||'Ship
        To Location'||'~'||'Qty Ordered'||'~'||'Net Qty Received'||'~'||'Qty
        Billed'||'~'||'Qty Accepted'||'~'||'Qty Rejected'||'~'||'Qty
        Cancelled'||'~'||'Received Qty Corrected'||'~'||'Net Qty RTV'||'~'||'Qty RTV
        Corrected'||'~'||'Receipt Num'||'~'||'Transaction Date'||'~'||'Transaction
        Type'||'~'||'Parent Transaction'||'~'||'Transaction amount'||'~'||'Unit';
–Opening the concurrent request's output file to write the data into it
        –Always prefix ":" with the field,when you refer to a field in the data model
        like
        :C_OUTPUT_FILE

v_file_name := TEXT_IO.FOPEN(:C_OUTPUT_FILE, 'A');
IF TEXT_IO.IS_OPEN(v_file_name) THEN
TEXT_IO.PUT_LINE(v_file_name, cmd_line);
END IF;
TEXT_IO.FCLOSE(v_file_name);
–If the return value is true then only this object will be included in the report output
return (TRUE);
end;
```

Similarly include the below code in the format trigger of the repeating frame to write the receipt records into the output file.

```
        function R_shipment_linesFormatTrigger return boolean is
        cmd_line VARCHAR2(2000);
        v_file_name text_io.file_type;
        begin
                cmd_line :=
                :Source||'~'||:Document_Number||'~'||:Line||'~'||:Description||'~'||:C_
                FLEX_ITEM_DISP||'~'||:C_FLEX_CAT_DISP||'~'||:Organization_na
                me||'~'||:Ship_To_Location||'~'
                ||:Quantity_Ordered||'~'||:C_qty_net_rcvd||'~'||:qty_billed||'~'||:qty_a
                ccepted||'~'||:qty_rejected||'~'||:qty_cancelled||'~'||:C_qty_corrected|
                |'~'||:C_qty_rtv_and_corrected||'~'||:C_qty_corrected_rtv||'~'||:Recei
                pt_Number||'~'||:Receipt_Date||'~'||:Transaction_Type||'~'
                ||:Parent_Transaction_Type||'~'||:Transaction_Quantity||'~'||:Transa
                ction_Unit;
        v_file_name := TEXT_IO.FOPEN(:C_OUTPUT_FILE, 'A');
        IF TEXT_IO.IS_OPEN(v_file_name) THEN
        TEXT_IO.PUT_LINE(v_file_name, cmd_line);
        END IF;
        TEXT_IO.FCLOSE(v_file_name);
        return (TRUE);
        end;
```

1. Now that the changes are done, save the report.
2. Connect to the database by navigating to File > Connect

1. Then compile the report by navigating to Program> Compile> All.
   Errors will be listed if there are any. Correct them and recompile. If there are no errors and the compilation was successful, you will get the below message. Click OK and save again.

1. Now move the report to the Custom top/Reports/US
2. Register it as a concurrent program in Oracle Applications and assign it to the desired responsibilities. Please refer to Concurrent Program registration article for registration details.

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

Handling multiple layouts in Xml Publisher (.Rtf)

Handling multiple layouts in Xml Publisher (.Rtf) :

Steps for handling multiple layouts in XML.

1. After developing the Report definition file (.rdf) we have to add one more parameter like follows.

2. This parameter value should be assigned to Place holder column(CP) like follows
   We can assign Parameter value either after parameter form or before report Triggers.

   In this we assigned in Before report Trigger like Below..

Note: – place holder column should be placed at Report level.

Then we can create multiple layouts in (.rtf).

Like below we have to add condition(If) Field for handling the multi layouts.

Double click on if condition which was added by ourselves. Then the following screen

Will display.

Click On **Add help text** Button

Posted 29th August 2012 by Krishnareddy

1 View comments

29th August 2012

XML Publisher

Overview: Oracle XML Publisher is a template-based publishing solution delivered with the Oracle E-Business Suite. It provides a new approach to report design and publishing by integrating familiar desktop word processing tools with existing E-Business Suite data reporting. At runtime, XML Publisher merges the custom templates with the concurrent request data extracts to generate output in PDF, HTML, RTF, EXCEL (HTML), or even TEXT for use with EFT and EDI transmissions

Basic Need for XML: Consider the following scenarios

We have a RDF report with tabular layout which prints in English

New Requirements:

1. User1 wants the same Report needs to be printed in Spanish
2. User2 wants the Same Report needs to be printed in chart format
3. User3 wants the Same Report output in Excel
4. User4 wants the Same Report output to be published on intranet or internet
5. User5 wants the Same Report output eliminating few columns and adding few other

A new RDF needs to be created for each requirement stated above or an existing RDF needs to be modified with huge amount of effort but whereas with XML Publisher it can be done very easily.

XML Publisher separates a reports data, layout and translation components into three manageable pieces at design time; at runtime all the three pieces are brought back together by XML Publisher to generate the final formatted, translated outputs like PDF, HTML, XLS and RTF. In future, if any there is any change in layout we just need to add/modify the Layout file

Data Logic Data extracted from database and converted into an XML string.

Layout – The layout templates to be used for the final output are stored and managed in the Template Manager.

Translation -The translation handler will manage the translation that is required at runtime

In brief the steps are as follows:-
a. Create a procedure and register it as Concurrent Program so that we write XML tags into output file.
b. Build a Data Definition & XML Template using XML Publisher.
c. Create a relation between XML Template & Concurrent Program and run the concurrent program

Requirements for XML Data Object Reports

1. Oracle XML Publisher Release 5.5 patch 4206181
2. Template Builder 5.5

Template builder is used to create template/layout for your report. Usually Template builder 5.5 is available in Oracle XML Publisher patch itself but you can also download it from http://edelivery.oracle.com. First select Oracle Application Server Products then select your platform and then locate the Oracle XML Publisher Release 5.6.2 Media Pack v1 for Microsoft Windows, as below:

Download the Desktop edition from the below:

When you download the XML Publisher Desktop edition you get a Zip file containing setup for XML Publisher Desktop Install Shield, this installs some components into Microsoft Word.

After installing, the Word Add-Ins is attached to the menu bar for the word document. This menu lets you attach an XML data source document, add the XML data to your template, set preferences and preview the output.

In detail along with screenshots:-

A concurrent program is written that spit out an XML file as output such concurrent program can be of type SQL or PL/SQL or Oracle Report or any other supportable type, provided it can produce a XML output.

1. Here I have a very simple PL/SQL procedure, which fetch the records from AR tables and write the output in xml tags.

```
CREATE OR REPLACE PROCEDURE APPS.Demo_XML_Publisher (errbuf
VARCHAR2,retcode NUMBER,v_customer_id VARCHAR2)

AS

/*Cursor to fetch Customer Records*/

CURSOR xml_parent

IS

SELECT   customer_name , customer_id

FROM      ra_customers

WHERE   customer_id = to_number(v_customer_id);

/*Cursor to fetch customer invoice records*/

CURSOR xml_detail(p_customer_id1 NUMBER)

IS

SELECT ra.customer_trx_id customer_trx_id, ra.ship_to_customer_id ship_to_customer_id,
ra.trx_number trx_number,aps.amount_due_original ams

FROM  ra_customer_trx_all ra, ar_payment_schedules_all aps

WHERE  ra.ship_to_customer_id = p_customer_id1

AND aps.customer_trx_id = ra.customer_trx_id

AND      ROWNUM<4;

BEGIN

/*First line of XML data should be <?xml version="1.0"?>*/

FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<?xml version="1.0"?>');

FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<CUSTOMERINFO>');

FOR v_customer IN xml_parent
```

```
LOOP

/*For each record create a group tag <P_CUSTOMER> at the start*/

FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<P_CUSTOMER>');

/*Embed data between XML tags for ex:- <CUSTOMER_NAME>ABCD</CUSTOMER_NAME>*/

FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<CUSTOMER_NAME>' ||
v_customer.customer_name

|| '</CUSTOMER_NAME>');

FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<CUSTOMER_ID>' || v_customer.customer_id ||

'</CUSTOMER_ID>');

FOR v_details IN xml_detail(v_customer.customer_id)

LOOP

/*For  customer invoices create a group tag <P_INVOICES> at the

start*/

FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<P_INVOICES>');

FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<CUSTOMER_TRX_ID>' ||

v_details.customer_trx_id || '</CUSTOMER_TRX_ID>');

FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<CUSTOMER_ID>' ||

v_details.ship_to_customer_id || '</CUSTOMER_ID>');

FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<INVOICE_NUMBER>'||

v_details.trx_number||'</INVOICE_NUMBER>');

FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'<AMOUNT_DUE_ORIGINAL>'||

v_details.trx_number||'</AMOUNT_DUE_ORIGINAL>');

/*Close the group tag </P_INVOICES> at the end of customer invoices*/

FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'</P_INVOICES>');

END LOOP;

/*Close the group tag </P_CUSTOMER> at the end of customer record*/
```

```
FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'</P_CUSTOMER>');

END LOOP;

/*Finally Close the starting Report tag*/

FND_FILE.PUT_LINE(FND_FILE.OUTPUT,'</CUSTOMERINFO>');

exception when others then

FND_FILE.PUT_LINE(FND_FILE.log,'Entered into exception');

END Demo_XML_Publisher;

/
```

2. Create an executable SampleXmlReport for the above procedure Demo_XMML_Publisher.

Go to Application Developer Responsibility->Concurrent->Executable

3. Create a new concurrent program SampleXmlReport that will call the SampleXmlReport executable declared above. Make sure that output format is placed as XML.

Go to Application Developer Responsibility -> Concurrent ->Program

4. Make sure we declare the parameters for the procedure.

5. Add this new concurrent program with Receivables request group. Either using the following code or through below application screen.

```
DECLARE
BEGIN
FND_PROGRAM.add_to_group
(
PROGRAM_SHORT_NAME  =>'CUST_XML_SAMPLE'
,PROGRAM_APPLICATION =>'AR'
,REQUEST_GROUP       => 'Receivables All'
,GROUP_APPLICATION   =>'AR'
) ;
commit;
exception
when others then
dbms_output.put_line('Object already exists');
```

END ;
/

Go to System Administrator Responsibility ->Security ->Responsibility->Request

6. From the receivables responsibility (depends on which responsibility we added our concurrent program here it is receivables)

From the menu View->Requests->Submit A New Request->Single Request

Note: The layout field is blank as we haven't attached any Template or layout to this concurrent program yet.

By submitting the above request we get the output in xml (depending on procedure) as follows:

<?xml version="1.0" ?>

- [http://oamdev.ventanamed.test:8000/OA_CGI/FNDWRR.exe?temp_id=2392214407]
<CUSTOMERINFO>

- [http://oamdev.ventanamed.test:8000/OA_CGI/FNDWRR.exe?temp_id=2392214407]
<P_CUSTOMER>

<CUSTOMER_NAME>**UNIV OF CHICAGO HOSP**</CUSTOMER_NAME>

<CUSTOMER_ID>**1119**</CUSTOMER_ID>

- [http://oamdev.ventanamed.test:8000/OA_CGI/FNDWRR.exe?temp_id=2392214407]
<P_INVOICES>

<CUSTOMER_TRX_ID>**929476**</CUSTOMER_TRX_ID>

<CUSTOMER_ID>**1119**</CUSTOMER_ID>

<INVOICE_NUMBER>**2484403**</INVOICE_NUMBER>

<AMOUNT_DUE_ORIGINAL>**8000**</AMOUNT_DUE_ORIGINAL>

</P_INVOICES>

- [http://oamdev.ventanamed.test:8000/OA_CGI/FNDWRR.exe?temp_id=2392214407]
<P_INVOICES

<CUSTOMER_TRX_ID>**929374**</CUSTOMER_TRX_ID>

<CUSTOMER_ID>**1119**</CUSTOMER_ID>

<INVOICE_NUMBER>**2484267**</INVOICE_NUMBER>

<AMOUNT_DUE_ORIGINAL>**380.68**</AMOUNT_DUE_ORIGINAL>

</P_INVOICES>

- [http://oamdev.ventanamed.test:8000/OA_CGI/FNDWRR.exe?temp_id=2392214407]
<P_INVOICES>

<CUSTOMER_TRX_ID>**806644**</CUSTOMER_TRX_ID>

<CUSTOMER_ID>**1119**</CUSTOMER_ID>

<INVOICE_NUMBER>**2421373**</INVOICE_NUMBER>

<AMOUNT_DUE_ORIGINAL>**615.96**</AMOUNT_DUE_ORIGINAL>

</P_INVOICES>

</P_CUSTOMER>

</CUSTOMERINFO>

7. Save the above code as SampleXmlReport.xml

Note: Before saving the XML string in notepad remove the dashes -

8. Create Template/Layout for the report using Template Builder. Here is a sample template.

Note the following:

The data fields that are defined on the template

For example: Customer Name, Customer Id

The elements of the template that will repeat when the report is run.

For example, Customer trx id , Invoice Number and Original Amount Due. All these fields on the template will repeat for each Employee that is reported.

9. Mark up your template layout.

Like a mail-merge document there's placeholders for the data you're going to add, and then you can add whatever formatting you like.

10. Now the next step is to select Data > Load XML Data from the toolbar menu, then pick up the

XML data a file ie.**SampleXmlReport.xml**. Once the data is loaded, a “Data Loaded Successfully” dialog box comes up and you can then start adding data items to the template.

11. To add data items from the XML file into your report, you locate in the document the placeholder for the field you’re going to add, highlight it and then select Insert > Field from the toolbar. A dialog box comes up with all of the available data items, you select the one you want and click insert as shown below:

12. You can add repeating rows into your document by selecting Insert > Table/Form from the toolbar. This brings up a different dialog box that lets you drag a parent node – in this case, “P Invoices” – into the middle section, which becomes your repeating rows.

13. Calculate the average for amount due original. Select Insert->Field from the Add Ins toolbar. Then select the tag that calculates the average for that particular field.

14. Once we are done with adding up all the fields in the template save it as an rtf which looks as below:

To confirm the output from the template we build. Click on preview and select the type in which format the output is required.

15. Adding the Template to ORACLE Application.

In order to add the template to application the user should have the responsibility XML Publisher Administrator assigned.

In this step we do 2 processes, registering concurrent program as Data Definition in template manager

And register the template using the data definition created.

Go to XML Publisher Administrator->Data Definitions->Create Data definition.

Here we fill all the details to create data definition

NOTE: Make sure the code of the data definition must be the same as the short name of the Concurrent Program we registered for the procedure. So that the concurrent manager can retrieve the templates associated with the concurrent program

We can add our xml file SampleXmlReport.xml in data definition here:

16. Now create the template with the help of template manager

At the runtime the concurrent managers request interface will present the list of available templates with respect to the data definition registered. We will upload the rtf template file created here. We will select the language and territory.

We can upload different templates for different languages and territories.

17. Now run the concurrent program to get the desired output.

From the receivables responsibility, use the submit request form to run the concurrent request.

The default layout is displayed which we can change as per the requirement. For changing the template click on options to look for all templates register with that concurrent program.

Here in the above example my template is SampleXmlReport which is displayed in layout field.

And once the request is submitted we will get the output in PDF as

This is the final output as desired in the PDF format.

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012 Get On Hand Quantities through API

Get On Hand Quantities through API :

This script can be used to get the below quantities.
1. On-hand Quantity
2. Available to Reserve
3. Quantity Reserved
4. Quantity Suggested
5. Available to Transact
6. Available to Reserve

You can also get the On-hand quantities from the table mtl_onhand_quantities

GET ON-HAND QUANTITIES API

```
DECLARE

   x_return_status         VARCHAR2 (50);

   x_msg_count             VARCHAR2 (50);

   x_msg_data              VARCHAR2 (50);

   v_item_id               NUMBER;

   v_org_id                NUMBER;

   v_qoh                   NUMBER;

   v_rqoh                  NUMBER;

   v_atr                   NUMBER;

   v_att                   NUMBER;

   v_qr                    NUMBER;

   v_qs                    NUMBER;

   v_lot_control_code      BOOLEAN;

   v_serial_control_code   BOOLEAN;

BEGIN

   -- Set the variable values

   v_item_id := '6566';

   v_org_id := 61;

   v_qoh := NULL;

   v_rqoh := NULL;

   v_atr := NULL;
```

```
v_lot_control_code := FALSE;

v_serial_control_code := FALSE;

-- Set the org context

fnd_client_info.set_org_context (1);

-- Call API

inv_quantity_tree_pub.query_quantities

(p_api_version_number       =&gt; 1.0,

 p_init_msg_lst             =&gt; 'F',

 x_return_status            =&gt; x_return_status,

 x_msg_count                =&gt; x_msg_count,

 x_msg_data                 =&gt; x_msg_data,

 p_organization_id          =&gt; v_org_id,

 p_inventory_item_id        =&gt; v_item_id,

 p_tree_mode                =&gt; apps.inv_quantity_tree_pub.g_transaction_mode,

 -- or 3

 p_is_revision_control      =&gt; FALSE,

 p_is_lot_control           =&gt; v_lot_control_code,

 -- is_lot_control,

 p_is_serial_control        =&gt; v_serial_control_code,

 p_revision                 =&gt; NULL,        -- p_revision,

 p_lot_number               =&gt; NULL,          -- p_lot_number,

 p_lot_expiration_date      =&gt; SYSDATE,

 p_subinventory_code        =&gt; NULL,    -- p_subinventory_code,

 p_locator_id               =&gt; NULL,          -- p_locator_id,

 -- p_cost_group_id            =&gt; NULL,       -- cg_id,
```

```
      p_onhand_source           =&gt; 3,
      x_qoh                     =&gt; v_qoh,      -- Quantity on-hand
      x_rqoh                    =&gt; v_rqoh,           --reservable quantity on-hand
      x_qr                      =&gt; v_qr,
      x_qs                      =&gt; v_qs,
      x_att                     =&gt; v_att,  -- available to transact
      x_atr                     =&gt; v_atr    -- available to reserve
     );
    DBMS_OUTPUT.put_line ('On-Hand Quantity: ' || v_qoh);
    DBMS_OUTPUT.put_line ('Available to reserve: ' || v_atr);
    DBMS_OUTPUT.put_line ('Quantity Reserved: ' || v_qr);
    DBMS_OUTPUT.put_line ('Quantity Suggested: ' || v_qs);
    DBMS_OUTPUT.put_line ('Available to Transact: ' || v_att);
    DBMS_OUTPUT.put_line ('Available to Reserve: ' || v_atr);
 EXCEPTION
    WHEN OTHERS
    THEN
       DBMS_OUTPUT.put_line ('ERROR: ' || SQLERRM);
 END;
```

GET ON-HAND QUANTITIES FROM TABLE

```
SELECT * FROM MTL_ONHAND_QUANTITIES;
```

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

Importing Blanket Purchase Agreements(BPA)

Importing Blanket Purchase Agreements(BPA) :

In this article we will see what a Blanket Purchase Agreement is and how we can import them along with the price breaks.

Overview of Blanket Purchase Agreements:

You create blanket purchase agreements when you know the detail of the goods or services you plan to buy from a specific supplier in a period, but you do not yet know the detail of your delivery schedules. You can use blanket purchase agreements to specify negotiated prices for your items before actually purchasing them.

Blanket Releases:

You can issue a blanket release against a blanket purchase agreement to place the actual order (as long as the release is within the blanket agreement effectivity dates. If your purchase agreement has price breaks, the quantity entered on the release determines what break price is defaulted into the Price field.

Import Process:

The Purchasing Document Open Interface concurrent program was replaced by two new concurrent Programs – Import Price Catalogs and Import Standard Purchase Orders.

Import Price Catalogs concurrent program is used to import Catalog Quotations, Standard Quotations, and Blanket Purchase Agreements.

Import Standard Purchase Orders concurrent program is used to import Unapproved or Approved Standard Purchase Orders.

You need to populate PO_HEADERS_INTERFACE and PO_LINES_INTERFACE to import header and line information into Purchasing. PO_LINES_INTERFACE table contains both line and shipment information, and imports data into both the PO_LINES and PO_LINE_LOCATIONS. The below are the additional columns that are required in PO_LINES_INTERFACE if you want to import price break information:

LINE_NUM
SHIPMENT_NUM
QUANTITY
UNIT_PRICE

If you are importing price break information through catalog quotations, you can also, optionally, populate the following columns in the PO_LINES_INTERFACE table:

MIN_ORDER_QUANTITY
MAX_ORDER_QUANTITY

Let's take an example to better understand. Suppose you want to create a blanket with one line and two price breaks and the details for the price break are as below:

1)quantity = 500, price = 10, effective date from '01-JAN-2006′ to '31-JUN-2006′
2)quantity = 500, price = 11, effective date from '01-JUL-2006′ to '01-JAN-2007′

To create the above the BPA, you would create ONE record in PO_HEADERS_INTERFACE and THREE records in PO_LINES_INTERFACE

LINE1: It will have only the line information. LINE NUM would be 1.
LINE2: For the first Price Break details but the LINE NUM will be the same as above i.e 1. SHIPMENT_NUM would be 1 and SHIPMENT_TYPE would be 'PRICE BREAK'
LINE3: For the second Price Break details but the LINE NUM will be the same as above i.e 1. SHIPMENT_NUM would be 2 and SHIPMENT_TYPE would be 'PRICE BREAK'
All the line-level records above must have the same INTERFACE_HEADER_ID.

–Inserting Header Information

insert into po_headers_interface

```
(interface_header_id,
action,
org_id,
document_type_code,
vendor_id,
vendor_site_id,
effective_date,
expiration_date,
Vendor_doc_num)
values
(po_headers_interface_s.nextval,
'ORIGINAL',
204,
'BLANKET',
21,
41,
'01-JAN-2006',
'01-JAN-2007',
'VENDOR04302006');

–Inserting Line Information

insert into po_lines_interface
(interface_line_id,
interface_header_id,
action,
item,
line_num,
unit_price,
unit_of_measure,
effective_date,
expiration_date,
ship_to_organization_id,
ship_to_location_id,
PRICE_BREAK_LOOKUP_CODE)
values
(po_lines_interface_s.nextval,
po_headers_interface_s.currval,
'ORIGINAL',
'AS54888',
1,
20,
'Each',
'01-JAN-2006',
'01-JAN-2007',
207,
207,
'NON CUMULATIVE');
```

Note: Cumulative: Price breaks apply to the cumulative quantity on all release shipments for the item.

Non–cumulative: Price breaks apply to quantities on individual release shipments for the item.

–Inserting First Price Break

```
insert into po_lines_interface
(interface_line_id,
interface_header_id,
action,
item,
line_num,
shipment_num,
shipment_type,
quantity,
unit_price,
unit_of_measure,
ship_to_organization_id,
ship_to_location_id,
effective_date,
expiration_date)
values
(po_lines_interface_s.nextval,
po_headers_interface_s.currval,
'ORIGINAL',
'AS54888',
1,
1,
'PRICE BREAK',
500,
10,
'Each',
207,
207,
'01-JAN-2006',
'30-JUN-2006');
```

–Inserting Second Price Break

```
insert into po_lines_interface
(interface_line_id,
interface_header_id,
action,
item,
line_num,
shipment_num,
shipment_type,
quantity,
unit_price,
unit_of_measure,
ship_to_organization_id,
ship_to_location_id,
effective_date,
```

```
expiration_date)
values
(po_lines_interface_s.nextval,
po_headers_interface_s.currval,
'ORIGINAL',
'AS54888',
1,
2,
'PRICE BREAK',
500,
11,
'Each',
207,
207,
'01-JUL-2006',
'01-JAN-2007');
```

Final Step:
Run Import Price Catalog Concurrent Program to create this Blanket Purchase Agreement.

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012 usage of $FLEX$

This article illustrates the usage of $FLEX$ with an example.
$FLEX$ is a special bind variable that can be used to base a parameter value on the other parameters (dependent parameters)

Syntax – :$FLEX$.Value_ Set_Name

Value_Set_Name is the name of value set for a prior parameter in the same parameter window that you

want your parameter to depend on.

Some scenarios where $FLEX$ can be used:

Example1:

Say you have a concurrent program with the below 2 parameters which are valuesets :

Parameter1 is Deparment

Parameter2 is Employee name

Let's say there are 100 deparments and each deparment has 200 employees. Therefore we have 2000 employees altogether.

If we display all department names in the valueset of parameter1 and all employee names in parameter2 value set then it might kill lot of performance and also it will be hard for a user to select an employee from the list of 2000 entries.

Better Solution is to let user select the department from the Department Valuset first. Based on the department selected, you can display only the employees in parameter2 that belong to the selected department in parameter1 valueset.

Example2:

Say you have a concurrent program with the below 2 parameters:

parameter1: directory path

parameter2: filename

Parameter1 and parameter2 are dependent on each other. If the user doesn't enter directory path, there is no point in enabling the parameter2 i.e filename. In such a case, parameter should be disabled.This can be achieved using $FLEX$.

Working Example of how to use $FLEX$:

Let's take the standard concurrent program "AP Withholding Tax Extract" to explain how to use $FLEX$.

This program has 7 parameters like "Date From", "Date To", "Supplier From", "Supplier To" etc

The requirement is to add an additional parameter called "File Name" where the user will give a name to the flat file where the tax extract will be written to, as a parameter. Instead of typing in the name of the file everytime you run the program, the file name should be defaulted with the value that the user provides for the parameter "Date From" plus ".csv" which is the file extension. Let us now see how this can be achieved using $FLEX$.

Navigation:

Application Developer responsibility > Concurrent > Program

Query up the Concurrent

Click "Parameters" Button

Add the parameter "File

- Seq: 80 (Something that is not already assigned to other parameters. It's always better to enter sequences in multiple of 5 or 10. So that you can insert any additional parameters if you want later in middle)
- Parameter: 'File Name'
- Description: 'File Name'
- Value set: '240 Characters'
- Prompt: File Name
- **Default Type: SQL Statement**
- **Default Value: Select :$FLEX$.FND_STANDARD_DATE||'.csv' from dual**

Here FND_STANDARD_DATE is the value set name of the parameter "Date From" as seen in the above screenshot.

$FLEX$.FND_STANDARD_DATE gets the value that the user enters for the parameter "Date From"

"select :$FLEX$.FND_STANDARD_DATE||'.csv' from dual" returns "Date From" parameter value appended with '.csv'

Save your work.

Now go to the respective responsibility and run the concurrent program.

When you enter the value of "Date From" and hit tab, File Name parameter will automatically be populated as shown in the below screenshot.

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012 Email the output of a concurrent program as Attachment

Email the output of a concurrent program as Attachment :

This article illustrates the steps to be followed to Email a concurrent program's output.

1. Write a procedure that will submit the concurrent program whose output has to be sent as an Email and once the program completes, send the output as Email using UTL_MAIL.send_attach_varchar2.
2. Register this procedure as a concurrent program so that this program can be run from Oracle Applications which will email a concurrent program's output.

Detailed explanation with sample code:

1. Write the below procedure which submits the desired concurrent program and waits until it completes and then sends the output of that program to the specified Email address using the utility **UTL_MAIL.send_attach_varchar2**

```
CREATE OR REPLACE PROCEDURE apps.erp_send_email

        (
        errbuf VARCHAR2,
        retode NUMBER,
        p_concurrent_program_name VARCHAR2,
        p_parameter1 NUMBER
        )
        IS
/*Variable declaration*/

        fhandle UTL_FILE.file_type;
        vtextout VARCHAR2 (32000);
        text VARCHAR2 (32000);
        v_request_id NUMBER := NULL;
        v_request_status BOOLEAN;
        v_phase VARCHAR2 (2000);
        v_wait_status VARCHAR2 (2000);
        v_dev_phase VARCHAR2 (2000);
        v_dev_status VARCHAR2 (2000);
        v_message VARCHAR2 (2000);
        v_application_id NUMBER;
        v_concurrent_program_id NUMBER;
        v_conc_prog_short_name VARCHAR2 (100);
        v_conc_prog_appl_short_name VARCHAR2 (100);
        v_output_file_path VARCHAR2 (200);

        BEGIN
        fnd_file.put_line (fnd_file.output,
        '_____
                                );
        fnd_file.put_line (fnd_file.output,
        'Conc Prog: ' || p_concurrent_program_name
```

```
        );
        fnd_file.put_line (fnd_file.output, 'Parameter 1:' ||
                                p_parameter1
                        );


                /* Get Concurrent_program_id of the desired program
                and application_id */
BEGIN

SELECT concurrent_program_id, application_id

        INTO v_concurrent_program_id, v_application_id
        FROM fnd_concurrent_programs_tl
        WHERE user_concurrent_program_name =
        p_concurrent_program_name;

        fnd_file.put_line (fnd_file.LOG,'Conc Prog ID:' ||
                                        v_concurrent_program_id
        );
        fnd_file.put_line (fnd_file.LOG, 'Application ID: ' ||
        v_application_id
                                        );

        /* Get the program's Short name */
        SELECT concurrent_program_name
        INTO v_conc_prog_short_name
        FROM fnd_concurrent_programs
        WHERE concurrent_program_id = v_concurrent_program_id;

        fnd_file.put_line (fnd_file.LOG,'Conc Prog Short Name: '
                                || v_conc_prog_short_name
        );

        /* Get the Application Short name */
        SELECT application_short_name
        INTO v_conc_prog_appl_short_name
        FROM fnd_application
        WHERE application_id = v_application_id;

        fnd_file.put_line (fnd_file.LOG,'Application Short Name:'
        || v_conc_prog_appl_short_name
        );
        EXCEPTION
        WHEN OTHERS
        THEN
        fnd_file.put_line (fnd_file.LOG, 'Error: ' ||
        SQLERRM);
        END;
        /* Calling fnd_request.submit_request to submit the desired
        the concurrent program*/
```

```
v_request_id:=
fnd_request.submit_request(v_conc_prog_appl_short_name,
v_conc_prog_short_name,
NULL, –Description
NULL, –Time to start the program
FALSE, – sub program
p_parameter1
);
fnd_file.put_line (fnd_file.LOG,'Concurrent Request Submitted
                    Successfully: ' || v_request_id
);
COMMIT;

IF v_request_id IS NOT NULL
THEN
/*Calling fnd_concurrent.wait_for_request to wait for the
          program to complete */
v_request_status:=
fnd_concurrent.wait_for_request
          (
          request_id => v_request_id,
INTERVAL => 10,
max_wait => 0,
phase => v_phase,
status => v_wait_status,
dev_phase => v_dev_phase,
dev_status => v_dev_status,
MESSAGE => v_message
);
v_dev_phase := NULL;
v_dev_status := NULL;
END IF;

/* Getting the path where output file of the program is
created */
SELECT outfile_name
INTO v_output_file_path
FROM fnd_concurrent_requests
WHERE request_id = v_request_id;

    /* Open the output file in Read mode */
    fhandle := UTL_FILE.fopen
                    ('/opt/oracle/OACRP1/common/admin/out/OACRP1_dtuusebs14','o'
                    || v_request_id || '.out', 'r');

IF UTL_FILE.is_open (fhandle)
THEN

DBMS_OUTPUT.put_line ('File read open');
ELSE
```

```
DBMS_OUTPUT.put_line ('File read not open');
END IF;

/* Get the contents of the file into variable "text"*/
LOOP
BEGIN
UTL_FILE.get_line (fhandle, vtextout);
text := text || vtextout || UTL_TCP.crlf;
EXCEPTION
WHEN NO_DATA_FOUND
THEN
EXIT;
END;
END LOOP;

UTL_FILE.fclose (fhandle);

/*Calling UTL_MAIL.send_attach_varchar2 to send the output as
Email attachment */
UTL_MAIL.send_attach_varchar2
(
sender => 'dtuebs@ventana.roche.com',
recipients => 'prudhvi.avuthu@contractors.roche.com',
subject => 'Testmail',
MESSAGE => 'Hello',
attachment => text,
att_inline => FALSE
);
END;
/
```

1. Register the above written procedure as a concurrent program

- Define Executable:

- Define Concurrent program with 2 parameters: Concurrent Program Name and Program short Name.

- Assign this concurrent program to the desired responsibility.
  For a more detailed explanation on how to register a concurrent program refer to the below article:
  http://www.erpschools.com/Apps/oracle-applications/articles/Sysadmin-and-AOL/Concurrent-Program-Registration-and-add-it-to-request-group/index.aspx
  [http://www.erpschools.com/Apps/oracle-applications/articles/Sysadmin-and-AOL/Concurrent-Program-Registration-and-add-it-to-request-group/index.aspx]

When this registered concurrent program is run, this program in turn submits the desired concurrent program and emails its output as an attachment to the required.

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

Interfaces and Conversions 2

Overview:

Oracle provides flexible and flexible tools in the form of Interface programs to import the master and transactional data like Customers, Invoices, and Sales Orders etc from external systems into Oracle Applications.

Conversion/Interface Strategy:

1. Data Mapping
   During the data mapping process, list of all the data sets and data elements that will need to be moved into the Oracle tables as part of conversion are identified. Data mapping tables are prepared as part of this activity that show what are the data elements that are needed by the target system to meet the business requirements and from where they will be extracted in the old system.
2. Download Programs
   After the conversion data mapping is complete, download programs are developed that are used to extract the identified conversion data elements from the current systems in the form of an ASCII flat file. The structure of the flat file must match the structure of the Oracle standard interface tables. These flat files generated may be in text form or a comma or space delimited, variable or fixed format data file.
3. Upload Program
   Once the data has been extracted to a flat file, it is then moved to the target file system and the data from the file is loaded into user defined staging tables in the target database using SQL Loader or UTL_FILE utilities. Then programs are written and run which validate the data in the staging tables and insert the same into the Oracle provided standard Interface tables.
4. Interface Program
   Once the interface tables are populated, the respective interface program (each data element interface has a specific interface program to run) is submitted. The interface programs validate the data, derive and assign the default values and ultimately populate the production base tables.

Interface/Conversion examples and details:
The below list of interfaces/conversions are covered in this section. Details like pre-requisites required, interface tables, interface program, base tables, validations that need

to be performed after inserting the details into the interface tables and required columns that need to be populated in the interface table are discussed for each interface.

- Order Import Interface (Sales Order Conversion)
- **Item import (Item conversion)**
- **Inventory On-hand quantity Interface**
- **Customer conversion**
- Auto Invoice Interface
- AR Receipts
- Lockbox Interface
- AP Invoices
- Vendor
- Purchase Orders
- Requisition
- Receiving
- Journal import
- Budget import
- Daily Conversion Rates

- **Order Import Interface (Sales Order Conversion)**
  Order Import enables you to import Sales Orders into Oracle Applications instead of manually entering them.
  Pre-requisites:

Order Type
Line Type
Items
Customers
Ship Method/ Freight Carrier
Sales Person
Sales Territories
Customer Order Holds
Sub Inventory/ Locations
On hand Quantity

Interface tables:

OE_HEADERS_IFACE_ALL
OE_LINES_IFACE_ALL
OE_ACTIONS_IFACE_ALL

OE_ORDER_CUST_IFACE_ALL
OE_PRICE_ADJS_IFACE_ALL
OE_PRICE_ATTS_IFACE_ALL

Base tables:
OE_ORDER_HEADERS_ALL
OE_ORDER_LINES_ALL
Pricing tables: QP_PRICING_ATTRIBUTES

Concurrent Program:
Order Import
Validations:
Check for sold_to_org_id. If does not exist, create new customer by calling create_new_cust_info API.
Check for sales_rep_id. Should exist for a booked order.
Ordered_date should exist (header level)
Delivery_lead_time should exist (line level)
Earliest_acceptable_date should exist.
Freight_terms should exist

Notes:
During import of orders, shipping tables are not populated.
If importing customers together with the order, OE_ORDER_CUST_IFACE_ALL has to be populated and the base tables are HZ_PARTIES, HZ_LOCATIONS.
Orders can be categorized based on their status:
1. Entered orders
2. Booked orders
3. Closed orders
Order Import API OE_ORDER_PUB.GET_ORDER and PROCESS_ORDER can also be used to import orders.

Some important columns that need to populated in the interface tables:
OE_HEADERS_IFACE_ALL:
ORIG_SYS_DOCUMENT_REF
ORDER_SOURCE
CONVERSION_RATE
ORG_ID
ORDER_TYPE_ID
PRICE_LIST
SOLD_FROM_ORG_ID
SOLD_TO_ORG_ID
SHIP_TO_ORG_ID
SHIP_FROM_ORG_ID
CUSTOMER_NAME
INVOICE_TO_ORG_ID
OPERATION_CODE

OE_LINES_IFACE_ALL
ORDER_SOURCE_ID
ORIG_SYS_DOCUMENT_REF

ORIG_SYS_LINE_REF
ORIG_SYS_SHIPMENT_REF
INVENTORY_ITEM_ID
LINK_TO_LINE_REF
REQUEST_DATE
DELIVERY_LEAD_TIME
DELIVERY_ID
ORDERED_QUANTITY
ORDER_QUANTITY_UOM
SHIPPING_QUANTITY
PRICING_QUANTITY
PRICING_QUANTITY_UOM
SOLD_FROM_ORG_ID
SOLD_TO_ORG_ID
INVOICE_TO_ ORG_ID
SHIP_TO_ORG_ID
PRICE_LIST_ID
PAYMENT_TERM_ID

• Item import (Item conversion)

The Item Interface lets you import items into Oracle Inventory.

Pre-requisites:

Creating an Organization
Code Combinations
Templates
Defining Item Status Codes
Defining Item Types

Interface tables:

MTL_SYSTEM_ITEMS_INTERFACE
MTL_ITEM_REVISIONS_INTERFACE (If importing revisions)
MTL_ITEM_CATEGORIES_INTERFACE (If importing categories)
MTL_INTERFACE_ERRORS (View errors after import)

Concurrent Program:

Item import
In the item import parameters form, for the parameter 'set process id', specify
the 'set process id' value given in the mtl_item_categories_interface table. The
parameter 'Create or Update' can have any value. Through the import process, we
can only create item category assignment(s). Updating or Deletion of item category assignment is not supported.

Base tables:

MTL_SYSTEM_ITEMS_B
MTL_ITEM_REVISIONS_B
MTL_CATEGORIES_B
MTL_CATEGORY_SETS_B
MTL_ITEM_STATUS
MTL_ITEM_TEMPLATES

Validations:

Check for valid item type.
Check for valid part_id/segment of the source table.
Validate part_id/segment1 for master org.
Validate and translate template id of the source table.
Check for valid template id. (Attributes are already set for items, default attributes for
that template, i.e., purchasable, stockable, etc)
Check for valid item status.
Validate primary uom of the source table.
Validate attribute values.
Validate other UOMs of the source table.
Check for unique item type. Discard the item, if part has non-unique item type.
Check for description, inv_um uniqueness
Validate organization id.
Load master records and category records only if all validations are passed.
Load child record if no error found.

Some important columns that need to populated in the interface tables:

MTL_SYSTEM_ITEMS_INTERFACE:

PROCESS_FLAG = 1 (1= Pending, 2= Assign Complete,
3= Assign/Validation Failed, 4= Validation succeeded; Import failed, 5 = Import in Process,
7 = Import succeeded)
TRANSACTION_TYPE = 'CREATE', 'UPDATE'
SET_PROCESS_ID = 1
ORGANIZATION_ID
DESCRIPTION
ITEM_NUMBER and/or SEGMENT (n)
MATERIAL_COST
REVISION
TEMPLATE_ID
SUMMARY_FLAG
ENABLED_FLAG
PURCHASING_ITEM_FLAG
SALES_ACCOUNT (defaulted from
MTL_PARAMETERS.SALES_ACCOUNT)
COST_OF_SALES_ACCOUNT (defaulted from MTL_PARAMETERS.
COST_OF_SALES_ACCOUNT)

MTL_ITEM_CATEGORIES_INTERFACE:

INVENTORY_ITEM_ID or ITEM_NUMBER.
ORGANIZATION_ID or ORGANIZATION_CODE or both.
TRANSACTION_TYPE = 'CREATE' ('UPDATE' or 'DELETE' is not possible through Item Import).
CATEGORY_SET_ID or CATEGORY_SET_NAME or both.
CATEGORY_ID or CATEGORY_NAME or both.
PROCESS_FLAG = 1
SET_PROCESS_ID (The item and category interface records should have the same set_process_id, if you are importing item and category assignment together)

MTL_ITEM_REVISIONS_INTERFACE:

INVENTORY_ITEM_ID or ITEM_NUMBER (Must match the item_number in mtl_system_items_interface table)
ORGANIZATION_ID or ORGANIZATION_CODE or both
REVISION
CHANGE_NOTICE
ECN_INITIATION_DATE
IMPLEMENTATION_DATE
IMPLEMENTED_SERIAL_NUMBER
EFFECTIVITY_DATE
ATTRIBUTE_CATEGORY
ATTRIBUTEn
REVISED_ITEM_SEQUENCE_ID
DESCRIPTION
PROCESS_FLAG = 1
TRANSACTION_TYPE = 'CREATE'
SET_PROCESS_ID = 1

Each row in the mtl_item_revisions_interface table must have the REVISION
and EFFECTIVITY_DATE in alphabetical (ASCII sort) and chronological order.

• Inventory On-hand quantity Interface

This interface lets you import the on hand inventory into Oracle.

Interface tables:

MTL_TRANSACTIONS_INTERFACE
MTL_MTL_TRANSACTION_LOTS_INTERFACE (If the item is Lot controlled)
MTL_SERIAL_NUMBERS_INTERFACE (If the item is Serial controlled)

Concurrent Program:

Launch the **Transaction Manager** through Interface Manager or explicitly

call the API – INV_TXN_MANAGER_PUB.PROCESS_TRANSACTIONS () to launch a dedicated transaction worker to process them.
The Transaction Manager picks up the rows to process based on the LOCK_FLAG, TRANSACTION_MODE, and PROCESS_FLAG. Only records with TRANSACTION_MODE of 3, LOCK_FLAG of '2', and PROCESS_FLAG of '1' will be picked up by the Transaction Manager and assigned to a Transaction Worker. If a record fails to process completely, then PROCESS_FLAG will be set to '3' and ERROR_CODE and ERROR_EXPLANATION will be populated with the cause for the error.

Base Tables:

MTL_ON_HAND_QUANTITIES
MTL_LOT_NUMBERS
MTL_SERIAL_NUMBERS

Validations:

Validate organization_id
Check if item is assigned to organization
Validate disposition_id
Check if the item for the org is lot controlled before inserting into the Lots interface table.
Check if the item for the org is serial controlled before inserting into Serial interface table.
Check if inventory already exists for that item in that org and for a lot.
Validate organization_id, organization_code.
Validate inventory item id.
Transaction period must be open.

Some important columns that need to be populated in the interface tables:

MTL_TRANSACTIONS_INTERFACE:
TRANSACTION_SOURCE_NAME (ANY USER DEFINED VALUE),
TRANSACTION_HEADER_ID
(MTL_MATERIAL_TRANSACTIONS_S.NEXTVAL)
TRANSACTION_INTERFACE_ID
(MTL_MATERIAL_TRANSACTIONS_S.NEXTVAL – If item is lot or serial controlled, use this field to link to mtl_transactions_interface otherwise leave it as NULL),
TRANSACTION_DATE,
TRANSACTION_TYPE_ID,
PROCESS_FLAG (1 = Yet to be processed, 2 = Processed, 3= Error)
TRANSACTION_MODE (2 = Concurrent – to launch a dedicated transaction worker
to explicitly process a set of transactions.
3 = Background – will be picked up by transaction manager
polling process and assigned to transaction worker. These will not be picked up until the transaction manager is running)
SOURCE_CODE,

SOURCE_HEADER_ID,
SOURCE_LINE_ID (Details about the source like Order Entry etc for tracking purposes)
TRANSACTION_SOURCE_ID

Source Type	Foreign Key Reference
Account	GL_CODE_COMBINATIONS.CODE_COMBINATION_ID
Account Alias	MTL_GENERIC_DISPOSITIONS.DISPOSITION_ID
Job or schedule	WIP_ENTITIES.WIP_ENTITY_ID
Sales Order	MTL_SALES_ORDERS.SALES_ORDER_ID

ITEM_SEGMENT1 TO 20,
TRANSACTION_QTY,
TRANSACTION_UOM,
SUBINVENTORY_CODE,
ORGANIZATION_ID,
LOC_SEGMENT1 TO 20.

MTL_TRANSACTION_LOTS_INTERFACE:
TRANSACTION_INTERFACE_ID,
LOT_NUMBER,
LOT_EXPIRATION_DATE,
TRANSACTION_QUANTITY,
SERIAL_TRANSACTION_TEMP_ID (This is required for items under both lot and serial control to identify child records in mtl_serial_numbers_interface)

MTL_SERIAL_NUMBERS_INTERFACE:
TRANSACTION_INTERFACE_ID,
FM_SERIAL_NUMBER,
TO_SERIAL_NUMBER,
VENDOR_SERIAL_NUMBER

Customer conversion

Customer Interface helps you create customers in Oracle Applications.

Interface tables:

RA_CUSTOMERS_INTERFACE_ALL
RA_CUSTOMER_PROFILES_INT_ALL
RA_CONTACT_PHONES_INT_ALL
RA_CUSTOMER_BANKS_INT_ALL
RA_CUST_PAY_METHOD_INT_ALL

Base tables:

RA_CUSTOMERS
RA_ADDRESSES_ALL
RA_CUSTOMER_RELATIONSHIPS_ALL
RA_SITE_USES_ALL

Concurrent program:

Customer Interface

Validations:

Check if legacy values fetched are valid.
Check if customer address site is already created.
Check if customer site use is already created.
Check is customer header is already created.
Check whether the ship_to_site has associated bill_to_site
Check whether associated bill_to_site is created or not.
Profile amounts validation:

Validate cust_account_id, validate customer status.

Check if the location already exists in HZ_LOCATIONS. If does not exist, create new location.

Some important columns that need to be populated in the interface tables:

RA_CUSTOMERS_INTERFACE_ALL:

ORIG_SYSTEM_CUSTOMER_REF
SITE_USE_CODE
ORIG_SYSTEM_ADDRESS_REF
INSERT_UPDATE_FLAG (I = Insert, U = Update)
CUSTOMER_NAME
CUSTOMER_NUMBER
CUSTOMER_STATUS
PRIMARY_SITE_USE_FLAG
LOCATION
ADDRESS1
ADDRESS2
ADDRESS3
ADDRESS4
CITY
STATE
PROVINCE
COUNTY
POSTAL_CODE
COUNTRY
CUSTOMER_ATTRIBUTE1
CUSTOMER_ATTRIBUTE2
CUSTOMER_ATTRIBUTE3
CUSTOMER_ATTRIBUTE4
CUSTOMER_ATTRIBUTE5
LAST_UPDATED_BY
LAST_UPDATE_DATE
CREATED_BY
CREATION_DATE
ORG_ID

CUSTOMER_NAME_PHONETIC

RA_CUSTOMER_PROFILES_INT_ALL:

INSERT_UPDATE_FLAG
ORIG_SYSTEM_CUSTOMER_REF
ORIG_SYSTEM_ADDRESS_REF
CUSTOMER_PROFILE_CLASS_NAME
CREDIT_HOLD
LAST_UPDATED_BY
LAST_UPDATE_DATE
CREATION_DATE
CREATED_BY
ORG_ID

RA_CONTACT_PHONES_INT_ALL:

ORIG_SYSTEM_CONTACT_REF
ORIG_SYSTEM_TELEPHONE_REF
ORIG_SYSTEM_CUSTOMER_REF
ORIG_SYSTEM_ADDRESS_REF
INSERT_UPDATE_FLAG
CONTACT_FIRST_NAME
CONTACT_LAST_NAME
CONTACT_TITLE
CONTACT_JOB_TITLE
TELEPHONE
TELEPHONE_EXTENSION
TELEPHONE_TYPE
TELEPHONE_AREA_CODE
LAST_UPDATE_DATE
LAST_UPDATED_BY
LAST_UPDATE_LOGIN
CREATION_DATE
CREATED_BY
EMAIL_ADDRESS
ORG_ID

- **Customer API**

Trading Community Architecture (TCA) is an architecture concept designed to support complex

trading communities. These APIs utilize the new TCA model, inserting directly to the HZ tables.

API Details:

1. Set the organization id
   Exec dbms_application_info.set_client_info('204');
2. Create a party and an account
   HZ_CUST_ACCOUNT_V2PUB.CREATE_CUST_ACCOUNT()
   HZ_CUST_ACCOUNT_V2PUB.CUST_ACCOUNT_REC_TYPE
   HZ_PARTY_V2PUB.ORGANIZATION_REC_TYPE
   HZ_CUSTOMER_PROFILE_V2PUB.CUSTOMER_PROFILE_REC_TYPE

3. Create a physical location
   HZ_LOCATION_V2PUB.CREATE_LOCATION()
   HZ_LOCATION_V2PUB.LOCATION_REC_TYPE

4. Create a party site using party_id you get from step 2 and location_id from step 3.
   HZ_PARTY_SITE_V2PUB.CREATE_PARTY_SITE()
   HZ_PARTY_SITE_V2PUB.PARTY_SITE_REC_TYPE

5. Create an account site using account_id you get from step 2 and party_site_id from step 4.
   HZ_CUST_ACCOUNT_SITE_V2PUB.CREATE_CUST_ACCT_SITE()
   HZ_CUST_ACCOUNT_SITE_V2PUB.CUST_ACCT_SITE_REC_TYPE

6. Create an account site use using cust_acct_site_id you get from step 5 ans site_use_code = 'BILL_TO'.
   HZ_CUST_ACCOUNT_SITE_V2PUB.CREATE_CUST_SITE_USE()
   HZ_CUST_ACCOUNT_SITE_V2PUB.CUST_SITE_USE_REC_TYPE
   HZ_CUSTOMER_PROFILE_V2PUB.CUSTOMER_PROFILE_REC_TYPE

Base table:

HZ_PARTIES
HZ_PARTY_SITES
HZ_LOCATIONS
HZ_CUST_ACCOUNTS
HZ_CUST_SITE_USES_ALL
HZ_CUST_ACCT_SITES_ALL
HZ_PARTY_SITE_USES

Validations:

Check if legacy values fetched are valid.
Check if customer address site is already created.
Check if customer site use is already created.
Check is customer header is already created.
Check whether the ship_to_site has associated bill_to_site
Check whether associated bill_to_site is created or not.
Profile amounts validation:
Validate cust_account_id, validate customer status.
Check if the location already exists in HZ_LOCATIONS. If does not exist, create new location.

For detailed explanation refer to the below article:
http://www.erpschools.com/Apps/oracle-applications/articles/financials/Receivables/Customer-TCA-Architecture-and-API/index.aspx [http://www.erpschools.com/Apps/oracle-applications/articles/financials/Receivables/Customer-TCA-Architecture-and-API/index.aspx]

Auto Invoice interface

This interface is used to import Customer invoices, Credit memos, Debit memos and On Account credits.

Pre-requisites:

Set of Books
Code combinations
Items
Sales representatives
Customers
Sales Tax rate
Payment Terms
Transaction Types
Freight Carriers
FOB
Batch Sources
Accounting Rules

Interface tables:

RA_INTERFACE_LINES_ALL
RA_INTERFACE_SALESCREDITS
RA_INTERFACE_DISTRIBUTIONS
RA_INTERFACE_ERRORS (details about the failed records)

Base tables:

RA_BATCHES
RA_CUSTOMER_TRX_ALL
RA_CUSTOMER_TRX_LINES_ALL
AR_PAYMENT_SCHEDULES_ALL RA_CUSTOMER_TRX_LINE_SALESREPS
RA_CUST_TRX_GL_DIST_ALL
RA_CUSTOMER_TRX_TYPES_ALL

Concurrent Program:

Auto invoice master program

Validations:

Check for amount, batch source name, conversion rate, conversion type.
Validate orig_system_bill_customer_id, orig_system_bill_address_id, quantity.
Validate if the amount includes tax flag.

Some important columns that need to be populated in the interface tables:

RA_INTERFACE_LINES_ALL:

AGREEMENT_ID
COMMENTS
CONVERSION_DATE
CONVERSION_RATE
CONVERSION_TYPE
CREDIT_METHOD_FOR_ACCT_RULE
CREDIT_METHOD_FOR_INSTALLMENTS
CURRENCY_CODE

CUSTOMER_BANK_ACCOUNT_ID
CUST_TRX_TYPE_ID
DOCUMENT_NUMBER
DOCUMENT_NUMBER_SEQUENCE_ID
GL_DATE
HEADER_ATTRIBUTE1–15
HEADER_ATTRIBUTE_CATEGORY
INITIAL_CUSTOMER_TRX_ID
INTERNAL_NOTES
INVOICING_RULE_ID
ORIG_SYSTEM_BILL_ADDRESS_ID
ORIG_SYSTEM_BILL_CONTACT_ID
ORIG_SYSTEM_BILL_CUSTOMER_ID
ORIG_SYSTEM_SHIP_ADDRESS_ID
ORIG_SYSTEM_SHIP_CONTACT_ID
ORIG_SYSTEM_SHIP_CUSTOMER_ID
ORIG_SYSTEM_SOLD_CUSTOMER_ID
ORIG_SYSTEM_BATCH_NAME
PAYMENT_SERVER_ORDER_ID
PREVIOUS_CUSTOMER_TRX_ID
PRIMARY_SALESREP_ID
PRINTING_OPTION
PURCHASE_ORDER
PURCHASE_ORDER_DATE
PURCHASE_ORDER_REVISION
REASON_CODE
RECEIPT_METHOD_ID
RELATED_CUSTOMER_TRX_ID
SET_OF_BOOKS_ID
TERM_ID
TERRITORY_ID
TRX_DATE
TRX_NUMBER

- # Receipt API

To bring in Unapplied Receipts and Conversion Receipts for Open Debit items to reduce the balance to the original amount due.
Pre-requisites:

Set of Books
Code combinations
Items
Quick Codes
Sales representatives
Customers
Sales Tax rate

API:

AR_RECEIPT_API_PUB.CREATE_CASH
AR_RECEIPT_API_PUB.CREATE_AND_APPLY

Base tables:

AR_CASH_RECEIPTS

Validations:

Check the currency and the exchange rate type to assign the exchange rate.
Validate bill to the customer.
Get bill to site use id.
Get the customer trx id for this particular transaction number.
Get payment schedule date for the customer trx id.

• Lockbox interface

AutoLockbox lets us automatically process receipts that are sent directly to the bank instead of manually feeding them in Oracle Receivables.

AutoLockbox is a three step process:

1. ***Import:*** During this step, Lockbox reads and formats the data from your bank file
into interface table AR_PAYMENTS_INTERFACE_ALL using a SQL *Loader
script.
2. ***Validation:*** The validation program checks data in this interface table for
compatibility with Receivables. Once validated, the data is transferred into QuickCash tables (AR_INTERIM_CASH_RECEIPTS_ALL and AR_INTERIM_CASH_RCPT_LINES_ALL).
3. ***Post QuickCash:*** This step applies the receipts and updates your customer's
balances.

Pre-Requisites:

Banks
Receipt Class
Payment Method
Receipt Source
Lockbox
Transmission format
AutoCash Rule sets

Interface tables:

AR_PAYMENTS_INTERFACE_ALL (Import data from bank file)

AR_INTERIM_CASH_RECEIPTS_ALL
AR_INTERIM_CASH_RCPT_LINES_ALL (Validate data in interface table and place in quick cash tables)

Base Tables:
AR_CASH_RECEIPTS
AR_RECEIVABLES_APPLICATIONS
AR_ADJUSTMENTS
AR_DISTRIBUTIONS_ALL
AR_PAYMENT_SCHEDULES_ALL

Concurrent program:
Lockbox

Validations:
Check for valid record type, transmission record id.
Validate sum of the payments within the transmission.
Identify the lockbox number (no given by a bank to identify a lockbox).

Some important columns that need to be populated in the interface tables:
AR_PAYMENTS_INTERFACE_ALL:
STATUS
RECORD_TYPE
LOCKBOX_NUMBER
BATCH_NAME
TRANSIT_ROUTING_NUMBER
ACCOUNT
CHECK_NUMBER
REMITTANCE_AMOUNT
DEPOSIT_DATE
ITEM_NUMBER
CURRENCY_CODE
DEPOSIT_TIME

- # AP invoice interface

This interface helps us to import vendor invoices into Oracle applications from external systems into Oracle Applications.
Pre-requisites:

Set of Books
Code combinations
Employees
Lookups

Interface tables:
AP_INVOICES_INTERFACE
AP_INVOICE_LINES_INTERFACE

Base tables:

AP_INVOICES_ALL – header information
AP_INVOICE_DISTRIBUTIONS_ALL – lines info

Concurrent program:

Payables Open Interface Import

Validations:

Check for valid vendor
Check for Source, Location, org_id, currency_code's validity
Check for valid vendor site code.
Check if record already exists in payables interface table.

Some important columns that need to be populated in the interface tables:

AP_INVOICES_INTERFACE:

INVOICE_ID
INVOICE_NUM
INVOICE_DATE
VENDOR_NUM
VENDOR_SITE_ID
INVOICE_AMOUNT
INVOICE_CURRENCY_CODE
EXCHANGE_RATE
EXCHANGE_RATE_TYPE
EXCHANGE_DATE
DESCRIPTION
SOURCE
PO_NUMBER
PAYMENT_METHOD_LOOKUP_CODE
PAY_GROUP_LOOKUP_CODE
ATTRIBUTE1 TO 15
ORG_ID

AP_INVOICE_LINES_INTERFACE:

INVOICE_ID
INVOICE_LINE_ID
LINE_TYPE_LOOKUP_CODE
AMOUNT
DESCRIPTION
TAX_CODE
PO_NUMBER
PO_LINE_NUMBER
PO_SHIPMENT_NUM
PO_DISTRIBUTION_NUM
PO_UNIT_OF_MEASURE
QUANTITY_INVOICED
DIST_CODE_CONCATENATED
DIST_CODE_COMBINATION_ID
ATTRIBUTE1
ATTRIBUTE2
ATTRIBUTE3

ATTRIBUTE4
ATTRIBUTE5
ORG_ID

Vendor conversion/interface

This interface is used to import suppliers, supplier sites and site contacts into Oracle applications.

Pre-requisites setup's required:

Payment terms
Pay Groups
CCID
Supplier classifications
Bank Accounts
Employees (if employees have to set up as vendors)

Interface tables:

AP_SUPPLIERS_INT
AP_SUPPLIER_SITES_INT
AP_SUP_SITE_CONTACT_INT

Base Tables:

PO_VENDORS
PO_VENDOR_SITES_ALL
PO_VENDOR_CONTACTS

Interface programs:

Supplier Open Interface Import
Supplier Sites Open Interface Import
Supplier Site Contacts Open Interface Import

Validations:

Check if vendor already exists
Check if vendor site already exists
Check if site contact already exists
Check if term is defined.

Some important columns that need to be populated in the interface tables:
AP_SUPPLIERS_INT:

VENDOR_NUMBER, VENDOR_NAME, VENDOR_TYPE, STATE_REPORTABLE, FED_REPORTABLE, NUM_1099, TYPE_1099, PAY_GROUP_LOOKUP_CODE, VENDOR_ID is auto

generated.

AP_SUPPLIER_SITES_INT:

VENDOR_SITE_ID, ORG_ID, VENDOR_SITE_CODE, INACTIVE_DATE, PAY_SITE, PURCHASING_SITE, SITE_PAYMENT_TERM, ADDRESS1, ADDRESS2.ADDRESS3, CITY, STATE, COUNTRY, ZIP, PH_NUM, FAX_NUMBER, TAX_REPORTING_SITE_FLAG.

AP_SUP_SITE_CONTACTS_INT:

VENDOR_ID, VENDOR_SITE_ID, FIRST_NAME, LAST_NAME, AREA_CODE, PHONE, EMAIL, ORG_ID

• Purchase Order conversion:

The Purchasing Document Open Interface concurrent program was replaced by two new concurrent programs Import Price Catalogs and Import Standard Purchase Orders. Import Price Catalogs concurrent program is used to import Catalog Quotations, Standard Quotations, and Blanket Purchase Agreements. Import Standard Purchase Orders concurrent program is used to import Unapproved or Approved Standard Purchase Orders.

Import Standard Purchase Orders

Pre-requisites:

Suppliers, sites and contacts
Buyers
Line Types
Items
PO Charge account setup

Interface Tables:

PO_HEADERS_INTERFACE
PO_LINES_INTERFACE
PO_DISTRIBUTIONS_INTERFACE
PO_INTERFACE_ERRORS (Fallouts)

Interface Program:

Import Standard Purchase Orders.

Base Tables:

PO_HEADERS_ALL
PO_LINES_ALL
PO_DISTRIBUTIONS_ALL
PO_LINE_LOCATIONS_ALL

Validations:

Header:

Check if OU name is valid
Check if Supplier is valid
Check if Supplier site is valid
Check if buyer is valid
Check if Payment term is valid
Check if Bill to and ship to are valid
Check if FOB, freight terms are valid

Lines:

Check if Line_type, ship_to_org, item, uom, ship_to_location_id, requestor, charge_account, deliver_to_location are valid

General:

Check for duplicate records in interface tables
Check if the record already exists in base tables.

Some important columns that need to be populated in the interface tables:

PO_HEADERS_INTERFACE:

INTERFACE_HEADER_ID (PO_HEADERS_INTERFACE_S.NEXTVAL), BATCH_ID, ORG_ID, INTERFACE_SOURCE_CODE, ACTION ('ORIGINAL','UPDATE','REPLACE'), GROUP_CODE, DOCUMENT_TYPE_CODE, PO_HEADER_ID (NULL), RELEASE_ID, RELEASE_NUM, CURRENCY_CODE, RATE, AGENT_NAME, VENDOR_ID, VENDOR_SITE_ID, SHIP_TO_LOCATION, BILL_TO_LOCATION, PAYMENT_TERMS

PO_LINES_INTERFACE:

INTERFACE_LINE_ID, INTERFACE_HEADER_ID, LINE_NUM, SHIPMENT_NUM, ITEM, REQUISITION_LINE_ID, UOM, UNIT_PRICE, FREIGHT_TERMS, FOB

PO_DISTRIBUTIONS_INTERFACE:

INTERFACE_LINE_ID, INTERFACE_HEADER_ID, INTERFACE_DISTRIBUTION_ID, DISTRIBUTION_NUM, QUANTITY_ORDERED, QTY_DELIVERED, QTY_BILLED, QTY_CANCELLED, DELIVER_TO_LOCATION_ID, DELIVER_TO_PERSON_ID, SET_OF_BOOKS, CHARGE_ACCT, AMOUNT_BILLED.

Import Blanket Purchase Agreements:

Interface Tables:

PO_HEADERS_INTERFACE
PO_LINES_INTERFACE

Interface program:

Import Price Catalogs

Base tables:

PO_HEADERS_ALL
PO_LINES_ALL
PO_LINE_LOCATIONS_ALL

Example:

Suppose you want to create a blanket with one line and two price breaks and the details for the price break are as below:
1) Quantity = 500, price = 10, effective date from '01-JAN-2006′ to '31-JUN-2006′
2) Quantity = 500, price = 11, effective date from '01-JUL-2006′ to '01-JAN-2007′
To create the above the BPA, you would create ONE record in PO_HEADERS_INTERFACE and THREE records in PO_LINES_INTERFACE
LINE1: It will have only the line information. LINE NUM would be 1.

LINE2: For the first Price Break details, LINE NUM will be the same as above i.e. 1. SHIPMENT_NUM would be 1 and SHIPMENT_TYPE would be 'PRICE BREAK'

LINE3: For the second Price Break details, LINE NUM will be the same as above i.e. 1. SHIPMENT_NUM would be 2 and SHIPMENT_TYPE would be 'PRICE BREAK'

All the line-level records above must have the same INTERFACE_HEADER_ID.
For detailed explanation refer to the below article:
http://www.erpschools.com/Apps/oracle-applications/articles/financials/Purchasing/Import-Blanket-Purchase-Agreements/index.aspx

- # Requisition import

You can automatically import requisitions into Oracle Applications using the Requisitions Open Interface
Pre-requisites:

Set of Books
Code combinations
Employees
Items
Define a Requisition Import Group-By method in the Default region of the Purchasing Options window.
Associate a customer with your deliver-to location using the Customer Addresses window for internally sourced requisitions.

Interface tables:

PO_REQUISITIONS_INTERFACE_ALL
PO_REQ_DIST_INTERFACE_ALL

Base tables:

PO_REQUISITIONS_HEADERS_ALL
PO_REQUISITION_LINES_ALL
PO_REQ_DISTRIBUTIONS_ALL

Concurrent program:

REQUISITION IMPORT

Validations:

Check for interface transaction source code, requisition destination type. Check for quantity ordered, authorization status type.

Some important columns that need to be populated in the interface tables:

PO_REQUISITIONS_INTERFACE_ALL:

INTERFACE_SOURCE_CODE (to identify the source of your imported Requisitions)
DESTINATION_TYPE_CODE
AUTHORIZATION_STATUS
PREPARER_ID or PREPARER_NAME
QUANTITY
CHARGE_ACCOUNT_ID or charge account segment values
DESTINATION_ORGANIZATION_ID or
DESTINATION_ORGANIZATION_
CODE
DELIVER_TO_LOCATION_ID or DELIVER_TO_LOCATION_CODE
DELIVER_TO_REQUESTOR_ID or
DELIVER_TO_REQUESTOR_NAME
ORG_ID
ITEM_ID or item segment values (values if the SOURCE_TYPE_CODE or DESTINATION_TYPE_CODE is 'INVENTORY')

PO_REQ_DIST_INTERFACE_ALL:

CHARGE_ACCOUNT_ID or charge account segment values
DISTRIBUTION_NUMBER
DESTINATION_ORGANIZATION_ID
DESTINATION_TYPE_CODE
INTERFACE_SOURCE_CODE
ORG_ID
DIST_SEQUENCE_ID (if MULTI_DISTRIBUTIONS is set to Y)

- **PO Receipts Interface**

The Receiving Open Interface is used for processing and validating receipt data that comes from sources other than the Receipts window in Purchasing.

Pre-requisites:
Set of Books
Code combinations
Employees
Items

Interface tables:
RCV_HEADERS_INTERFACE
RCV_TRANSACTIONS_INTERFACE
PO_INTERFACE_ERRORS

Concurrent program:
RECEIVING OPEN INTERFACE

Base tables:
RCV_SHIPMENT_HEADERS
RCV_SHIPMENT_LINES
RCV_TRANSACTIONS

Validations:
Check that SHIPPED_DATE should not be later than today.
Check if vendor is valid.
If Invoice number is passed, check for its validity
Check if Item is valid

Some important columns that need to be populated in the interface tables:

RCV_HEADERS_INTERFACE:
HEADER_INTERFACE_ID
GROUP_ID
PROCESSING_STATUS_
CODE
RECEIPT_SOURCE_CODE
TRANSACTION_TYPE
SHIPMENT_NUM
RECEIPT_NUM
VENDOR_NAME
SHIP_TO_
ORGANIZATION_CODE
SHIPPED_DATE
INVOICE_NUM
INVOICE_DATE
TOTAL_INVOICE_
AMOUNT
PAYMENT_TERMS_ID
EMPLOYEE_NAME
VALIDATION_FLAG (Indicates whether to validate a row or not, values 'Y',

'N')

RCV_TRANSACTIONS_INTERFACE:

INTERFACE_TRANSACTION_ID
GROUP_ID
TRANSACTION_TYPE ('SHIP' for a standard shipment (an ASN or ASBN)
or 'RECEIVE' for a standard receipt)
TRANSACTION_DATE
PROCESSING_STATUS_CODE ='PENDING'
CATEGORY_ID
QUANTITY
UNIT_OF_MEASURE
ITEM_DESCRIPTION
ITEM_REVISION
EMPLOYEE_ID
AUTO_TRANSACT_CODE
SHIP_TO_LOCATION_ID
RECEIPT_SOURCE_CODE
TO_ORGANIZATION_CODE
SOURCE_DOCUMENT_CODE
PO_HEADER_ID
PO_RELEASE_ID
PO_LINE_ID
PO_LINE_LOCATION_ID
PO_DISTRIBUTION_ID
SUBINVENTORY
HEADER_INTERFACE_ID
DELIVER_TO_PERSON_NAME
DELIVER_TO_LOCATION_CODE
VALIDATION_FLAG
ITEM_NUM
VENDOR_ITEM_NUM
VENDOR_ID
VENDOR_SITE_ID
ITEM_ID
ITEM_DESCRIPTION
SHIP_TO_LOCATION_ID

- # GL Journal interface

This interface lets you import journals from other applications like Receivables, Payables etc to integrate the information with General Ledger.
Pre-requisites:

Set of Books
Flex field Value sets
Code Combinations
Currencies

Categories
Journal Sources

Interface tables:

GL_INTERFACE

Base tables:

GL_JE_HEADERS
GL_JE_LINES
GL_JE_BACTHES

Concurrent Program:

Journal Import
Journal Posting — populates GL_BALANCES

Validations:

Validate SOB, journal source name, journal category name, actual flag
A – Actual amounts
B – Budget amounts
E – Encumbrance amount

If you enter E in the interface table, then enter appropriate encumbrance ID, if
B enter budget id.
Check if accounting date or GL date based period name is valid (i.e., not closed).
Check if accounting date falls in open or future open period status.
Check chart of accounts id based on Sob id.
Check if code combination is valid and enabled.
Check if record already exists in GL interface table.
Check if already journal exists in GL application.

Some important columns that need to be populated in the interface tables:

GL_INTERFACE:

STATUS
SET_OF_BOOKS_ID
ACCOUNTING_DATE
CURRENCY_CODE
DATE_CREATED
CREATED_BY
ACTUAL_FLAG
USER_JE_CATEGORY_NAME
USER_JE_SOURCE_NAME
CURRENCY_CONVERSION_DATE
ENCUMBRANCE_TYPE_ID
BUDGET_VERSION_ID
USER_CURRENCY_CONVERSION_TYPE
CURRENCY_CONVERSION_RATE
SEGMENT1 to
ENTERED_DR

ENTERED_CR
ACCOUNTED_DR
ACCOUNTED_CR
TRANSACTION_DATE
PERIOD_NAME
JE_LINE_NUM
CHART_OF_ACCOUNTS_ID
FUNCTIONAL_CURRENCY_CODE
CODE_COMBINATION_ID
DATE_CREATED_IN_GL
GROUP_ID

- # GL budget interface

Budget interface lets you load budget data from external sources into Oracle Applications.

Pre-requisites:

Set of Books
Flex field Value sets
Code Combinations

Interface tables:
GL_BUDGET_INTERFACE

Base tables:
GL_BUDGETS
GL_BUDGET_ASSIGNMENTS
GL_BUDGET_TYPES

Concurrent program:
Budget Upload

Validations:
Check if CURRENCY_CODE is valid.
Check if SET_OF_BOOKS_ID is valid.
Check if BUDGET_ENTITY_NAME (budget organization) is valid.

Some important columns that need to be populated in the interface tables:

GL_BUDGET_INTERFACE:

BUDGET_NAME NOT
BUDGET_ENTITY_NAME
CURRENCY_CODE
FISCAL_YEAR

UPDATE_LOGIC_TYPE
BUDGET_ENTITY_ID
SET_OF_BOOKS_ID
CODE_COMBINATION_ID
BUDGET_VERSION_ID
PERIOD_TYPE
DR_FLAG
STATUS
ACCOUNT_TYPE
PERIOD1_AMOUNT through PERIOD60_AMOUNT
SEGMENT1 through SEGMENT30

- **GL daily conversion rates**
  This interface lets you load the rates automatically into General Ledger.
  Pre-requisites:

  Currencies
  Conversion rate Types

Interface tables:

GL_DAILY_RATES_INTERFACE

Base tables:

GL_DAILY_RATES
GL_DAILY_CONVERSION_TYPES

Concurrent Program:

You do not need to run any import programs. The insert, update, or deletion of rates in GL_DAILY_RATES is done automatically by database triggers on the GL_DAILY_RATES_INTERFACE. All that is required is to develop program to populate the interface table with daily rates information.

Validations:

Check if
FROM_CURRENCY and TO_CURRENCY are valid.
Check if USER_CONVERSION_TYPE is valid.

Some important columns that need to be populated in the interface tables:

GL_DAILY_RATES_INTERFACE:

FROM_CURRENCY
TO_CURRENCY
FROM_CONVERSION_DATE
TO_CONVERSION_DATE
USER_CONVERSION_TYPE
CONVERSION_RATE
MODE_FLAG (D= Delete, I = Insert, U = Update)

INVERSE_CONVERSION_RATE

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

Order to Cash cycle (O2C)

In this article, we will go through the Order to Cash cycle. The below are the steps in short:

- Enter the Sales Order
- Book the Sales Order
- Launch Pick Release
- Ship Confirm
- Create Invoice
- Create the Receipts either manually or using Auto Lockbox (In this article we will concentrate on Manual creation)
- Transfer to General Ledger
- Journal Import
- Posting

Let's get into the details of each step mentioned above.

Enter the Sales Order:

Navigation:

Order Management Super User Operations (USA)>Orders Returns >Sales Orders

Enter the Customer details (Ship to and Bill to address), Order type.

Click on Lines Tab. Enter the Item to be ordered and the quantity required.

Line is scheduled automatically when the Line Item is saved.

Scheduling/unscheduling can be done manually by selecting Schedule/Un schedule from the Actions Menu.

You can check if the item to be ordered is available in the Inventory by clicking on Availability Button.

Save the work.

Underlying Tables affected:

In Oracle, Order information is maintained at the header and line level.

The header information is stored in OE_ORDER_HEADERS_ALL and the line information in OE_ORDER_LINES_ALL when the order is entered. The column called FLOW_STATUS_CODE is available in both the headers and lines tables which tell us the status of the order at each stage.

At this stage, the FLOW_STATUS_CODE in OE_ORDER_HEADERS_ALL is 'Entered'

Book the Sales Order:

Book the Order by clicking on the Book Order button.

Now that the Order is BOOKED, the status on the header is change accordingly.

Underlying tables affected:

At this stage:
The FLOW_STATUS_CODE in the table OE_ORDER_HEADERS_ALL would be 'BOOKED'
The FLOW_STATUS_CODE in OE_ORDER_LINES_ALL will be
'AWAITING_SHIPPING'.
Record(s) will be created in the table WSH_DELIVERY_DETAILS with
RELEASED_STATUS='R' (Ready to Release)
Also Record(s) will be inserted into WSH_DELIVERY_ASSIGNMENTS.
At the same time DEMAND INTERFACE PROGRAM runs in the background and inserts

into MTL_DEMAND

Launch Pick Release:

Navigation:

Shipping > Release Sales Order > Release Sales Orders.

Key in Based on Rule and Order Number

In the Shipping Tab key in the below:

Auto Create Delivery: Yes

Auto Pick Confirm: Yes

Auto Pack Delivery: Yes

In the Inventory Tab:

Auto Allocate: Yes

Enter the Warehouse

Click on Execute Now Button.

On successful completion, the below message would pop up as shown below.

Pick Release process in turn will kick off several other requests like Pick Slip Report,

Shipping Exception Report and Auto Pack Report

Underlying Tables affected:

If Autocreate Delivery is set to 'Yes' then a new record is created in the table WSH_NEW_DELIVERIES.
DELIVERY_ID is populated in the table WSH_DELIVERY_ASSIGNMENTS.
The RELEASED_STATUS in WSH_DELIVERY_DETAILS would be now set to 'Y' (Pick Confirmed) if Auto Pick Confirm is set to Yes otherwise RELEASED_STATUS is 'S' (Release to Warehouse).

Pick Confirm the Order:

IF Auto Pick Confirm in the above step is set to NO, then the following should be done.

Navigation:

Inventory Super User > Move Order> Transact Move Order

In the HEADER tab, enter the BATCH NUMBER (from the above step) of the order. Click FIND. Click on VIEW/UPDATE Allocation, then Click TRANSACT button. Then Transact button will be deactivated then just close it and go to next step.

Ship Confirm the Order:

Navigation:

Order Management Super User>Shipping >Transactions.

Query with the Order Number.

Click On Delivery Tab

Click on Ship Confirm.

The Status in Shipping Transaction screen will now be closed.

This will kick off concurrent programs like.INTERFACE TRIP Stop, Commercial Invoice, Packing Slip Report, Bill of Lading

Underlying tables affected:

RELEASED_STATUS in WSH_DELIVERY_DETAILS would be 'C' (Ship Confirmed)

FLOW_STATUS_CODE in OE_ORDER_HEADERS_ALL would be "BOOKED"

FLOW_STATUS_CODE in OE_ORDER_LINES_ALL would be "SHIPPED"

Create Invoice:

Run workflow background Process.

Navigation:

Order Management >view >Requests

Workflow Background Process inserts the records RA_INTERFACE_LINES_ALL with

INTERFACE_LINE_CONTEXT = 'ORDER ENTRY'
INTERFACE_LINE_ATTRIBUTE1= Order_number
INTERFACE_LINE_ATTRIBUTE3= Delivery_id

and spawns Auto invoice Master Program and Auto invoice import program which creates Invoice for that particular Order.

The Invoice created can be seen using the Receivables responsibility

Navigation:

Receivables Super User> Transactions> Transactions

Query with the Order Number as Reference.

Underlying tables:

RA_CUSTOMER_TRX_ALL will have the Invoice header information. The column INTERFACE_HEADER_ATTRIBUTE1 will have the Order Number.

RA_CUSTOMER_TRX_LINES_ALL will have the Invoice lines information. The column INTERFACE_LINE_ATTRIBUTE1 will have the Order Number.

Create receipt:

Navigation:

Receivables> Receipts> Receipts

Enter the information.

Click on Apply Button to apply it to the Invoice.

Underlying tables:

AR_CASH_RECEIPTS_ALL

Transfer to General Ledger:

To transfer the Receivables accounting information to general ledger, run General Ledger Transfer Program.

Navigation:

Receivables> View Requests

Parameters:

- Give in the Start date and Post through date to specify the date range of the transactions to be transferred.
- Specify the GL Posted Date, defaults to SYSDATE.
- Post in summary: This controls how Receivables creates journal entries for your transactions in the interface table. If you select 'No', then the General Ledger Interface program creates at least one journal entry in the interface table for each transaction in your posting submission. If you select 'Yes', then the program creates one journal entry for each general ledger account.
- If the Parameter Run Journal Import is set to 'Yes', the journal import program is kicked off automatically which transfers journal entries from the interface table to General Ledger, otherwise follow the topic **Journal Import** to import the journals to General Ledger manually.

.

Underlying tables:

This transfers data about your adjustments, chargeback, credit memos, commitments, debit memos, invoices, and receipts to the GL_INTERFACE table.

Journal Import:

To transfer the data from General Ledger Interface table to General Ledger, run the Journal Import program from Oracle General Ledger.

Navigation:

General Ledger > Journal> Import> Run

Parameters:

- Select the appropriate Source.
- Enter one of the following Selection Criteria:
  No Group ID: To import all data for that source that has no group ID. Use this option if you specified a NULL group ID for this source.

  All Group IDs: To import all data for that source that has a group ID. Use this option to import multiple journal batches for the same source with varying group IDs.

  Specific Group ID: To import data for a specific source/group ID combination. Choose a specific group ID from the List of Values for the Specific Value field.

  If you do not specify a Group ID, General Ledger imports all data from the specified journal entry source, where the Group_ID is null.
- Define the Journal Import Run Options (optional)
  Choose Post Errors to Suspense if you have suspense posting enabled for your set of books to post the difference resulting from any unbalanced journals to your suspense account.

  Choose Create Summary Journals to have journal import create the following:

  • one journal line for all transactions that share the same account, period, and currency and that has a debit balance

  • one journal line for all transactions that share the same account, period, and currency and that has a credit balance.
- Enter a Date Range to have General Ledger import only journals with accounting dates in that range. If you do not specify a date range, General Ledger imports all journals data.
- Choose whether to Import Descriptive Flexfields, and whether to import them with validation.

Click on Import button.

Underlying tables:

GL_JE_BATCHES, GL_JE_HEADERS, GL_JE_LINES

Posting:

We have to Post journal batches that we have imported previously to update the account balances in General Ledger.

Navigation:

General Ledger> Journals > Enter

Query for the unposted journals for a specific period as shown below.

From the list of unposted journals displayed, select one journal at a time and click on Post button to post the journal.

If you know the batch name to be posted you can directly post using the Post window

Navigation:

General Ledger> Journals> Post

Underlying tables:

GL_BALANCES.

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

Procure to Pay Cycle (P2P)

In this article, we will see the steps involved in Procure to Pay Cycle. Here is the diagrammatic representation:

1) Create Requisition:

Requisition is nothing but a formal request to buy something (like Inventory material, office supplies etc) needed for the enterprise. Only an employee can create one. There are *two types of requisitions:*

Internal Requisition: Internal requisitions provide the mechanism for requesting and transferring material from one inventory to other inventory.

Purchase requisition: Unlike Internal requisitions, Purchase requisitions are used for requesting material from suppliers.

Navigation:

Purchasing Vision Operations (USA) > Requisitions > Requisitions

Choose the requisition type and enter the Item, quantity, Price details in the Lines tab.

In Source Details tab, specify the Buyer name.

Click the Distributions button. Enter the Charge Account.

Save the work. The status of the requisition will now be “Incomplete”. And now the Approve button is highlighted. The requisition needs to be approved first before proceeding further by the concerned authority. Submit this requisition for Approval by clicking on the Approve button. The status will now be updated to “In Process” .The workflow then will send an Approval notification to the concerned person (derived based on hierarchy used—Position or Supervisor hierarchy) using which he can Approve or Reject the requisition.

At any time the *status of requisition* can be checked using the Requisition summary window.

Navigation:

Requisitions > Requisition Summary

Enter requisition number and click on the find button.

We can also check the *Action History of requisition* (it will show details about who has submitted, approved and cancelled the requisitions)

as below:

Navigation:

Tools menu > Action History.

Underlying Tables:

PO_REQUISITION_HEADERS_ALL

PO_REQUISITION_LINES_ALL

PO_REQ_DISTRIBUTIONS_ALL

2) Create Purchase Order:

There are 4 types of Purchase Orders:

1. **Standard PO:** A Standard PO is created for one–time purchase of various items

2. **Planned PO:** A Planned PO is a long–term agreement committing to buy items or services from a single source. You must specify tentative delivery schedules and all details for goods or services that you want to buy, including charge account, quantities, and estimated cost.

3. **Blanket agreement:** A Blanket PO is created when you know the detail of the goods or services you plan to buy from a specific supplier in a period, but you do not know the detail of your delivery schedules.

4. **Contract agreement:** Contract purchase agreements are created with your suppliers to agree

on specific terms and conditions without indicating the goods and services that you will be purchasing

Navigation for creating a standard PO:

Purchase Orders > Purchase Orders

Choose type as Standard Purchase Order. Enter the Supplier, Buyer. In the Lines tab, specify the line number, line type, Item, quantity, price etc.

Click Terms to enter terms, conditions, and control information for purchase orders.

Click Currency button to enter and change currency information for purchase orders, RFQs, and quotations.

Click Shipments button to enter multiple shipments for standard and planned purchase order lines Purchase order shipment specifies the quantity, ship–to organization and location, date you want your supplier to deliver the items on a purchase order line, and country of origin for the items. When you save, Purchasing creates distributions depending on the default information available.

To enter more shipment information, select the More tab.

- Enter the Receipt Close Tolerance percent, Invoice Close Tolerance percent to set the receiving and invoice close point.

- Select one of the following options for Match Approval Level:

Two–Way: Purchase order and invoice quantities must match within tolerance before the corresponding invoice can be paid.
Three–Way: Purchase order, receipt, and invoice quantities must match within tolerance before the corresponding invoice can be paid.
Four–Way: Purchase order, receipt, accepted, and invoice quantities must match within tolerance before the corresponding invoice can be paid.

- Select an Invoice Match Option:

Purchase Order: Payables must match the invoice to the purchase order.

Receipt: Payables must match the invoice to the receipt.

Save the work.

Click the Receiving Controls button to enter receiving control information for purchase orders.

- Enter the maximum acceptable number of Days Early and Days Late for receipts.

- Enter the Action for receipt date control.

- Enter the maximum acceptable over–receipt Tolerance percent (receipts that exceed the quantity received tolerance).

- Enter the Action for Overreceipt Quantity.

- Select Allow Substitute Receipts to indicate that receivers can receive substitute items in place of ordered items.

- Enter the default Receipt Routing that you assign goods: Direct Delivery**,** Inspection Required, or Standard Receipt.

- Enter the Enforce Ship To location option to determine whether the receiving location must be the same as the ship–to location.

Save the work.

Click Distributions button to enter distributions for the shipments.

Select more tab to enter more details and the requisition number (optional).

Save the work.

Click on the Approve button to initiate the Approval process.

Underlying Tables:

PO_HEADERS_ALL

PO_LINES_ALL

PO_DISTRIBUTIONS_ALL (REQ_HEADER_REFERENCE_NUM in Distributions table is the Requisition number for this PO)

3)Create Receipt:

Create a receipt to receive the items in the Purchase Order.

Navigation:

Receiving→Receipts

Enter the PO Number and select find button.

Go to Lines, check the lines you want to receive in the PO.

Click on Header button and Save which creates the receipt.

Receipt Tables are:

RCV_SHIPMENT_HEADERS

RCV_SHIPMENT_LINES (Lines Table has PO_HEADER_ID)

4)Create Invoice in Payables:

Once the goods are received, it's time to pay the vendor for the goods purchased and hence the invoices are created.

Navigation:

Payables, Vision Operations (USA) > Invoices→Entry→Invoices

Enter type — Standard, supplier information and amount.

Click the Match button to match to either Purchase Order or Receipt (depending on the Invoice Match option specified on the PO) and avoid manually entering the invoice.

Enter the PO Number you want match to and click Find.

Select the lines required and click on Match button.

Click on Distribute button
to navigate to the Match to Purchase Order Distributions window.

This creates the invoice and you can see the status of the invoice as "Never Validated". it has to be Validated and Accounted before you can pay it.

Validating the Invoice:

Click on Actions Button and Select “Validate”. Click on OK button.

Now you can see the status of the invoice as “Validated”, if there are no issues during validation.

Create Accounting Entries:

Click on Actions Button and Select “Create Accouting”. Click on OK button.

Now we can see the Accounted status as “Yes”.

You can see the Accounting Entries here:

Tools →View Accounting

Invoice Tables:

AP_INVOICES_ALL

AP_INVOICE_DISTRIBUTIONS_ALL

Accounting Entries Tables:

AP_ACCOUNTING_EVENTS_ALL

AP_AE_HEADERS_ALL

AP_AE_LINES_ALL

5)Making a Payment:

Go to the Invoice window and query the invoice you want to pay. You would see Amount paid as 0.00 before you make a payment.

Click Actions button. Select “Pay in full” and click “OK”.

Select the Bank Account and Document. Save the Work.

Now that the payment is made, when you query for the invoice in Invoice window, you will the Amount Paid as $4,000.00.

Create Accounting entries for payment.

Click Actions and select “Create Accounting”

Select the void checkbox to cancel the payment.

View Accounting Entries:

In the Payments window, query for the payment.

Tools menu→View Accounting

Payment Tables:

AP_INVOICE_PAYMENTS_ALL

AP_PAYMENT_SCHEDULES_ALL

AP_CHECKS_ALL

AP_CHECK_FORMATS

AP_BANK_ACCOUNTS_ALL

AP_BANK_BRANCHES

AP_TERMS

You can also pay the invoices using Payment Batch screen. Refer to the article "Make AP Payments through Payment Batches"

6)Transfer to General Ledger:

Navigation:

Payables Responsibility > View Requests

Run the concurrent program "Payables Transfer to General Ledger" with the required parameters.

Journal Import:

Refer to the Article "Order to Cash Cycle".

Posting:

Refer to the Article "Order to Cash Cycle".

Procure to pay Query :

Includes two scripts to fetch all the transactions information related with in a procure to pay cycle.
Two scripts are provided to use one with receipts and other when receipts are not created.
Few important fields that were included in the script are Requisition Number, Purchase Order Number, Invoice Number, Customer Number, Invoice Amount, GL Transfer flag e.t.c

WITH OUT RECEIPTS

–Procure to Pay query without receipts

```
select distinct

       reqh.segment1 REQ_NUM,

       reqh.AUTHORIZATION_STATUS REQ_STATUS,

--       poh.po_header_id,

       poh.segment1 PO_NUM,

       pol.line_num,

       poh.AUTHORIZATION_STATUS PO_STATUS,

--       i.invoice_id,

       i.invoice_num,

       i.invoice_amount,

       i.amount_paid,

        i.vendor_id,

--       v.vendor_name,
```

```
--       p.check_id,
       c.check_number,
       h.gl_transfer_flag,
       h.period_name
from ap_invoices_all i,
     ap_invoice_distributions_all invd,
     po_headers_all poh,
     po_lines_all pol,
     po_distributions_all pod,
     po_vendors v,
     po_requisition_headers_all reqh,
     po_requisition_lines_all reql,
     po_req_distributions_all reqd,
     ap_invoice_payments_all p,
     ap_checks_all c,
     ap_ae_headers_all h,
     ap_ae_lines_all l
where 1=1
and i.vendor_id = v.vendor_id
and c.check_id = p.check_id
and p.invoice_id = i.invoice_id
and poh.PO_HEADER_ID = pol.PO_HEADER_ID
and reqh.REQUISITION_HEADER_ID = reql.REQUISITION_HEADER_ID
and reqd.REQUISITION_LINE_ID = reql.REQUISITION_LINE_ID
and pod.REQ_DISTRIBUTION_ID = reqd.DISTRIBUTION_ID
and pod.PO_HEADER_ID = poh.PO_HEADER_ID
```

```
and pod.PO_DISTRIBUTION_ID = invd.PO_DISTRIBUTION_ID

and invd.INVOICE_ID = i.INVOICE_ID

and h.ae_header_id = l.ae_header_id

and l.SOURCE_TABLE = 'AP_INVOICES'

AND l.SOURCE_ID = i.invoice_id

--and poh.segment1 = 4033816 -- PO NUMBER

and reqh.segment1 = '501'    -- REQ NUMBER

--and i.invoice_num = 3114      -- INVOICE NUMBER

--and c.check_number =      -- CHECK NUMBER

    --and vendor_id =           -- VENDOR ID
```

WITH RECEIPTS

– PROCURE TO PAY CYCLE QUERY WITH RECEIPTS

```
SELECT DISTINCT reqh.segment1 req_num, reqh.authorization_status req_statu
s,

       --       POH.PO_HEADER_ID,

                 poh.segment1 po_num, pol.line_num,

                 poh.authorization_status po_status, rcvh.receipt_num,

                 rcv.inspection_status_code,

       --       I.INVOICE_ID,

                 i.invoice_num, i.invoice_amount,

                 i.amount_paid, i.vendor_id,

       --       V.VENDOR_NAME,

       --       P.CHECK_ID,

                 c.check_number, h.gl_transfer_flag,

                 h.period_name

            FROM ap_invoices_all i,
```

```
                ap_invoice_distributions_all invd,
                po_headers_all poh,
                po_lines_all pol,
                po_distributions_all pod,
                po_vendors v,
                po_requisition_headers_all reqh,
                po_requisition_lines_all reql,
                po_req_distributions_all reqd,
                rcv_transactions rcv,
                rcv_shipment_headers rcvh,
                rcv_shipment_lines rcvl,
                ap_invoice_payments_all p,
                ap_checks_all c,
                ap_ae_headers_all h,
                ap_ae_lines_all l
          WHERE 1 = 1
            AND i.vendor_id = v.vendor_id
            AND c.check_id = p.check_id
            AND p.invoice_id = i.invoice_id
            AND poh.po_header_id = pol.po_header_id
            AND reqh.requisition_header_id = reql.requisition_header_id
            AND reqd.requisition_line_id = reql.requisition_line_id
            AND pod.req_distribution_id = reqd.distribution_id
            AND pod.po_header_id = poh.po_header_id
            --AND POH.PO_HEADER_ID = RCV.PO_HEADER_ID
            AND rcvh.shipment_header_id = rcv.shipment_header_id(+)
```

```
--AND RCVH.SHIPMENT_HEADER_ID = RCVL.SHIPMENT_HEADER_ID

--AND RCV.TRANSACTION_TYPE = 'RECEIVE'

--AND RCV.SOURCE_DOCUMENT_CODE = 'PO'

--AND POL.PO_LINE_ID = RCV.PO_LINE_ID

--AND POD.PO_DISTRIBUTION_ID = RCV.PO_DISTRIBUTION_ID

AND pod.po_distribution_id = invd.po_distribution_id

AND invd.invoice_id = i.invoice_id

AND h.ae_header_id = l.ae_header_id

AND l.source_table = 'AP_INVOICES'

AND l.source_id = i.invoice_id

--AND POH.SEGMENT1 = 36420 -- PO NUMBER

AND reqh.segment1 = '501'  -- REQ NUMBER

--AND I.INVOICE_NUM = 3114     -- INVOICE NUMBER

--AND C.CHECK_NUMBER =     -- CHECK NUMBER

--AND VENDOR_ID =          -- VENDOR ID

--AND RECEIPT_NUM = 692237
```

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

Approval hierarchies for the Requisition in Purchasing

Approval hierarchies let you automatically route documents for approval. There are two kinds of approval hierarchies in Purchasing: position hierarchy and employee/supervisor relationships.

If an employee/supervisor relationship is used, the approval routing structures are defined as you enter employees using the Enter Person window. In this case, positions are not required to be setup.

If you choose to use position hierarchies, you must set up positions. Even though the position hierarchies require more initial effort to set up, they are easy to maintain and allow you to define approval routing structures that remain stable regardless of how frequently individual employees leave your organization or relocate within it.

Setups required for Position hierarchy in detail:

Say we want to implement the below position hierarchy for the Requisition Document approval routing in Purchasing (Each rectangle represents a Position), below are the setups required:

1) Enable the check box “Use Approval Hierarchies”

Navigation:

Purchasing Responsibility > Setup > Organizations > Financial Options

Click Human Resources tab. to use positions and position hierarchies to determine approval paths for your documents within Purchasing. (Disable this option if you want approval paths based on the supervisor structure)

2) Create Positions S20, MGR, DIR based on the above hierarchy in HR.

Navigation:

Human Resources responsibility > Work Structures > Position > Description

Click on New Button and enter the required fields.

Similarly create MGR and DIR positions.

3) Assign the positions created to the Employees who fall under the portfolio. Each position can

be assigned to more than one employee.

Navigation:

Human Resources responsibility > People > Enter and Maintain

Query for the Employee Name. Click on Find Button.

Click on Assignment button

Enter the desired value in the Position field.

Save the work.

4) Create the Position hierarchy

Navigation:

Human Resources Responsibility > Work Structures > Position > Hierarchy

- Enter Hierarchy Name – Save
- Enter the From Date – Save
- Click in the POSITION field and query the position you would like to be your Top position. In our example you would query for DIR.
- After the Position is selected Press the Down Key. Your cursor will now be in the subordinate region, choose the position MGR to go under DIR position.
- Again hit the Blue Down Arrow. You will now see the Subordinate position shift to where the Top Position was located. Add a new Position S20.
- Save the work.

5) Create Approval groups for each Position.

Approval Groups window lets you define the Approval limits for a particular group which will be assigned to a position.

As per our example, we will create three Approval Groups S20, MGR, DIR and define the limits.

Navigation:

Purchasing Responsibility > Setup > Approvals > Approval Groups

Enter the Name of the approval group

Select Enabled to permit the approval group to be assigned to a position in the Approval Assignments window.

Choose one of the following Objects:

Account Range – (Required) You enter the accounting flexfields for the Low and High

Values.

Document Total – (Required) The document total refers to the monetary limit on an individual document.
Item Category Range – You enter the purchasing category flexfields for the Low and High Values.
Item Range – For this option, you enter the item flexfields for the Low and High Values.
Location – The location refers to the deliver-to location on a requisition as well as the ship-to location on purchase orders and releases.

Select the rule Type: Include or Exclude indicates whether to allow objects that fall within the selected range.

Enter the Amount Limit. This is the maximum amount that a control group can authorize for a particular object range.

Enter the Low Value. This is the lowest flexfield (accounting, purchasing category, or item) in the range pertinent to this rule. When the object is Location, enter the location. You cannot enter this field when the object is Document Total.

Enter the High Value. This is the highest flexfield (accounting, purchasing category, or item) in the range pertinent to this rule. You cannot enter this field when the object is Location or Document Total.

Save your work.

6) Assign the Approval groups to the Position

Navigation:

Purchasing Responsibility > Setup > Approvals > Approval Assignments

Query the Position for which you want to assign the Approval group.

Select the Document type you want to assign to this position or job (As per our example we will choose “Approve Purchase Requisition”)

Enter the approval group that you want to assign to the selected position or job. The list of values includes only enabled approval groups with at least one approval rule. (Assign it to the approval group DIR as per our example)

Enter the Start Date and End Date for the assignment.

Save your work.

Similarly assign the Approval groups MGR and S20 to the positions MGR and S20 respectively for the document type “Approve Purchase Requisition”.

7) Assign the Position hierarchy created to the desired Document Type.

Document Types window can be used to define access, security, and control specifications for all Purchasing documents.

Navigation:

Purchasing Responsibility > Setup > Purchasing > Document Types

Find Document Types window appears. Select the document type “Requisition” .

- Enter your Document Name for the document. The description must be unique for the given document type.
- Check Owner Can Approve to indicate that document preparers can approve their own documents.
- Check Approver Can Modify to indicate that document approvers can modify documents. Check Can Change Forward-To to indicate that users can change the person the document is forwarded to.
- Check Can Change Forward-From to indicate that users can change the name of the document creator. This field is applicable only when the Document Type is Requisition.
- Check Can Change Approval Hierarchy to indicate that approvers can change the approval hierarchy in the Approve Documents window. This field is not applicable when the Document Type is Quotation
  or RFQ.
- Check Disable to disable a document type.
- For Purchase requisitions only, select Use Contract Agreements for Auto Sourcing to require the requisition creation autosourcing logic to include approved contract purchase agreements.
- *Include Non-Catalog Requests* – For Oracle iProcurement only, this checkbox is used in conjunction with the Use Contract Agreements for Auto Sourcing. Select this checkbox to enable the use of contract purchase agreements when autosourcing non-catalog requisitions.
- Choose one of the following options:
  Hierarchy – Only the document owner and users above the owner in the defined purchasing security hierarchy may access these documents.

Private – Only the document owner may access these documents.

Public – Any user may access these documents.
Purchasing – Only the document owner and users listed as buyers in the Define Buyers

window may access these documents.

- Choose one of the following Access Level options:

Full – Users can view, modify, cancel, and final close documents.

Modify – Users can only view and modify documents.
View Only – Users can only view documents.

- Choose one of the following options:

Direct – The default approver is the first person in the preparer's approval path that has

sufficient approval authority.
Hierarchy – The default approver is the next person in the preparer's approval path

regardless of authority. (Each person in the approval path must take approval action until the person with sufficient approval authority is reached.)

- Choose the Position Hierarchy that we have created previously "Purchasing Approval Test" as the Default Hierarchy field.

8) Run the 'Fill Employee Hierarchy' Request

Navigation:

Purchasing Responsibility > View Requests

Click on Submit a New Request button and select Single Request.

Approval routing explanation based on the example hierarchy:

1. Say the S20 position creates a Purchase Order for $4,000 and submits for Approval.

2. The system will look at which Approval Hierarchy to be used from the Document Types window. Since in our example "Owner Can Approve" is not checked, it will try to determine what the Approval group is for the position MGR as MGR is the next Position in our hierarchy.

3. Once the Approval Group is located, the rules will then be considered one at a time, from first to last.

4. If all rules are satisfied, the Purchase Order will be forwarded to the Manager for Approval.

If a rule cannot be satisfied, the Purchase Order will be forwarded to the next position (DIR) in the Hierarchy. Whoever is assigned to that position will then need to take action on the Purchase Order.

In release 11.0, when attempting to Approve a Purchase Order, if the system doesn't find any of the positions defined in the hierarchy to have the authority to Approve, then the Document will remain 'Incomplete' without any warnings instead of 'In Process'.

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

PO QUERIES

Technical Queries related to Oracle Purchasing

] TO LIST OUT ALL CANCEL REQUISITIONS:->> list My cancel Requistion

```
select prh.REQUISITION_HEADER_ID, prh.PREPARER_ID , prh.SEGMENT1 "REQ
NUM", trunc(prh.CREATION_DATE), prh.DESCRIPTION,
prh.NOTE_TO_AUTHORIZERfrom apps.Po_Requisition_headers_all prh,
apps.po_action_history pah where Action_code='CANCEL' and
pah.object_type_code='REQUISITION'
and pah.object_id=prh.REQUISITION_HEADER_ID
```

2] TO LIST ALL INTERNAL REQUISITIONS THAT DO NOT HAVE AN ASSOCIATED INTERNAL SALES ORDER

```
>> Select RQH.SEGMENT1
REQ_NUM,RQL.LINE_NUM,RQL.REQUISITION_HEADER_ID
,RQL.REQUISITION_LINE_ID,RQL.ITEM_ID ,RQL.UNIT_MEAS_LOOKUP_CODE
,RQL.UNIT_PRICE ,RQL.QUANTITY
,RQL.QUANTITY_CANCELLED,RQL.QUANTITY_DELIVERED ,RQL.CANCEL_FLAG
,RQL.SOURCE_TYPE_CODE ,RQL.SOURCE_ORGANIZATION_ID
,RQL.DESTINATION_ORGANIZATION_ID,RQH.TRANSFERRED_TO_OE_FLAGfromPO_
REQUISITION_LINES_ALL RQL, PO_REQUISITION_HEADERS_ALL
RQHwhereRQL.REQUISITION_HEADER_ID = RQH.REQUISITION_HEADER_IDand
RQL.SOURCE_TYPE_CODE = 'INVENTORY'and RQL.SOURCE_ORGANIZATION_ID is
not nulland not exists (select 'existing internal order'from OE_ORDER_LINES_ALL
LINwhere LIN.SOURCE_DOCUMENT_LINE_ID = RQL.REQUISITION_LINE_IDand
LIN.SOURCE_DOCUMENT_TYPE_ID = 10)
ORDER BY RQH.REQUISITION_HEADER_ID, RQL.LINE_NUM;
```

3] Display what requisition and PO are linked(Relation with Requisition and PO)>>

```
select r.segment1 "Req Num", p.segment1 "PO Num"from po_headers_all p,
po_distributions_all d,po_req_distributions_all rd, po_requisition_lines_all
rl,po_requisition_headers_all r where p.po_header_id = d.po_header_id and
d.req_distribution_id = rd.distribution_id and rd.requisition_line_id = rl.requisition_line_id
and rl.requisition_header_id = r.requisition_header_id
```

4] List all Purchase Requisition without a Purchase Order that means a PR has not been autocreated to PO. (Purchase Requisition without a Purchase Order)>>

```
select prh.segment1 "PR NUM", trunc(prh.creation_date) "CREATED ON",
trunc(prl.creation_date) "Line Creation Date" , prl.line_num "Seq #", msi.segment1 "Item
Num", prl.item_description "Description", prl.quantity "Qty", trunc(prl.need_by_date)
"Required By", ppf1.full_name "REQUESTOR", ppf2.agent_name "BUYER" from
po.po_requisition_headers_all prh, po.po_requisition_lines_all prl, apps.per_people_f
ppf1, (select distinct agent_id,agent_name from apps.po_agents_v ) ppf2,
po.po_req_distributions_all prd, inv.mtl_system_items_b msi, po.po_line_locations_all pll,
po.po_lines_all pl, po.po_headers_all ph WHERE prh.requisition_header_id =
prl.requisition_header_id and prl.requisition_line_id = prd.requisition_line_id and
ppf1.person_id = prh.preparer_id and prh.creation_date between
ppf1.effective_start_date and ppf1.effective_end_date and ppf2.agent_id(+) =
msi.buyer_id and msi.inventory_item_id = prl.item_id and msi.organization_id =
```

prl.destination_organization_id and pll.line_location_id(+) = prl.line_location_id and pll.po_header_id = ph.po_header_id(+) AND PLL.PO_LINE_ID = PL.PO_LINE_ID(+) AND PRH.AUTHORIZATION_STATUS = 'APPROVED' AND PLL.LINE_LOCATION_ID IS NULL AND PRL.CLOSED_CODE IS NULL AND NVL(PRL.CANCEL_FLAG,'N') <> 'Y'
ORDER BY 1,2

5] list all information form PR to PO …as a requisition moved from different stages till converting into PR. This query capture all details related to that PR to PO.>> LIST AND ALL DATA ENTRY FROM PR TILL PO
select distinct u.description "Requestor", porh.segment1 as "Req Number", trunc(porh.Creation_Date) "Created On", pord.LAST_UPDATED_BY, porh.Authorization_Status "Status", porh.Description "Description", poh.segment1 "PO Number", trunc(poh.Creation_date) "PO Creation Date", poh.AUTHORIZATION_STATUS "PO Status", trunc(poh.Approved_Date) "Approved Date"from apps.po_headers_all poh, apps.po_distributions_all pod, apps.po_req_distributions_all pord, apps.po_requisition_lines_all porl, apps.po_requisition_headers_all porh, apps.fnd_user u where porh.requisition_header_id = porl.requisition_header_id and porl.requisition_line_id = pord.requisition_line_id and pord.distribution_id = pod.req_distribution_id(+) and pod.po_header_id = poh.po_header_id(+) and porh.created_by = u.user_id
order by 2

6] Identifying all PO's which does not have any PR's>>LIST ALL PURCHASE REQUISITION WITHOUT A PURCHASE ORDER THAT MEANS A PR HAS NOT BEEN AUTOCREATED TO PO. select prh.segment1 "PR NUM", trunc(prh.creation_date) "CREATED ON", trunc(prl.creation_date) "Line Creation Date" , prl.line_num "Seq #", msi.segment1 "Item Num", prl.item_description "Description", prl.quantity "Qty", trunc(prl.need_by_date) "Required By", ppf1.full_name "REQUESTOR", ppf2.agent_name "BUYER" from po.po_requisition_headers_all prh, po.po_requisition_lines_all prl, apps.per_people_f ppf1, (select distinct agent_id,agent_name from apps.po_agents_v) ppf2, po.po_req_distributions_all prd, inv.mtl_system_items_b msi, po.po_line_locations_all pll, po.po_lines_all pl, po.po_headers_all ph WHERE prh.requisition_header_id = prl.requisition_header_id and prl.requisition_line_id = prd.requisition_line_id and ppf1.person_id = prh.preparer_id and prh.creation_date between ppf1.effective_start_date and ppf1.effective_end_date and ppf2.agent_id(+) = msi.buyer_id and msi.inventory_item_id = prl.item_id and msi.organization_id = prl.destination_organization_id and pll.line_location_id(+) = prl.line_location_id and pll.po_header_id = ph.po_header_id(+) AND PLL.PO_LINE_ID = PL.PO_LINE_ID(+) AND PRH.AUTHORIZATION_STATUS = 'APPROVED' AND PLL.LINE_LOCATION_ID IS NULL AND PRL.CLOSED_CODE IS NULL AND NVL(PRL.CANCEL_FLAG,'N') <> 'Y' ORDER BY 1,2

7] Relation between Requisition and PO tables>>Here is link:
PO_DISTRIBUTIONS_ALL =>PO_HEADER_ID, REQ_DISTRIBUTION_IDPO_HEADERS_ALL=>PO_HEADER_ID, SEGMENT1PO_REQ_DISTRIBUTIONS_ALL =>DISTRIBUTION_ID, REQUISITION_LINE_IDPO_REQUISITION_LINES_ALL =>REQUISITION_LINE_ID)PO_REQUISITION_HEADERS_ALL =>REQUISITION_HEADER_ID, REQUISITION_LINE_ID, SEGMENT1
What you have to make a join on PO_DISTRIBUTIONS_ALL (REQ_DISTRIBUTION_ID) and PO_REQ_DISTRIBUTIONS_ALL (DISTRIBUTION_ID) to see if there is a PO for the req.

--You need to find table which hold PO Approval path…
These two table keeps the data:
PO_APPROVAL_LIST_HEADERS
PO_APPROVAL_LIST_LINES

8] List all the PO's with there approval ,invoice and Payment Details>>LIST AND PO WITH THERE APPROVAL , INVOICE AND PAYMENT DETAILSselect a.org_id "ORG ID", E.SEGMENT1 "VENDOR NUM",e.vendor_name "SUPPLIER NAME",UPPER(e.vendor_type_lookup_code) "VENDOR TYPE", f.vendor_site_code "VENDOR SITE CODE",f.ADDRESS_LINE1 "ADDRESS",f.city "CITY",f.country "COUNTRY", to_char(trunc(d.CREATION_DATE)) "PO Date", d.segment1 "PO NUM",d.type_lookup_code "PO Type", c.quantity_ordered "QTY ORDERED", c.quantity_cancelled "QTY CANCELLED", g.item_id "ITEM ID" , g.item_description "ITEM DESCRIPTION",g.unit_price "UNIT PRICE", (NVL(c.quantity_ordered,0)-NVL(c.quantity_cancelled,0))*NVL(g.unit_price,0) "PO Line Amount", (select decode(ph.approved_FLAG, 'Y', 'Approved') from po.po_headers_all ph where ph.po_header_ID = d.po_header_id)"PO Approved?", a.invoice_type_lookup_code "INVOICE TYPE",a.invoice_amount "INVOICE AMOUNT", to_char(trunc(a.INVOICE_DATE)) "INVOICE DATE", a.invoice_num "INVOICE NUMBER", (select decode(x.MATCH_STATUS_FLAG, 'A', 'Approved') from ap.ap_invoice_distributions_all x where x.INVOICE_DISTRIBUTION_ID = b.invoice_distribution_id)"Invoice Approved?", a.amount_paid,h.amount, h.check_id, h.invoice_payment_id "Payment Id", i.check_number "Cheque Number", to_char(trunc(i.check_DATE)) "PAYMENT DATE" FROM AP.AP_INVOICES_ALL A, AP.AP_INVOICE_DISTRIBUTIONS_ALL B, PO.PO_DISTRIBUTIONS_ALL C, PO.PO_HEADERS_ALL D, PO.PO_VENDORS E, PO.PO_VENDOR_SITES_ALL F, PO.PO_LINES_ALL G, AP.AP_INVOICE_PAYMENTS_ALL H, AP.AP_CHECKS_ALL I where a.invoice_id = b.invoice_id and b.po_distribution_id = c. po_distribution_id (+) and c.po_header_id = d.po_header_id (+) and e.vendor_id (+) = d.VENDOR_ID and f.vendor_site_id (+) = d.vendor_site_id and d.po_header_id = g.po_header_id and c.po_line_id = g.po_line_id and a.invoice_id = h.invoice_id and h.check_id = i.check_id and f.vendor_site_id = i.vendor_site_id and c.PO_HEADER_ID is not null and a.payment_status_flag = 'Y'
and d.type_lookup_code != 'BLANKET'

10] To know the link to GL_JE_LINES table for purchasing accrual and budgetary control actions..The budgetary (encumbrance) and accrual actions in the purchasing module generate records that will be imported into GL for the corresponding accrual and budgetary journals.
The following reference fields are used to capture and keep PO information in the GL_JE_LINES table.
These reference fields are populated when the Journal source (JE_SOURCE in GL_JE_HEADERS) isPurchasing.
Budgetary Records from PO (These include reservations, reversals and cancellations):
REFERENCE_1- Source (PO or REQ)
REFERENCE_2- PO Header ID or Requisition Header ID (from po_headers_all.po_header_id orpo_requisition_headers_all.requisition_header_id)
REFERENCE_3- Distribution ID (from po_distributions_all.po_distribution_id orpo_req_distributions_all.distribution_id)
REFERENCE_4- Purchase Order or Requisition number (from po_headers_all.segment1 orpo_requisition_headers_all.segment1)

REFERENCE_5- (Autocreated Purchase Orders only) Backing requisition number (from po_requisition_headers_all.segment1)
Accrual Records from PO:
REFERENCE_1- Source (PO)
REFERENCE_2- PO Header ID (from po_headers_all.po_header_id)
REFERENCE_3- Distribution ID (from po_distributions_all.po_distribution_id
REFERENCE_4- Purchase Order number (from po_headers_all.segment1)
REFERENCE_5- (ON LINE ACCRUALS ONLY) Receiving Transaction ID (from rcv_receiving_sub_ledger.rcv_transaction_id)
Take a note for Period end accruals, the REFERENCE_5 column is not used.

11] List all open PO'S>> select h.segment1 "PO NUM", h.authorization_status "STATUS", l.line_num "SEQ NUM", ll.line_location_id, d.po_distribution_id , h.type_lookup_code "TYPE" from po.po_headers_all h, po.po_lines_all l, po.po_line_locations_all ll, po.po_distributions_all d where h.po_header_id = l.po_header_id and ll.po_line_id = l.po_Line_id and ll.line_location_id = d.line_location_id and h.closed_date is null and h.type_lookup_code not in ('QUOTATION')

12] There are different authorization_status can a requisition have.Approved
Cancelled
In Process
Incomplete
Pre-Approved
Rejected
and you should note: When we finally close the requisition from Requisition Summary form the authorization_status of the requisition does not change. Instead it’s closed_code becomes ‘FINALLY CLOSED’.

13] A standard Quotations one that you can tie back to a PO.Navigate to RFQ -> Auto create -> enter a PO and reference it back.14] To debug for a PO , where should I start.Thats is possible, your PO get stuck somewhere, so what you have to do is to analyze which stage it stucked.Get po_header_id first and run each query and then analyze the data.For better understanding this is splited into 5 major stages.
Stage 1: PO Creation :
PO_HEADERS_ALL
select po_header_id from po_headers_all where segment1 =;
select * from po_headers_all where po_header_id =;
po_lines_all
select * from po_lines_all where po_header_id =;
po_line_locations_all
select * from po_line_locations_all where po_header_id =;
po_distributions_all
select * from po_distributions_all where po_header_id =;
po_releases_all
SELECT * FROM po_releases_all WHERE po_header_id =;
Stage 2: Once PO is received data is moved to respective receving tables and inventory tables
RCV_SHIPMENT_HEADERS
select * from rcv_shipment_headers where shipment_header_id in(select shipment_header_id from rcv_shipment_lineswhere po_header_id =);
RCV_SHIPMENT_LINES

select * from rcv_shipment_lines where po_header_id =;
RCV_TRANSACTIONS
select * from rcv_transactions where po_header_id =;
RCV_ACCOUNTING_EVENTS
SELECT * FROM rcv_Accounting_Events WHERE rcv_transaction_id IN(select transaction_id from rcv_transactionswhere po_header_id =);
RCV_RECEIVING_SUB_LEDGER
select * from rcv_receiving_sub_ledger where rcv_transaction_id in (select transaction_id from rcv_transactions where po_header_id =);
RCV_SUB_LEDGER_DETAILS
select * from rcv_sub_ledger_detailswhere rcv_transaction_id in (select transaction_id from rcv_transactions where po_header_id =);
MTL_MATERIAL_TRANSACTIONS
select * from mtl_material_transactions where transaction_source_id =;
MTL_TRANSACTION_ACCOUNTS
select * from mtl_transaction_accounts where transaction_id in (select transaction_id from mtl_material_transactions where transaction_source_id = =);
Stage 3: Invoicing details
AP_INVOICE_DISTRIBUTIONS_ALL
select * from ap_invoice_distributions_all where po_distribution_id in (select po_distribution_id from po_distributions_all where po_header_id =);
AP_INVOICES_ALL
select * from ap_invoices_all where invoice_id in(select invoice_id from ap_invoice_distributions_all where po_distribution_id in(select po_distribution_id from po_distributions_all where po_header_id =));
Stage 4 : Many Time there is tie up with Project related PO
PA_EXPENDITURE_ITEMS_ALL
select * from pa_expenditure_items_all peia where peia.orig_transaction_reference in(select to_char(transaction_id) from mtl_material_transactionswhere transaction_source_id =);
Stage 5 : General Ledger
Prompt 17. GL_BC_PACKETS ..This is for encumbrances
SELECT * FROM gl_bc_packets WHERE reference2 IN (‘‘);
GL_INTERFACE
SELECT *FROM GL_INTERFACE GLIWHERE user_je_source_name =’Purchasing’AND gl_sl_link_table =’RSL’AND reference21=’PO’AND EXISTS(SELECT 1FROM rcv_receiving_sub_ledger RRSLWHERE GLI.reference22 =RRSL.reference2AND GLI.reference23 =RRSL.reference3AND GLI.reference24 =RRSL.reference4AND RRSL.rcv_transaction_id in(select transaction_id from rcv_transactionswhere po_header_id));
GL_IMPORT_REFERENCES
SELECT *FROM gl_import_references GLIRWHERE reference_1=’PO’AND gl_sl_link_table =’RSL’AND EXISTS(SELECT 1FROM rcv_receiving_sub_ledger RRSLWHERE GLIR.reference_2 =RRSL.reference2AND GLIR.reference_3 =RRSL.reference3AND GLIR.reference_4 =RRSL.reference4AND RRSL.rcv_transaction_id in(select transaction_id from rcv_transactions where po_header_id =))

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012 TCA (Trading Community Architecture)

Overview:

Trading Community Architecture (TCA) is an architecture concept designed to support complex trading communities. This document provides information about how to create a customer using TCA API. These APIs utilize the new TCA model, inserting directly to the HZ tables.

[http://1.bp.blogspot.com/-N3VWhPDL0aM/UD2YSmOvdul/AAAAAAAAAQ0/8W4bdQ1z-vk/s1600/hz_tables_in_ar.jpg]

Architecture

Create Organization

```
DECLARE
p_organization_rec    hz_party_v2pub.organization_rec_type;
x_return_status       VARCHAR2 (2000);
x_msg_count           NUMBER;
x_msg_data            VARCHAR2 (2000);
x_party_id            NUMBER;
x_party_number        VARCHAR2 (2000);
x_profile_id          NUMBER;
BEGIN
p_organization_rec.organization_name := 'erpschools';
p_organization_rec.created_by_module := 'ERPSCHOOLS_DEMO';
hz_party_v2pub.create_organization ('T',
p_organization_rec,
x_return_status,
x_msg_count,
x_msg_data,
x_party_id,
x_party_number,
x_profile_id
);
DBMS_OUTPUT.put_line ('party id ' || x_party_id);
DBMS_OUTPUT.put_line (SUBSTR ('x_return_status = ' || x_return_status,
1,
```

```
255

)

);

DBMS_OUTPUT.put_line ('x_msg_count = ' || TO_CHAR (x_msg_count));

DBMS_OUTPUT.put_line (SUBSTR ('x_msg_data = ' || x_msg_data, 1, 255));

IF x_msg_count > 1

THEN

FOR i IN 1 .. x_msg_count

LOOP

DBMS_OUTPUT.put_line

(   i

|| '. '

|| SUBSTR

(fnd_msg_pub.get (p_encoded       => fnd_api.g_false),

1,

255

)

);

END LOOP;

END IF;

END;
```

Note: The above API creates a record in hz_parties table and one record in hz_organization_profiles table. Similarly you can call hz_party_v2pub.create_person to create a record in the HZ_PARTIES and one record in HZ_PERSON_PROFILES tables.

Create a Location

```
DECLARE

p_location_rec HZ_LOCATION_V2PUB.LOCATION_REC_TYPE;
```

```
x_location_id NUMBER;
x_return_status VARCHAR2(2000);
x_msg_count NUMBER;
x_msg_data VARCHAR2(2000);
BEGIN
p_location_rec.country := 'US';
p_location_rec.address1 := '2500 W Higgins Rd';
p_location_rec.address2 := 'Suite 920';
p_location_rec.city := 'Thumuluru';
p_location_rec.postal_code := '60118';
p_location_rec.state := 'IL';
p_location_rec.created_by_module := 'ERPSCHOOLS_DEMO';
hz_location_v2pub.create_location(
'T',
p_location_rec,
x_location_id,
x_return_status,
x_msg_count,
x_msg_data);
dbms_output.put_line('location id '||x_location_id);
dbms_output.put_line(SubStr('x_return_status = '||x_return_status,1,255));
dbms_output.put_line('x_msg_count = '||TO_CHAR(x_msg_count));
dbms_output.put_line(SubStr('x_msg_data = '||x_msg_data,1,255));
IF x_msg_count >1 THEN
FOR I IN 1..x_msg_count
```

LOOP

dbms_output.put_line(l||'.
'||SubStr(FND_MSG_PUB.Get(p_encoded =>FND_API.G_FALSE), 1, 255));

END LOOP;

END IF;

END

Note: The above API shall create an address record in hz_locations table.

Create a Party Site:

Use the organization_id and location_id created above and create a party site.

DECLARE

p_party_site_rec hz_party_site_v2pub.party_site_rec_type;

x_party_site_id NUMBER;

x_party_site_number VARCHAR2 (2000);

x_return_status VARCHAR2 (2000);

x_msg_count NUMBER;

x_msg_data VARCHAR2 (2000);

BEGIN

p_party_site_rec.party_id := 1272023;

p_party_site_rec.location_id := 359086;

p_party_site_rec.identifying_address_flag := 'Y';

p_party_site_rec.created_by_module := 'ERPSCHOOLS_DEMO';

hz_party_site_v2pub.create_party_site ('T',

p_party_site_rec,

x_party_site_id,

x_party_site_number,

x_return_status,

```
x_msg_count,
x_msg_data
);
DBMS_OUTPUT.put_line ('party site id ' || x_party_site_id);
DBMS_OUTPUT.put_line (SUBSTR ('x_return_status = ' || x_return_status,
1,
255
)
);
DBMS_OUTPUT.put_line ('x_msg_count = ' || TO_CHAR (x_msg_count));
DBMS_OUTPUT.put_line (SUBSTR ('x_msg_data = ' || x_msg_data, 1, 255));
IF x_msg_count > 1
THEN
FOR i IN 1 .. x_msg_count
LOOP
DBMS_OUTPUT.put_line
(   i
|| '. '
|| SUBSTR
(fnd_msg_pub.get (p_encoded       => fnd_api.g_false),
1,
255
)
);
END LOOP;
END IF;
```

```
END;
```

Note: The above API creates a record in hz_party_sites table.

Create Party Site Use

Use the above party site created

```
DECLARE
p_party_site_use_rec    hz_party_site_v2pub.party_site_use_rec_type;
x_party_site_use_id     NUMBER;
x_return_status         VARCHAR2 (2000);
x_msg_count             NUMBER;
x_msg_data              VARCHAR2 (2000);
BEGIN
p_party_site_use_rec.site_use_type := 'SHIP_TO';
p_party_site_use_rec.party_site_id := 349327;
p_party_site_use_rec.created_by_module := 'ERPSCHOOLS_DEMO';
hz_party_site_v2pub.create_party_site_use ('T',
p_party_site_use_rec,
x_party_site_use_id,
x_return_status,
x_msg_count,
x_msg_data
);
DBMS_OUTPUT.put_line (SUBSTR ('x_return_status = ' || x_return_status,
1,
255
)
```

```
);

DBMS_OUTPUT.put_line ('x_msg_count = ' || TO_CHAR (x_msg_count));

DBMS_OUTPUT.put_line (SUBSTR ('x_msg_data = ' || x_msg_data, 1, 255));

IF x_msg_count > 1

THEN

FOR i IN 1 .. x_msg_count

LOOP

DBMS_OUTPUT.put_line

(   i

|| '. '

|| SUBSTR

(fnd_msg_pub.get (p_encoded       => fnd_api.g_false),

1,

255

)

);

END LOOP;

END IF;

END;
```

Create a Contact Point

```
DECLARE

p_contact_point_rec    hz_contact_point_v2pub.contact_point_rec_type;

p_edi_rec              hz_contact_point_v2pub.edi_rec_type;

p_email_rec            hz_contact_point_v2pub.email_rec_type;

p_phone_rec            hz_contact_point_v2pub.phone_rec_type;

p_telex_rec            hz_contact_point_v2pub.telex_rec_type;
```

```
p_web_rec             hz_contact_point_v2pub.web_rec_type;
x_return_status       VARCHAR2 (2000);
x_msg_count           NUMBER;
x_msg_data            VARCHAR2 (2000);
x_contact_point_id    NUMBER;
BEGIN
p_contact_point_rec.contact_point_type := 'PHONE';
p_contact_point_rec.owner_table_name := 'HZ_PARTIES';
p_contact_point_rec.owner_table_id := '1272023';
p_contact_point_rec.primary_flag := 'Y';
p_contact_point_rec.contact_point_purpose := 'BUSINESS';
p_phone_rec.phone_area_code := '650';
p_phone_rec.phone_country_code := '1';
p_phone_rec.phone_number := '506-7000';
p_phone_rec.phone_line_type := 'GEN';
p_contact_point_rec.created_by_module := 'ERPSCHOOLS_DEMO';
hz_contact_point_v2pub.create_contact_point ('T',
p_contact_point_rec,
p_edi_rec,
p_email_rec,
p_phone_rec,
p_telex_rec,
p_web_rec,
x_contact_point_id,
x_return_status,
```

```
x_msg_count,
x_msg_data
);
DBMS_OUTPUT.put_line (SUBSTR ('x_return_status = ' || x_return_status,
1,
255
)
);
DBMS_OUTPUT.put_line ('x_msg_count = ' || TO_CHAR (x_msg_count));
DBMS_OUTPUT.put_line (SUBSTR ('x_msg_data = ' || x_msg_data, 1, 255));
IF x_msg_count > 1
THEN
FOR i IN 1 .. x_msg_count
LOOP
DBMS_OUTPUT.put_line
(   i
|| '. '
|| SUBSTR
(fnd_msg_pub.get (p_encoded       => fnd_api.g_false),
1,
255
)
);
END LOOP;
END IF;
END;
```

Create an Org Contact:

```
DECLARE

p_org_contact_rec    hz_party_contact_v2pub.org_contact_rec_type;

x_org_contact_id     NUMBER;

x_party_rel_id       NUMBER;

x_party_id           NUMBER;

x_party_number       VARCHAR2 (2000);

x_return_status      VARCHAR2 (2000);

x_msg_count          NUMBER;

x_msg_data           VARCHAR2 (2000);

BEGIN

p_org_contact_rec.department_code := 'ACCOUNTING';

p_org_contact_rec.job_title := 'ACCOUNTS OFFICER';

p_org_contact_rec.decision_maker_flag := 'Y';

p_org_contact_rec.job_title_code := 'APC';

p_org_contact_rec.created_by_module := 'ERPSCHOOLS_DEMO';

p_org_contact_rec.party_rel_rec.subject_id := 16077;

p_org_contact_rec.party_rel_rec.subject_type := 'PERSON';

p_org_contact_rec.party_rel_rec.subject_table_name := 'HZ_PARTIES';

p_org_contact_rec.party_rel_rec.object_id := 1272023;

p_org_contact_rec.party_rel_rec.object_type := 'ORGANIZATION';

p_org_contact_rec.party_rel_rec.object_table_name := 'HZ_PARTIES';

p_org_contact_rec.party_rel_rec.relationship_code := 'CONTACT_OF';

p_org_contact_rec.party_rel_rec.relationship_type := 'CONTACT';

p_org_contact_rec.party_rel_rec.start_date := SYSDATE;
```

```
hz_party_contact_v2pub.create_org_contact ('T',

p_org_contact_rec,

x_org_contact_id,

x_party_rel_id,

x_party_id,

x_party_number,

x_return_status,

x_msg_count,

x_msg_data

);

DBMS_OUTPUT.put_line (SUBSTR ('x_return_status = ' || x_return_status,

1,

255

)

);

DBMS_OUTPUT.put_line ('x_msg_count = ' || TO_CHAR (x_msg_count));

DBMS_OUTPUT.put_line (SUBSTR ('x_msg_data = ' || x_msg_data, 1, 255));

IF x_msg_count > 1

THEN

FOR i IN 1 .. x_msg_count

LOOP

DBMS_OUTPUT.put_line

(   i

|| '. '

|| SUBSTR

(fnd_msg_pub.get (p_encoded       => fnd_api.g_false),
```

```
1,

255

)

);

END LOOP;

END IF;

END;
```

Note: The above API creates a record in hz_org_contacts table and one record in hz_relationships table. When a contact is created, a record in hz_parties table gets created with party_type as 'PARTY_RELATIONSHIP'.

Create a Customer Account:

```
DECLARE

p_cust_account_rec       hz_cust_account_v2pub.cust_account_rec_type;

p_person_rec             hz_party_v2pub.person_rec_type;

p_customer_profile_rec   hz_customer_profile_v2pub.customer_profilerec_type;

x_cust_account_id        NUMBER;

x_account_number         VARCHAR2 (2000);

x_party_id               NUMBER;

x_party_number           VARCHAR2 (2000);

x_profile_id             NUMBER;

x_return_status          VARCHAR2 (2000);

x_msg_count              NUMBER;

x_msg_data               VARCHAR2 (2000);

BEGIN

p_cust_account_rec.account_name := 'John"s A/c';

p_cust_account_rec.created_by_module := 'ERPSCHOOLS_DEMO';
```

```
p_person_rec.person_first_name := 'John';

p_person_rec.person_last_name := 'Smith';

hz_cust_account_v2pub.create_cust_account ('T',

p_cust_account_rec,

p_person_rec,

p_customer_profile_rec,

'F',

x_cust_account_id,

x_account_number,

x_party_id,

x_party_number,

x_profile_id,

x_return_status,

x_msg_count,

x_msg_data

);

DBMS_OUTPUT.put_line (SUBSTR ('x_return_status = ' || x_return_status,

1,

255

)

);

DBMS_OUTPUT.put_line ('x_msg_count = ' || TO_CHAR (x_msg_count));

DBMS_OUTPUT.put_line (SUBSTR ('x_msg_data = ' || x_msg_data, 1, 255));

IF x_msg_count > 1

THEN

FOR i IN 1 .. x_msg_count
```

```
LOOP

DBMS_OUTPUT.put_line

(   i

|| '. '

|| SUBSTR

(fnd_msg_pub.get (p_encoded       => fnd_api.g_false),

1,

255

)

);

END LOOP;

END IF;

END;
```

Note:

This routine is used to create a Customer Account. The API creates a record in the HZ_CUST_ACCOUNTS table for party type Person or Organization. Account can be created for an existing party by passing party_id of the party. Alternatively, this routine creates a new party and an account for the party.

Customer profile record in the HZ_CUSTOMER_PROFILES can also be created while calling this routine based on value passed in p_customer_profile_rec. The routine is overloaded for Person and Organization.

Create a Customer Account Site

Use an existing Party Site

```
DECLARE

p_cust_acct_site_rec    hz_cust_account_site_v2pub.cust_acct_site_rec_type;

x_return_status         VARCHAR2 (2000);

x_msg_count              NUMBER;

x_msg_data              VARCHAR2 (2000);
```

```
x_cust_acct_site_id    NUMBER;

BEGIN

p_cust_acct_site_rec.cust_account_id := 3472;

p_cust_acct_site_rec.party_site_id := 1024;

p_cust_acct_site_rec.LANGUAGE := 'US';

p_cust_acct_site_rec.created_by_module := 'TCA-EXAMPLE';

hz_cust_account_site_v2pub.create_cust_acct_site ('T',

p_cust_acct_site_rec,

x_cust_acct_site_id,

x_return_status,

x_msg_count,

x_msg_data

);

DBMS_OUTPUT.put_line (SUBSTR ('x_return_status = ' || x_return_status,

1,

255

)

);

DBMS_OUTPUT.put_line ('x_msg_count = ' || TO_CHAR (x_msg_count));

DBMS_OUTPUT.put_line (SUBSTR ('x_msg_data = ' || x_msg_data, 1, 255));

IF x_msg_count > 1

THEN

FOR i IN 1 .. x_msg_count

LOOP

DBMS_OUTPUT.put_line
```

```
( i
|| '. '
|| SUBSTR
(fnd_msg_pub.get (p_encoded      => fnd_api.g_false),
1,
255
)
);
END LOOP;
END IF;
END;
```

Create Customer Account Site Use Code:

```
DECLARE
p_cust_site_use_rec      hz_cust_account_site_v2pub.cust_site_use_rec_type;
p_customer_profile_rec   hz_customer_profile_v2pub.customer_profile_rec_type;
x_site_use_id            NUMBER;
x_return_status          VARCHAR2 (2000);
x_msg_count              NUMBER;
x_msg_data               VARCHAR2 (2000);
BEGIN
p_cust_site_use_rec.cust_acct_site_id := 3580;
p_cust_site_use_rec.site_use_code := 'INV';
p_cust_site_use_rec.LOCATION := 'TCA';
p_cust_site_use_rec.created_by_module := 'ERPSCHOOLS_DEMO';
hz_cust_account_site_v2pub.create_cust_site_use ('T',
p_cust_site_use_rec,
```

```
p_customer_profile_rec,
'',
'',
x_site_use_id,
x_return_status,
x_msg_count,
x_msg_data
);
DBMS_OUTPUT.put_line (SUBSTR ('x_return_status = ' || x_return_status,
1,
255
)
);
DBMS_OUTPUT.put_line ('x_msg_count = ' || TO_CHAR (x_msg_count));
DBMS_OUTPUT.put_line (SUBSTR ('x_msg_data = ' || x_msg_data, 1, 255));
IF x_msg_count > 1
THEN
FOR i IN 1 .. x_msg_count
LOOP
DBMS_OUTPUT.put_line
(   i
|| '. '
|| SUBSTR
(fnd_msg_pub.get (p_encoded       => fnd_api.g_false),
1,
```

255

)

);

END LOOP;

END IF;

END;

More Customer API's:

Org Contact Role	Hz_party_contact_v2pub.Create_Org_Contact_Role
Relationships	HZ_CUST_ACCOUNT_V2PUB.CREATE_CUST_ACCT_RELATE
Customer Profile	HZ_CUSTOMER_PROFILE_V2PUB. create_customer_profile
Customer Profile Amount	HZ_CUSTOMER_PROFILE_V2PUB. create_cust_profile_amt
Customer Credit Rating	HZ_PARTY_INFO_V2PUB.create_credit_rating
Sales Person	JTF_RS_SALESREPS_PUB.CREATE_SALESREP
Sales reps Territories	JTF_RS_SRP_TERRITORIES_PUB.CREATE_RS_SRP_TERRITORIES
Customer contacts	HZ_CUST_ACCOUNT_ROLE_V2PUB.CREATE_CUST_ACCOUNT_ROLE
Customer Contact Role	HZ_CUST_ACCOUNT_ROLE_V2PUB.create_role_responsibility

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012 FAQ'S

PURCHASING FAQ'S:

Purchasing
What are different types of PO's? [http://erpschools.com/faq/what-are-different-types-of-pos]
What is dollar limit and how can you change it? [http://erpschools.com/faq]
Can any on give me tha detailed tables & theire relationship….in detail….
[http://erpschools.com/faq]
EXPLAIN ABOUT PO FLOW [http://erpschools.com/faq]
After done contact purchase order . we can Create standard order for contract ., in that we mention Contract Po No. in Reff Doc column. In which table and what column that value is stored ? [http://erpschools.com/faq]

What is common Id for Requisition and po ? [http://erpschools.com/faq]
What is backdated Receipt? [http://erpschools.com/faq]
Is it possible to create a backdated receipt with a receipt date falling in closed GL period? [http://erpschools.com/faq]
What is Receipt Date Tolerance? [http://erpschools.com/faq]
How do we set the Receipt Tolerance? [http://erpschools.com/faq]
what are the tables link between PO & Gl technically? [http://erpschools.com/faq]
Can you create PO in iProcurement ? [http://erpschools.com/faq]
Give me some PO tables [http://erpschools.com/faq]
Tell me about PO cycle? [http://erpschools.com/faq]
Explain Approval Heirarchies in PO. [http://erpschools.com/faq]
How does workflow determines whom to send the PO for approval? [http://erpschools.com/faq]
What is Purchase order workflow ? [http://erpschools.com/faq]
What is Planned PO? [http://erpschools.com/faq]
What functions you do from iProcurement? [http://erpschools.com/faq]
What is difference between positional hierarchy and supervisor hierarchy? [http://erpschools.com/faq]
What is the difference between purchasing module and iProcurement module? [http://erpschools.com/faq]
What is Blanket PO? [http://erpschools.com/faq]

ORDER MANAGEMENT FAQ'S :

Order Management
Can there be any step/procedure of Expiry of a Sales Order whether all shipments have been made or not? [http://erpschools.com/faq]
What are the required fields when entering an order in the Order Entry Screen? [http://erpschools.com/faq]
What is the prerequisite to do before customer converion? [http://erpschools.com/faq]
What is Quilifier and Modifier? [http://erpschools.com/faq]
what are the accounts will populate in OM? [http://erpschools.com/faq]
What is the difference between purchase order (PO) and sales order? [http://erpschools.com/faq]
What are the Process Constraints? [http://erpschools.com/faq]
What is a Pick Slip Report? [http://erpschools.com/faq]
At what stage an order cannot be cancelled? [http://erpschools.com/faq]
When the order import program is run it validates and the errors occurred can be seen in? [http://erpschools.com/faq]
What are picking rules? [http://erpschools.com/faq]
What does Back ordered mean in OM? [http://erpschools.com/faq]
What is ONT stands for? [http://erpschools.com/faq]
What is order type? [http://erpschools.com/faq]
How is move order generated? [http://erpschools.com/faq]
Name some tables in shipping/order/move order/inventory? [http://erpschools.com/faq]
What are primary and secondary price lists? [http://erpschools.com/faq]
Explain the Order Cycle? [http://erpschools.com/faq]
What is packing slip? [http://erpschools.com/faq]
What is Document Sequence? [http://erpschools.com/faq]
Where do you find the order status column? [http://erpschools.com/faq]
What is Order Import and What are the Setup’s involved in Order Import?

[http://erpschools.com/faq]
What are validation templates? [http://erpschools.com/faq]
What are different Order Types? [http://erpschools.com/faq]
What are Defaulting Rules? [http://erpschools.com/faq]
What is drop ship in OM? [http://erpschools.com/faq]
What is Advanced Shipping Notice? [http://erpschools.com/faq]
When an order cannot be deleted? [http://erpschools.com/faq]
What is Drop shipment? [http://erpschools.com/faq]
What is pick slip? [http://erpschools.com/faq]
what is link between Purchase Order to Sales Order? [http://erpschools.com/faq/what-is-link-between-purchase-order-to-sales-order]

2ND

Can object group have a block? [http://erpschools.com/faq]
What is the procedure and trigger called when a window is closed? [http://erpschools.com/faq]
What are different Visual Attributes in Forms? [http://erpschools.com/faq]
What are user-exits? [http://erpschools.com/faq]
What is SECURE property? [http://erpschools.com/faq]
What is Data Block? [http://erpschools.com/faq]
How many types of canvases are there? [http://erpschools.com/faq]
Give some Built in Package Names? [http://erpschools.com/faq]
What are the types of triggers and how the sequence of firing in text item? [http://erpschools.com/faq]
What triggers are available in Forms? [http://erpschools.com/faq]
If you have property class attached to an item and you have same trigger written for the item .Which will fire first? [http://erpschools.com/faq]
What are Object Groups? [http://erpschools.com/faq]
Can you call WIN-SDK through user exits? [http://erpschools.com/faq]
What is path setting for DLL? [http://erpschools.com/faq]
What are property classes? Can property classes have trigger? [http://erpschools.com/faq]
What is custom.pll used for? [http://erpschools.com/faq]
What is mouse navigate property of button? [http://erpschools.com/faq]
What are program units in forms? How to do you use them? [http://erpschools.com/faq]
what are key-mode and locking mode properties? level ? [http://erpschools.com/faq]
Can you store pictures in database? How? [http://erpschools.com/faq]
What are record groups? [http://erpschools.com/faq]
Can record groups created at run-time? [http://erpschools.com/faq]
Does user exits supports DLL on MSWINDOWS ? [http://erpschools.com/faq]
How do you attach a library to a form? [http://erpschools.com/faq]
Can you issue DDL in forms? If so how do you? [http://erpschools.com/faq]
How do you build a popup menu? [http://erpschools.com/faq]
How to create a file in forms? [http://erpschools.com/faq]
how to develop new form in oracle apps? [http://erpschools.com/faq]
What is FORMS_MDI_WINDOW? [http://erpschools.com/faq]
What are savepoint mode and cursor mode properties level? [http://erpschools.com/faq]
What is difference between PL/SQL library and object library in Forms? [http://erpschools.com/faq]
Can a button have icon and lable at the same time? [http://erpschools.com/faq]
What is the process/steps for Vendor Conversion? [http://erpschools.com/faq]

What is Invoice Tolerance? [http://erpschools.com/faq]
Explain the set up used for Automatic or Manual Supplier Numbering. [http://erpschools.com/faq]
What is Contract PO? [http://erpschools.com/faq]
What is a Payable Document? [http://erpschools.com/faq]
In which table we can find the vendor number? [http://erpschools.com/faq]
Give the cycle from creating an invoice to transferring it to GL in AP. [http://erpschools.com/faq]
What are the different types of Invoices in Payables? [http://erpschools.com/faq]
You have created a new SOB. How will you attach this SOB to AP? [http://erpschools.com/faq]
can we create invoice without PO in payables? then how? [http://erpschools.com/faq]
In AP the suppliers didn¿t visible in India Creditors Ledger Report Parameter? [http://erpschools.com/faq]
What will accrue in Payables? [http://erpschools.com/faq]
What is a Hold? Explain the types of Hold. [http://erpschools.com/faq]
Which module is the owner of Vendor/Supplier tables? [http://erpschools.com/faq]
What is Payment Terms? [http://erpschools.com/faq]
How many key flexfields are there in Payables? [http://erpschools.com/faq]
What is the Distribution Type while entering the Invoice? [http://erpschools.com/faq]
What are the Prepayment types? [http://erpschools.com/faq]
What is Aging Periods? [http://erpschools.com/faq]
Whats the difference between the “Payables Open Interface Import” Program and the “Payables Invoice Import” program? [http://erpschools.com/faq]
What is prepayment & steps to apply it to an Invoice? [http://erpschools.com/faq]
Can you hold the partial payment if yes then how? [http://erpschools.com/faq]
How you will transfer payables to general ledger? [http://erpschools.com/faq]
What program is used to transfer AP transactions to GL? [http://erpschools.com/faq]
What is use of AP Accounting Periods? [http://erpschools.com/faq]
What are the different interface programs in AP? [http://erpschools.com/faq]
What is Debit Memo & Credit Memo in Payables? [http://erpschools.com/faq]
Name some Flexfields in AR. [http://erpschools.com/faq]
Explain the steps involved in Transfer to GL from AR. [http://erpschools.com/faq]
What is the dbnumber of a particular cusotmer TCA? [http://erpschools.com/faq]
Where can you find the Customer payment terms? [http://erpschools.com/faq]
What is the link between OM and AR? [http://erpschools.com/faq]
What kind of transactions can be created using AutoInvoice? [http://erpschools.com/faq]
What are the different Transaction types in AR? [http://erpschools.com/faq]
Explain the different steps in implementing Autolockbox. [http://erpschools.com/faq]
What are the different Invoice matching types? [http://erpschools.com/faq]
What are the underlying tables and validations required during AutoInvoice Interface? [http://erpschools.com/faq]
how can we adjust the money in AR? [http://erpschools.com/faq]
Tell me about TCA? [http://erpschools.com/faq]
What are different types of PO’s? [http://erpschools.com/faq/what-are-different-types-of-pos]
What is dollar limit and how can you change it? [http://erpschools.com/faq]
Can any on give me tha detailed tables & theire relationship….in detail…. [http://erpschools.com/faq]
EXPLAIN ABOUT PO FLOW [http://erpschools.com/faq]
After done contact purchase order . we can Create standard order for contract ., in that we mention Contract Po No. in Reff Doc column. In which table and what column that

value is stored ? [http://erpschools.com/faq]
What is common Id for Requisition and po ? [http://erpschools.com/faq]
What is backdated Receipt? [http://erpschools.com/faq]
Is it possible to create a backdated receipt with a receipt date falling in closed GL period? [http://erpschools.com/faq]
What is Receipt Date Tolerance? [http://erpschools.com/faq]
How do we set the Receipt Tolerance? [http://erpschools.com/faq]
what are the tables link between PO & GI technically? [http://erpschools.com/faq]
Can you create PO in iProcurement ? [http://erpschools.com/faq]
Give me some PO tables [http://erpschools.com/faq]
Tell me about PO cycle? [http://erpschools.com/faq]
Explain Approval Heirarchies in PO. [http://erpschools.com/faq]
How does workflow determines whom to send the PO for approval? [http://erpschools.com/faq]
What is Purchase order workflow ? [http://erpschools.com/faq]
What is Planned PO? [http://erpschools.com/faq]
What functions you do from iProcurement? [http://erpschools.com/faq]
What is difference between positional hierarchy and supervisor hierarchy? [http://erpschools.com/faq]
What is the difference between purchasing module and iProcurement module? [http://erpschools.com/faq]
What is Blanket PO? [http://erpschools.com/faq]
Can there be any step/procedure of Expiry of a Sales Order whether all shipments have been made or not? [http://erpschools.com/faq]
What are the required fields when entering an order in the Order Entry Screen? [http://erpschools.com/faq]
What is the prerequisite to do before customer converion? [http://erpschools.com/faq]
What is Quilifier and Modifier? [http://erpschools.com/faq]
what are the accounts will populate in OM? [http://erpschools.com/faq]
What is the difference between purchase order (PO) and sales order? [http://erpschools.com/faq]
What are the Process Constraints? [http://erpschools.com/faq]
What is a Pick Slip Report? [http://erpschools.com/faq]
At what stage an order cannot be cancelled? [http://erpschools.com/faq]
When the order import program is run it validates and the errors occurred can be seen in? [http://erpschools.com/faq]
What are picking rules? [http://erpschools.com/faq]
What does Back ordered mean in OM? [http://erpschools.com/faq]
What is ONT stands for? [http://erpschools.com/faq]
What is order type? [http://erpschools.com/faq]
How is move order generated? [http://erpschools.com/faq]
Name some tables in shipping/order/move order/inventory? [http://erpschools.com/faq]
What are primary and secondary price lists? [http://erpschools.com/faq]
Explain the Order Cycle? [http://erpschools.com/faq]
What is packing slip? [http://erpschools.com/faq]
What is Document Sequence? [http://erpschools.com/faq]
Where do you find the order status column? [http://erpschools.com/faq]
What is Order Import and What are the Setup's involved in Order Import? [http://erpschools.com/faq]
What are validation templates? [http://erpschools.com/faq]
What are different Order Types? [http://erpschools.com/faq]

What are Defaulting Rules? [http://erpschools.com/faq]
What is drop ship in OM? [http://erpschools.com/faq]
What is Advanced Shipping Notice? [http://erpschools.com/faq]
When an order cannot be deleted? [http://erpschools.com/faq]
What is Drop shipment? [http://erpschools.com/faq]
What is pick slip? [http://erpschools.com/faq]
what is link between Purchase Order to Sales Order? [http://erpschools.com/faq/what-is-link-between-purchase-order-to-sales-order]
How to retry multiple errored workflow processes? [http://erpschools.com/faq]
What is the access level in workflow used for? [http://erpschools.com/faq]
How do you define start and end functions in workflow? How does they differ from normal functions? [http://erpschools.com/faq]
Give me some workflow tables? [http://erpschools.com/faq]
What is the difference between a function and notification in workflow? [http://erpschools.com/faq]
I have sent two different notifications to two different users and I want to wait till both they are approved to send 3rd notification. How can you achieve it? [http://erpschools.com/faq]
What is item type and item key in workflow? [http://erpschools.com/faq]
How do you use attribute values in workflow messages? [http://erpschools.com/faq]
How do you use lookups in workflow? [http://erpschools.com/faq]
What are roles in workflow and how they are used? [http://erpschools.com/faq]
How do you download or upload a workflow from a server? [http://erpschools.com/faq]
What are steps to customize the workflow? [http://erpschools.com/faq]
What functions can you perform from workflow administrator responsibility? [http://erpschools.com/faq]
To send an email to the user workflow notification is the only way or is there any other ways to send it? [http://erpschools.com/faq]
Give me some workflow standard procedures? [http://erpschools.com/faq]
How can you run/start/kickoff workflow? [http://erpschools.com/faq]
What is wf_engine package used for? [http://erpschools.com/faq]
How many processes can each workflow contain? [http://erpschools.com/faq]
What is Runnable option in workflow? At what level it exists? [http://erpschools.com/faq]
What are different types of attributes in workflow? [http://erpschools.com/faq]
How do you reassign a notification? [http://erpschools.com/faq]
What is process in workflow? [http://erpschools.com/faq]
How can you send direct oracle form link through workflow notifications? [http://erpschools.com/faq]
How can you send a notification to multiple users? Can you change the list dynamically? [http://erpschools.com/faq]
Can you send html code in workflow notification? [http://erpschools.com/faq]
I have sent two different notifications to two different users and I want to wait till atleast one is approved to send 3rd notification. How can you achieve it? [http://erpschools.com/faq]
How can you grey out/ highlight/hide some records based on conditions in a report? [http://erpschools.com/faq]
What is the diff. between normal report and XML report? give me atleast 4 differences? [http://erpschools.com/faq]
What is pick slip report and customizations of the report [http://erpschools.com/faq]
What is sales order DFF [http://erpschools.com/faq]
waht is Data conversion? [http://erpschools.com/faq]
How do you call a concurrent program or another report from a report? [http://erpschools.com/faq]

How do you mail the output of a report? [http://erpschools.com/faq]
Where in reports do you set the context information (like org_id)? [http://erpschools.com/faq]
What is Anchor in reports? [http://erpschools.com/faq]
How do you write the report output to Excel file or text file? [http://erpschools.com/faq]
How do you resolve the following layout issues in reports? [http://erpschools.com/faq]
Give an example of the implementation of “between pages trigger” in reports. [http://erpschools.com/faq]
The total printed at the bottom of first page has to be carried to the top of the next page. How do u do this technically? [http://erpschools.com/faq]
What is the difference between Conditional Formatting and format trigger? [http://erpschools.com/faq]
How do you display only one record on each page in a report? [http://erpschools.com/faq]
How do you print barcode in the reports? [http://erpschools.com/faq]
What are the different sections in the Layout? [http://erpschools.com/faq]
What is SRW package and some procedures in SRW? [http://erpschools.com/faq]
What are bind parameter and lexical parameter used for? [http://erpschools.com/faq]
What are user exits in reports and name a few? [http://erpschools.com/faq]
Name the report triggers. [http://erpschools.com/faq]
How many minimum fields are require to develop the Matrix, Matrix with Group Reports? [http://erpschools.com/faq]
What is the mandatory parameter in mutli org report? [http://erpschools.com/faq]
If we have two data models then how many frames we need to design? [http://erpschools.com/faq]
What is the difference between “AFTER PARAMETER FORM” trigger and “BEFORE REPORT” trigger? [http://erpschools.com/faq]
How do you resolve the following layout issues in reports? [http://erpschools.com/faq]
What are the mandatory parameters given in Reports [http://erpschools.com/faq]
Report is not displaying data present in database, what could be the reason? [http://erpschools.com/faq]
Is it possible to change the margins for oracle report? [http://erpschools.com/faq]
What is Report Busting? [http://erpschools.com/faq]
what is report bursting? [http://erpschools.com/faq]
How can we display header information for every page while developing XML report? [http://erpschools.com/faq]
while creating XML reoprt,which one is mandatory either Data Definition or Template creation? and why? [http://erpschools.com/faq]
why we use incompatability option in concurant program window? Explian with an example in real time. [http://erpschools.com/faq]
What do you know about Trace and Tuning? [http://erpschools.com/faq]
How do you set the profile option in PL/SQL Procedure? [http://erpschools.com/faq]
Have you ever used TABLE datatype and what is it used for? [http://erpschools.com/faq]
How do you set profile options from PL/SQL procedure? [http://erpschools.com/faq]
what is External table? [http://erpschools.com/faq]
Can two users update the same row at the same time? if so how? [http://erpschools.com/faq]
How do you retrieve the last N records from a table? [http://erpschools.com/faq]
How do you eliminate duplicate rows from a table? [http://erpschools.com/faq]
How do you declare user defined Exception? [http://erpschools.com/faq]
What is a mutating table? [http://erpschools.com/faq]
“UPDATE; CREATE TABLE ; ROLL BACK;” To which save point will the changes be Rolled Back? [http://erpschools.com/faq/update-create-table-roll-back-to-which-save-point-will-the-changes-be-rolled-back]

What is tkprof and the syntax? [http://erpschools.com/faq]
How do you set the profile option from a PL/SQL procedure? [http://erpschools.com/faq]
What is the SQL statement used to display the text of a procedure stored in database? [http://erpschools.com/faq]
How do you retrieve the last N records from a table? [http://erpschools.com/faq]
Name the different database triggers? [http://erpschools.com/faq]
What is the difference between TRUNCATE and DELETE? [http://erpschools.com/faq]
Can you use COMMIT in a trigger? [http://erpschools.com/faq]
Can triggers be used on views? If so How? [http://erpschools.com/faq]
What is Ref Cursor? [http://erpschools.com/faq]
Can you call a sequence in SQL Loader? [http://erpschools.com/faq]
What are the three files that are generated when you load data using SQL Loader? [http://erpschools.com/faq]
What are the types of Exceptions? [http://erpschools.com/faq]
What are the differences between Function and Procedure? [http://erpschools.com/faq]
What is dynamic SQL? [http://erpschools.com/faq]
How do you submit a concurrent program from PL/SQL Procedure? [http://erpschools.com/faq]
What is the difference between View and Materialized view? [http://erpschools.com/faq]
What is RAISE_APPLICATION_ERROR used for? [http://erpschools.com/faq]
Give the structure of the trigger? [http://erpschools.com/faq]
What is an autonomous transaction ? [http://erpschools.com/faq]
What are the different cursors available in PL/SQL ? [http://erpschools.com/faq]
What is difference between TRUNCATE & DELETE [http://erpschools.com/faq]
What is ROWID? [http://erpschools.com/faq]
What are the advantages of VIEW? [http://erpschools.com/faq]
What are SQLCODE and SQLERRM and why are they important for PL/SQL developers? [http://erpschools.com/faq]
What is tkprof and how is it used? [http://erpschools.com/faq]
What is global temporary table? [http://erpschools.com/faq]
how to find out duplicate records from the table? [http://erpschools.com/faq]
What is conditional filtering at database level? (Hint: New feature released in 10g) [http://erpschools.com/faq]
How to develop a PO print report using XML publisher? [http://erpschools.com/faq]
What is the difference between org_id and organization_id? [http://erpschools.com/faq]
In OAFramework, can we have two AM's for single page? [http://erpschools.com/faq]
what is API and how can we find the API's in oracle apps? [http://erpschools.com/faq]
What is the initial setup before using utl_file type? [http://erpschools.com/faq]
What are types of version control tools used in Oracle apps? [http://erpschools.com/faq]
Even though we are loading segment values into tables,Why should we load KFF data in combined format? [http://erpschools.com/faq]
what are the tables of DFF flex fields? [http://erpschools.com/faq]
What are the steps to be follow before implementation for a customer with relate to documents? [http://erpschools.com/faq]
Can we use discard file as data file again (after removing errors)? [http://erpschools.com/faq]
In which mode you will transfer files (default or ascii)? [http://erpschools.com/faq]
Can we use date function in SQL loader? [http://erpschools.com/faq]
What are global variables ? How they can be used ? [http://erpschools.com/faq]
What are custom events in apps and how to you enable/disable it? [http://erpschools.com/faq]

How do you enable trace/debug in APPS? [http://erpschools.com/faq]
What is the difference between Key flex field and Descriptive flex field? [http://erpschools.com/faq]
What are the steps to register Concurrent Program? [http://erpschools.com/faq]
Tell me about Multi Org? [http://erpschools.com/faq]
What is difference between concurrent program and concurrent request? [http://erpschools.com/faq]
What is ALERT? [http://erpschools.com/faq]
What is profile option in APPS? [http://erpschools.com/faq]
What is diagnostics in apps? How do you enable/disable it? [http://erpschools.com/faq]
Explain the Order to Cash cycle and the underlying tables. [http://erpschools.com/faq]
What is FNDLOADER used for its syntax? [http://erpschools.com/faq]
Explain the Procure to Pay cycle and the underlying tables. [http://erpschools.com/faq]
If the business entity has 5 Operating Units,How many times do you have to implement Apps? [http://erpschools.com/faq]
How do you run diagnostics for a particular module to send it to Oracle support? [http://erpschools.com/faq]
What are folders? How do you modify them? [http://erpschools.com/faq]
what is asp.net [http://erpschools.com/faq]
What is Forms Personalization? Where it is used? What For? [http://erpschools.com/faq]
What are different triggers that can be used in personalization? [http://erpschools.com/faq]
What is the difference between having personalization at function level rather than form level? [http://erpschools.com/faq]
Can you use global variables in forms personalization? If so How? [http://erpschools.com/faq]
Can you hide a text field using personalization? How? [http://erpschools.com/faq]
How do you make a field mandatory or not mandatory using Forms Personalization? [http://erpschools.com/faq]
Can you transfer the data from one form to another form using personalization? [http://erpschools.com/faq]
How do you move personalization from one instance/database to other? [http://erpschools.com/faq]
What is personalization and what features can be achieved through personalization? [http://erpschools.com/faq]
Can you default a value in a text filed through personalization? How? [http://erpschools.com/faq]
How can you restrict the access (to oracle apps) to A GROUP OF users using personalization? [http://erpschools.com/faq]
At what level you can restrict personalization code? [http://erpschools.com/faq]
What can be implemented through Forms Personalization? [http://erpschools.com/faq]
How to do implement ZOOM Functionality using personalization? [http://erpschools.com/faq]
What are the advantages/disadvantages of Forms Personalization when compared to CUSTOM.pll? [http://erpschools.com/faq]
When you display a error message at WHEN-VALIDATE Trigger will the data will be saved into database? [http://erpschools.com/faq]
Test FAQ [http://erpschools.com/faq/test-faq]
What is the best approach to load 50,000 rows into a table? [http://erpschools.com/faq/what-is-the-best-approach-to-load-50000-rows-into-a-table]
How is Case statement different from Decode statement? [http://erpschools.com/faq/how-is-case-statement-different-from-decode-statement]

GENERAL FAQ'S :

General

How to develop a PO print report using XML publisher? [http://erpschools.com/faq]
What is the difference between org_id and organization_id? [http://erpschools.com/faq]
In OAFramework, can we have two AM's for single page? [http://erpschools.com/faq]
what is API and how can we find the API's in oracle apps? [http://erpschools.com/faq]
What is the initial setup before using utl_file type? [http://erpschools.com/faq]
What are types of version control tools used in Oracle apps? [http://erpschools.com/faq]
Even though we are loading segment values into tables,Why should we load KFF data in combined format? [http://erpschools.com/faq]
what are the tables of DFF flex fields? [http://erpschools.com/faq]
What are the steps to be follow before implementation for a customer with relate to documents? [http://erpschools.com/faq]
Can we use discard file as data file again (after removing errors)? [http://erpschools.com/faq]
In which mode you will transfer files (default or ascii)? [http://erpschools.com/faq]
Can we use date function in SQL loader? [http://erpschools.com/faq]
What are global variables ? How they can be used ? [http://erpschools.com/faq]
What are custom events in apps and how to you enable/disable it? [http://erpschools.com/faq]
How do you enable trace/debug in APPS? [http://erpschools.com/faq]
What is the difference between Key flex field and Descriptive flex field? [http://erpschools.com/faq]
What are the steps to register Concurrent Program? [http://erpschools.com/faq]
Tell me about Multi Org? [http://erpschools.com/faq]
What is difference between concurrent program and concurrent request? [http://erpschools.com/faq]
What is ALERT? [http://erpschools.com/faq]
What is profile option in APPS? [http://erpschools.com/faq]
What is diagnostics in apps? How do you enable/disable it? [http://erpschools.com/faq]
Explain the Order to Cash cycle and the underlying tables. [http://erpschools.com/faq]
What is FNDLOADER used for its syntax? [http://erpschools.com/faq]
Explain the Procure to Pay cycle and the underlying tables. [http://erpschools.com/faq]
If the business entity has 5 Operating Units,How many times do you have to implement Apps? [http://erpschools.com/faq]
How do you run diagnostics for a particular module to send it to Oracle support? [http://erpschools.com/faq]
What are folders? How do you modify them? [http://erpschools.com/faq]
what is asp.net [http://erpschools.com/faq]
What is the best approach to load 50,000 rows into a table? [http://erpschools.com/faq/what-is-the-best-approach-to-load-50000-rows-into-a-table]
Waht is the best approach to write 50,000 rows to a data file? [http://erpschools.com/faq/waht-is-the-best-approach-to-write-50000-rows-to-a-data-file]

PAYABLES FAQ'S :

Payables

What is the process/steps for Vendor Conversion? [http://erpschools.com/faq]
What is Invoice Tolerance? [http://erpschools.com/faq]

Explain the set up used for Automatic or Manual Supplier Numbering. [http://erpschools.com/faq]
What is Contract PO? [http://erpschools.com/faq]
What is a Payable Document? [http://erpschools.com/faq]
In which table we can find the vendor number? [http://erpschools.com/faq]
Give the cycle from creating an invoice to transferring it to GL in AP. [http://erpschools.com/faq]
What are the different types of Invoices in Payables? [http://erpschools.com/faq]
You have created a new SOB. How will you attach this SOB to AP? [http://erpschools.com/faq]
can we create invoice without PO in payables? then how? [http://erpschools.com/faq]
In AP the suppliers didn¿t visible in India Creditors Ledger Report Parameter? [http://erpschools.com/faq]
What will accrue in Payables? [http://erpschools.com/faq]
What is a Hold? Explain the types of Hold. [http://erpschools.com/faq]
Which module is the owner of Vendor/Supplier tables? [http://erpschools.com/faq]
What is Payment Terms? [http://erpschools.com/faq]
How many key flexfields are there in Payables? [http://erpschools.com/faq]
What is the Distribution Type while entering the Invoice? [http://erpschools.com/faq]
What are the Prepayment types? [http://erpschools.com/faq]
What is Aging Periods? [http://erpschools.com/faq]
Whats the difference between the “Payables Open Interface Import” Program and the “Payables Invoice Import” program? [http://erpschools.com/faq]
What is prepayment & steps to apply it to an Invoice? [http://erpschools.com/faq]
Can you hold the partial payment if yes then how? [http://erpschools.com/faq]
How you will transfer payables to general ledger? [http://erpschools.com/faq]
What program is used to transfer AP transactions to GL? [http://erpschools.com/faq]
What is use of AP Accounting Periods? [http://erpschools.com/faq]
What are the different interface programs in AP? [http://erpschools.com/faq]
What is Debit Memo & Credit Memo in Payables? [http://erpschools.com/faq]

RECEIVABLES FAQ'S:

Receivables
Name some Flexfields in AR. [http://erpschools.com/faq]
Explain the steps involved in Transfer to GL from AR. [http://erpschools.com/faq]
What is the dbnumber of a particular cusotmer TCA? [http://erpschools.com/faq]
Where can you find the Customer payment terms? [http://erpschools.com/faq]
What is the link between OM and AR? [http://erpschools.com/faq]
What kind of transactions can be created using AutoInvoice? [http://erpschools.com/faq]
What are the different Transaction types in AR? [http://erpschools.com/faq]
Explain the different steps in implementing Autolockbox. [http://erpschools.com/faq]
What are the different Invoice matching types? [http://erpschools.com/faq]
What are the underlying tables and validations required during AutoInvoice Interface? [http://erpschools.com/faq]
how can we adjust the money in AR? [http://erpschools.com/faq]
Tell me about TCA? [http://erpschools.com/faq]

REPORTS FAQ'S:

Reports

How can you grey out/ highlight/hide some records based on conditions in a report? [http://erpschools.com/faq]
What is the diff. between normal report and XML report? give me atleast 4 differences? [http://erpschools.com/faq]
What is pick slip report and customizations of the report [http://erpschools.com/faq]
What is sales order DFF [http://erpschools.com/faq]
waht is Data conversion? [http://erpschools.com/faq]
How do you call a concurrent program or another report from a report? [http://erpschools.com/faq]
How do you mail the output of a report? [http://erpschools.com/faq]
Where in reports do you set the context information (like org_id)? [http://erpschools.com/faq]
What is Anchor in reports? [http://erpschools.com/faq]
How do you write the report output to Excel file or text file? [http://erpschools.com/faq]
How do you resolve the following layout issues in reports? [http://erpschools.com/faq]
Give an example of the implementation of “between pages trigger” in reports. [http://erpschools.com/faq]
The total printed at the bottom of first page has to be carried to the top of the next page. How do u do this technically? [http://erpschools.com/faq]
What is the difference between Conditional Formatting and format trigger? [http://erpschools.com/faq]
How do you display only one record on each page in a report? [http://erpschools.com/faq]
How do you print barcode in the reports? [http://erpschools.com/faq]
What are the different sections in the Layout? [http://erpschools.com/faq]
What is SRW package and some procedures in SRW? [http://erpschools.com/faq]
What are bind parameter and lexical parameter used for? [http://erpschools.com/faq]
What are user exits in reports and name a few? [http://erpschools.com/faq]
Name the report triggers. [http://erpschools.com/faq]
How many minimum fields are require to develop the Matrix, Matrix with Group Reports? [http://erpschools.com/faq]
What is the mandatory parameter in mutli org report? [http://erpschools.com/faq]
If we have two data models then how many frames we need to design? [http://erpschools.com/faq]
What is the difference between “AFTER PARAMETER FORM” trigger and “BEFORE REPORT” trigger? [http://erpschools.com/faq]
How do you resolve the following layout issues in reports? [http://erpschools.com/faq]
What are the mandatory parameters given in Reports [http://erpschools.com/faq]
Report is not displaying data present in database, what could be the reason? [http://erpschools.com/faq]
Is it possible to change the margins for oracle report? [http://erpschools.com/faq]
What is Report Busting? [http://erpschools.com/faq]
what is report bursting? [http://erpschools.com/faq]
How can we display header information for every page while developing XML report? [http://erpschools.com/faq]
while creating XML reoprt,which one is mandatory either Data Definition or Template creation? and why? [http://erpschools.com/faq]
why we use incompatability option in concurant program window? Explian with an example in real time. [http://erpschools.com/faq]

Posted 29th August 2012 by Krishnareddy

0 Add a comment

29th August 2012

URLs for APPS

URLs for APPS
OA Framework
==========
http://rajakumareddy.blogspot.com
http://oracle.anilpassi.com/oa-framework.html
http://oracle.anilpassi.com/xmlimporter-in-oracle-applications-framework.html
http://apps2fusion.com/at/61-kv/317-oa-framework-page-without-login-guest-no-security
http://www.orafaq.com/wiki/JDeveloper
http://oracle.anilpassi.com/jdr-utils.html
http://oracle.anilpassi.com/oa-framework-tutorials-training.html
http://www.dulcian.com/papers/MAOP-AOTC/2002/Don'tBeAMuggle.htm
http://www.dulcian.com/papers/OracleOpenWorld/2002/What%20You%20Need%20to%20Know%20Before%20Building%20Applications%20with%20JDeveloper%209i.htm
http://apps2fusion.com/apps/oa-framework/14-fwk/151-oa-framework-screen-extension-by-embedding-a-custom-page
http://oracle.anilpassi.com/oa-framework-tutorial-01-2.html
http://oracle.anilpassi.com/oa-framework-table-based-screen-2.html
http://www.dbforums.com/oracle/1630066-jdeveloper-resolving-errors-encountered.html
http://appstechnical.blogspot.com/2007/01/oa-framework-tutorials.html
http://oraclearea51.com/oracle-technical-articles/oa-framework/oa-framework-beginners-guide/322-exporting-oa-page-definitions.html
http://oraclearea51.com/oracle-technical-articles/oa-framework/oa-framework-beginners-guide.html
http://www.oracle.com/technology/products/jdev/tips/muench/designpatterns/index.html
http://www.oracle.com/technology/pub/articles/vohra-jdev-xmlpub.html
http://mukx.blogspot.com/2010/01/upload-file-to-application-server-using.html
http://blog.cholaconsulting.com/search/label/OA%20Framework
http://sabersurge.com/oracleappsdev/index.php?option=com_content&view=article&id=54%3Afile-upload-to-database-server-oa-framework&catid=34%3Aoa-framework&Itemid=1
http://www.tier1inc.com/blog_comments.php?pid=12-----Comparing OA Framework with Forms 6i
http://oracle-applications-rama.blogspot.com/2009/01/how-to-search-apps-documents-in-google.html

http://www.confluentminds.com/Trainings/SCM/---------->scm

Oracle Forms Web
Upload, edit and download files from/to the database with the
Webutil library
==========
http://sheikyerbouti.developpez.com/webutil-docs/Webutil_store_edit_docs.htm

Check Java Version
==========
http://java.com/en/download/installed.jsp?jre_version=1.6.0_07&vendor=Sun+Microsystems+Inc.&os=Windows+2000&os_version=5.0

Linux Commands
==========
http://www.ss64.com/bash/
http://teachmeoracle.com/unixa.html
http://www.nixblog.org/post/2008/03/14/UNIX-ID-ORACLE-SESSION
http://www.unix.com/shell-programming-scripting/84635-unix-script-detect-new-file-entry-directory.html

Register Shell Scripts As Concurrent Program
==========
http://www.notesbit.com/index.php/scripts-oracle/oracle-applications-steps-to-register-shell-script-as-a-concurrent-program/

UTL_FILE_DIR
==========
http://oracleappstechnology.blogspot.com/2008/03/minimize-usage-of-utlfiledir.html

Oracle Applications
==========
http://becomeappsdba.blogspot.com/
http://www.ddbcinc.com/askDDBC/
http://beginapps.blogspot.com/2007_09_01_archive.html
http://knoworacle.wordpress.com/tag/apps-table/
http://appsdba4u.blogspot.com/2007/08/oracle-apps-dba-interview-questions.html
http://cnubandi.blogspot.com/
http://idbasolutions.com/category/papers/3480000-115102/3480000-final-run/
http://becomeappsdba.blogspot.com/2006/08/startup-shutdown-apps-services.html
http://oracleappss.blogspot.com/2008/07/supplier-additional-information.html
http://erp-consultancy.blogspot.com/2008/03/tds-flow-in-accounts-payable-oracle.html
http://etrm.oracle.com/license/MustLogin.html
http://oracle-magic.blogspot.com/2007/06/concurrent-request-and-its-database-sid.html
http://oracle-applications-rama.blogspot.com/
http://www.oracleappshub.com/11i/purchase-order-approval-hierarchy/
http://confluentminds.com/Trainings/SCM/Topic1.1_Ch1_Part4.html
http://forums.oracle.com/forums/thread.jspa?threadID=457983
http://download.oracle.com/docs/cd/A60725_05/html/comnls/us/alr/multins.htm
http://oracle.ittoolbox.com/groups/technical-functional/oracle-apps-l/
http://www.aboutoracleapps.com/2007/07/oracle-purchasing-po-faq.html

http://forums.oracle.com/forums/thread.jspa?threadID=664806&tstart=0
http://oracle.anilpassi.com/technical-interview-questions-in-oracle-apps.html
http://www.oracleappshub.com/accounts-receivable/ar-back-to-basictechnical-foundation/
http://www.oracleappshub.com/aol/setting-default-to-excel-for-exported-file-from-file-export/
http://asoracle.blogspot.com/2007/11/key-tables-financials.html
http://oracle.ittoolbox.com/groups/technical-functional/oracle-apps-l/switch-responsibility-icon-on-toolbar-283079
http://oracle.anilpassi.com/forms-personalizations.html
http://www.erpschools.com/Oracle_Apps_Form_Customization.asp
http://www.erpschools.com/
http://realworldoracleapps.blogspot.com/search/label/Welcome%20Note
http://oracleappsrealworld.blogspot.com/
http://mastanreddy.blogspot.com/
http://www.scribd.com/doc/3256741/Oracle-Applications-Developers-Guide
http://garethroberts.blogspot.com/2007/10/excel-file-output-from-oracle.html
http://garethroberts.blogspot.com/2008/01/changing-default-layout-format-from-pdf.html
http://erpcrmapps.blogspot.com/2008/01/using-keyboard-shortcuts.html
http://www.hrmsaces.co.uk/ubbthreads.php/forums/8/1
http://bbs.erp100.com/archiver/tid-36506.html
http://edelivery.oracle.com/EPD/WelcomePage/get_form?ARU_LANG=US
http://oraclepitstop.wordpress.com/2007/04/18/versions-of-components-in-oracle-apps/
http://www.aboutoracleapps.com/2007/08/oracle-general-ledger.html
http://oracle.anilpassi.com/oracle-payroll-tables.html
http://confluentminds.com/Trainings/SCM/Topic2.3_Ch2_Part2.html
http://aksenthil.blogspot.com/
http://knoworacle.wordpress.com/category/oracle-applications-technical/oracle-fixed-assets-technical/
https://metalink.oracle.com/
http://solutionbeacon.blogspot.com/2007/07/simple-tutorial-for-publishing-fsg.html
http://sbllc3.solutionbeacon.net/pls/a159vis2/fndgfm/fnd_help.get/US@PSA_US/fnd/@e_cp
http://apps2fusion.com/apps/oracle-hrms/oracle-hr/hrms
http://oracle.anilpassi.com/creating-customer-address-in-tca-step-by-step.html
http://www.aboutoracleapps.com/2007/07/oracle-apps-manufacturingaolforms.html
http://forums.oracle.com/forums/thread.jspa?threadID=590547
(All In one Blog) http://www.sap-img.com/oracle-database/oracle-application-hints-and-tips.htm
http://www.dba-oracle.com/art_dbazine_conc_mgr.htm
http://dineshnair.wordpress.com/
http://oracle.anilpassi.com/basic-concepts-list-of-useful-oracle-apps-articles-2.html
http://chandramatta.blogspot.com/search/label/Concurrent%20Programs------Very Useful Blog for all Topics

Interfaces
==========
(AR Customer Interface Info.)
http://sbllc3.solutionbeacon.net/pls/a159vis2/fndgfm/fnd_help.get/US@PSA_US/ar/@n_tbl_val@PSA_US
http://download-west.oracle.com/docs/cd/A60725_05/html/comnls/us/ar/cusimprt.htm#n_cust_import

http://download.oracle.com/docs/cd/A60725_05/html/comnls/us/ar/autoin05.htm#n_autoval
http://www.erpschools.com/apps/oracle-applications/Articles/General/Interfaces-and-Conversions-in-Oracle-Applications/index.aspx
(Payables)
http://sbllc3.solutionbeacon.net/pls/a159vis2/fndgfm/fnd_help.get/US@PSA_US/AP/@r_openaud@PSA_US
http://irep11i10.oracle.com/OA_HTML/OA.jsp?page=/oracle/apps/fnd/rep/webui/OpenInterfaceDetailsPG&CategoryValue=F&_ti=199580375&retainAM=Y&addBreadCrumb=Y&oapc=7&oas=biKg9_cvMvUQM4gNQIA6ug..
(Customer Interface Error Code Meaning)
http://sbllc3.solutionbeacon.net/pls/a159vis2/fndgfm/fnd_help.get/US@PSA_US/ar/@custerrs@PSA_US

FND LOAD
==========
http://oracle.anilpassi.com/fndload-for-oracle-web-adi.html

R12 Vision Instance Login
==========
http://vis1200.solutionbeacon.net/OA_HTML/RF.jsp?function_id=1032924&resp_id=-1&resp_appl_id=-1&security_group_id=0&lang_code=US¶ms=gEzpj7eR1-rfQLgP8Ol8DQ3u3xBOeCmcdxx.JgrY94g&oas=-Q6TtBxoQEySwZJoZFr0Fw..

General
==========
http://www.httpsurf.com/
http://www.webindia123.com/history/index.html
http://download.oracle.com/docs/cd/A57673_01/DOC/api/doc/PC_22/ch02.htm
http://www.wapuser.co.cc/2009/12/how-to-hack-others-yahoo-password.html
http://www.osun.org/
http://oracle-applications-rama.blogspot.com/2009/01/how-to-search-apps-documents-in-google.html

Oracle Apps Data Structure
==========
http://www.scribd.com/doc/404946/Oracle-Apps-Data-Structure

Oracle Apps Scripts
==========
http://www.erpschools.com/Oracle_Apps_Scripts.asp
http://www.erpschools.com/
http://scripts4oracle.blogspot.com/

Metalink Notes
==========
http://teachmeoracle.com/metalink.html
http://www.oracle.com/technology/tech/globalization/index.html

Orcale DB Concepts
==========
http://www.adp-gmbh.ch/ora/concepts/

Autoconfig
==========
http://becomeappsdba.blogspot.com/2006/10/autoconfig-in-apps-template-files.html
http://onlineappsdba.com/index.php/2008/01/28/autoconfig-in-oracle-apps-11i-r12-12i/

TroubleShooting Concurrent Managers
==========
http://becomeappsdba.blogspot.com/2006/10/troubleshooting-concurrent-managers.html

NLS Language Parameters
==========
http://it.toolbox.com/wiki/index.php/ORA-12700#NLS_Parameters
http://www.oracle.com/technology/tech/globalization/htdocs/nls_lang%20faq.htm#_Toc110410543

Trace Concurrent Request and Generate TKPROF
==========
http://knoworacle.wordpress.com/2008/06/27/tkprof-trace-a-program/

How to find Processes running in Sunsolaris Operating System
==========
http://www.unix.com/sun-solaris/25208-how-find-number-processes.html

Character Set
==========
http://download-uk.oracle.com/docs/cd/B19306_01/server.102/b14225/ch11charsetmig.htm
http://download.oracle.com/docs/cd/B10500_01/server.920/a96529/ch2.htm#745
http://download.oracle.com/docs/cd/B28359_01/server.111/b28298/applocaledata.htm#i634428
http://www.oracledba.co.uk/tips/character_set.htm

ASCII Characters
==========
http://www.alcyone.com/max/reference/compsci/ascii.html
http://www.oracle.com/technology/obe/obe9ir2/obe-nls/localbld/localbld.htm

Oracle SMS Getway Guide
==========
http://www3.ozekisms.com/index.php?ow_page_number=166

Hindustan Times Corporate News
==========
http://www.hindustantimes.com/ListingPage/ListingPage.aspx?Category=Chunk-HT-UI-BusinessSectionPage-Corporate&SectionName=BusinessSectionPage&BC=CorporateNews

MS Word Mail Merge Tutorial
==========
http://personal-computer-tutor.com/mailmerge.htm
http://www.frontpage2002.com/frontpage_2003_tutorial_guide.htm

How to start perticular Concurrent Manager
==========
http://teachmeoracle.com/forum/viewtopic.php?t=4320

Company Profile & Overview
==========
http://info.shine.com/company/Infosys-Technologies-Ltd/102.aspx

Letter Generation using WEB-ADI
==========
http://apps2fusion.com/at/38-ss/351-generate-recruitment-letters-web-adi

Oracle HRMS Fast Farmula Tutorial
==========
http://www.aboutoracleapps.com/2008/11/oracle-hrms-fast-formula-tutorial.html
http://oracle.anilpassi.com/index.php?Itemid=4&id=117&option=com_content&show=1&task=view
http://www.erpschools.com/Apps/oracle-applications/articles/hrms/fast-formulas-in-hrms/index.aspx
http://www.aboutoracleapps.com/2009/01/how-to-generate-develop-or-edit.html

Oracle Workflow
========
http://oracleappstechnology.blogspot.com/2008/02/workflow-mails-not-moving-after-fresh.html
http://forums.oracle.com/forums/thread.jspa?messageID=2327979
http://apps2fusion.com/at/gt/tc/328-workflow-mailer-debugging-script-for-debugging-emails-issues
http://onlineappsdba.com/index.php/2008/07/16/oracle-workflow-notification-mailer-outbound-processing/
http://oracleebusinesssuite.wordpress.com/2008/10/18/debugging-the-approval-workflow-for-purchase-order-or-purchase-requisition/
http://oracle.anilpassi.com/workflows-business-events-training-lesson-4.html
http://oracleappstechnology.blogspot.com/2008/05/disable-retention-on-workflow-queues.html
http://www.freelists.org/post/ora-apps-dba/EXPIRED-messages-in-the-WF-NOTIFICATION-OUT-queue-THE-SOLUTION
https://csslxa03.hkcss.org.hk:16298/OA_HTML/oam/helpdoc/oam_wfoam_notificationmailerg.htm
http://arunrathod.blogspot.com/2008/08/workflow-wfload-through-unix.html
http://oracle.anilpassi.com/apps-technology/2.html
http://www.erpschools.com/Oracle_Applications_Workflow_Launching.asp
http://www.erpschools.com/Oracle_Applications_Workflow_Tutorial.asp
http://oracleebusinesssuite.wordpress.com/2008/10/18/debugging-the-approval-workflow-for-purchase-order-or-purchase-requisition/

AME
======
http://greenoracleapps.blogspot.com/

Make Packages Valid
====================
http://idbasolutions.com/3480000-115102-first-run/

DB Upgrade Context File Creation
========================
http://forums.oracle.com/forums/thread.jspa?
threadID=672921&tstart=0&messageID=2612978

Oracle PL/SQL SMS
==========================
http://forums.oracle.com/forums/thread.jspa?messageID=1827704
http://www.dbasupport.com/forums/archive/index.php/t-24763.html
http://asktom.oracle.com/pls/asktom/f?p=100:11:0::::P11_QUESTION_ID:839412906735
http://www.erpschools.com/Apps/oracle-applications/Interview-Questions/Tools/PL-SQL/default.aspx

Oracle Apps DBA
============================
http://www.teachmeoracle.com/imptables.html
http://www.dba-oracle.com/art_tr_dyn_sga.htm
http://download.oracle.com/docs/cd/A60725_05/html/comnls/us/alr/multins.htm
http://blogs.oracle.com/gverma/2007/11/accrue_economically_dont_recal.html
http://becomeappsdba.blogspot.com/2007/01/change-oracle-apps-password-fndcpass.html
http://oracleappstechnology.blogspot.com/2007/07/query-to-find-out-your-atgpfh-rollup.html
http://onlineappsdba.blogspot.com/2008/06/personalizing-login-page.html
http://www.oracleutilities.com/
http://www.shutdownabort.com/
http://www.dbatoolz.com/sql-scripts
http://krackoracle.blogspot.com/2009_02_01_archive.html

Insert BOB Object (Image) in Database and Display In Report 6i
==
http://sureshvaishya.blogspot.com/2009/08/insert-blob-image-file-into-database.html
http://oracleappsexperience.blogspot.com/

Tunning Oracle Reports
=========================
http://download.oracle.com/docs/cd/B10464_05/bi.904/b13673/pbr_tune.htm#i1007365

FSG Reports
======================
http://www.oracleappshub.com/finance/fsg-what-is-it/

XML / BI Publisher
==========
http://www.oracleappshub.com/xml-publisher/migration-of-xmlpub-templates-and-data-definitions-across-instances/#more-1084
http://blogs.oracle.com/xmlpublisher/2007/05/howto_java_concurrent_programs.html
http://blogs.oracle.com/xmlpublisher/templates/templates_rtf/
http://asun.ifmo.ru/docs/XMLP/help/en_US/htmfiles/B25951_01/T421739T421827.htm

apps ----------modules=----------

http://oracleappss.blogspot.in/2008/07/wip-job-creation.html

http://www.confluentminds.com/Trainings/SCM/---------->scm

http://www.appsbi.com/category/erp

http://www.oracleappshub.com/account-payable/payables-transfer-to-gl/

http://oraclemaniac.com/2012/04/08/enable-audit-trails-for-oracle-apps-tables/

http://www.appsbi.com/oracle-accounts-payables-key-tables-1

http://docs.oracle.com/cd/A60725_05/html/comnls/us/po/index.html

http://oraclemaniac.com/category/tech-stack/plsql/

http://oracle.ittoolbox.com/groups/technical-functional/oracle-apps-l/purchase-requisition-po-report-443015

http://webtopicture.com/oracle/oracle-applications-dba-joyjeet-banerjee-pdf.html

http://pavandba.com/

http://psoug.org/browse.htm?cid=4

http://psoug.org/definition/SELECT.htm

http://psoug.org/reference/library.html

http://www.oracleappsonlinetraining.com/interview.html ------->

http://docstore.mik.ua/orelly/oracle/prog2/ch08_05.htm

http://garethroberts.blogspot.in/search/label/ap

http://knoworacle.wordpress.com/2010/03/02/oracle-payables-useful-tables/

http://knoworacle.wordpress.com/2010/03/22/oracle-financials-accounting-interview-technical-fucntional-questions/

http://eoracleapps.blogspot.in/2009/05/oracle-r12-internal-requisition-and.html

=====

http://alloracletech.blogspot.in/

Contact Person :
Phanindhar Reddy Phone No:
9877349809, 9866349809

http://functionalguy.blogspot.in/2009/02/back-to-back-order-cycle.html

http://oracleapplicationsfunctional.blogspot.in/2011/08/order-to-cash-cycle-in-order-management_28.html

http://knoworacle.wordpress.com/2010/04/14/oracle-applications-general-interview-technical-functional-questions/

http://freshers-oracleapplications.blogspot.in/2008/06/some-of-frequently-asked-questions-in.html

http://www.techonthenet.com/oracle/questions/index.php

http://www.oracle-database-tips.com/oracle_sql_background.html

http://www.psoug.org/reference/merge.html

======

http://www.dba-oracle.com/training.html

http://osamamustafa.blogspot.in

http://support.sas.com/documentation/cdl/en/sqlproc/62086/HTML/default/viewer.htm#a001349366.html

http://sairamgoudmalla.blogspot.in/2008/12/interface-tables-for-most-used-modules.html

http://oracleappsyashrajvarsity.blogspot.in/ ------>good

http://sandeeporaclenotes.blogspot.in/2011/07/unix-material.html?spref=bl --> interfaces,conversions using bulk collect

http://www.oratable.com/nth-highest-salary-in-oracle/

http://sandeeporaclenotes.blogspot.in/2011_07_19_archive.html

http://oracally.blogspot.in/2010/03/r12how-moac-enviornment-is-setup.html

http://2lyoracleapps.blogspot.in/---------->purchasing module

https://incometaxindiaefiling.gov.in/portal/KnowTan.do

http://www.erpstuff.com/forums/default.asp?CAT_ID=1---------->forum for oracle apps

http://oracleappsa2z.blogspot.in/2011/09/all-tables-in-oracle-apps-11i-r12.html

http://www.aboutoracleapps.com/2007/07/oracle-apps-manufacturingaolforms.html

http://sandeeporaclenotes.blogspot.in/

Read more: http://prasanthapps.blogspot.com/2011/05/urls-for-apps.html#ixzz1Tfuff9aY

Posted 29th August 2012 by Krishnareddy

0 Add a comment