13) How to Enable/Disable Concurrent Program Parameter Dynamically

Concurrent program is a data-intensive task.A concurrent program is an executable file that runs simultaneously with other concurrent programs and with online operations, fully utilizing system's hardware capacity.

Standard Request Submission (SRS) is an Oracle Applications feature that allows us to select and run any concurrent programs from a single, standard form (Submit Request) or window (Schedule Request). Requests to run concurrent programs are called concurrent requests.

Basic Requirement

Our basic requirement is to enable or disable concurrent program parameter dynamically.Here we will take two different requirement scenario

Scenario I

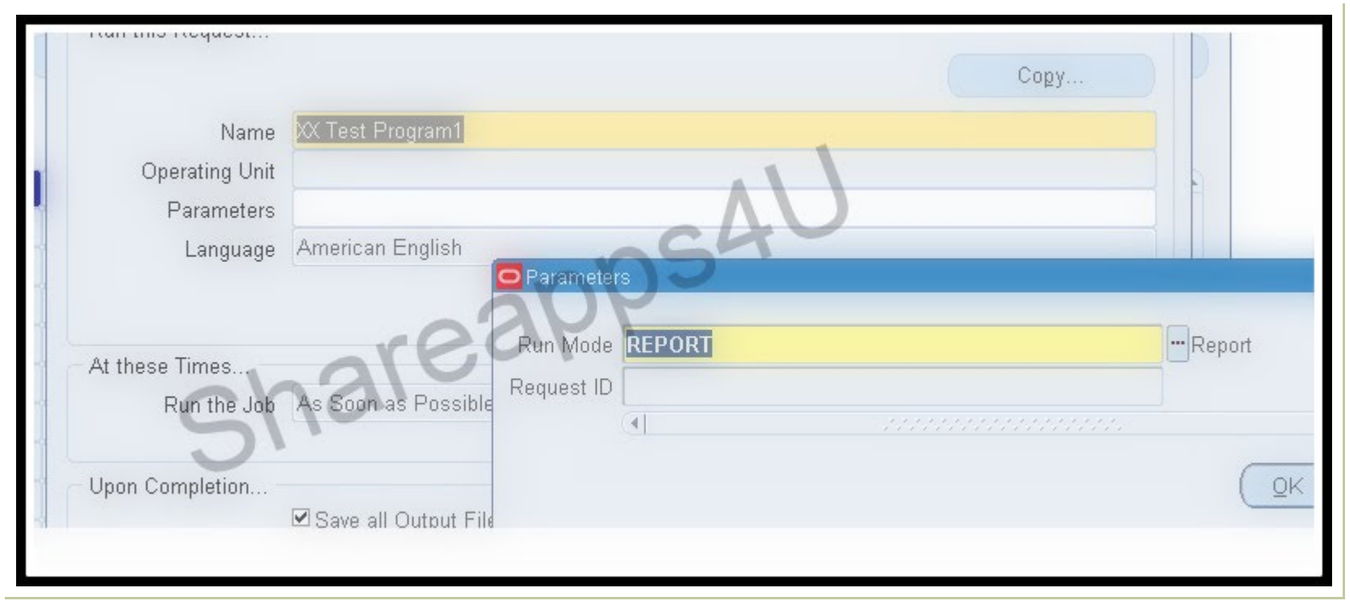
We have to define a concurrent program with the following parameter

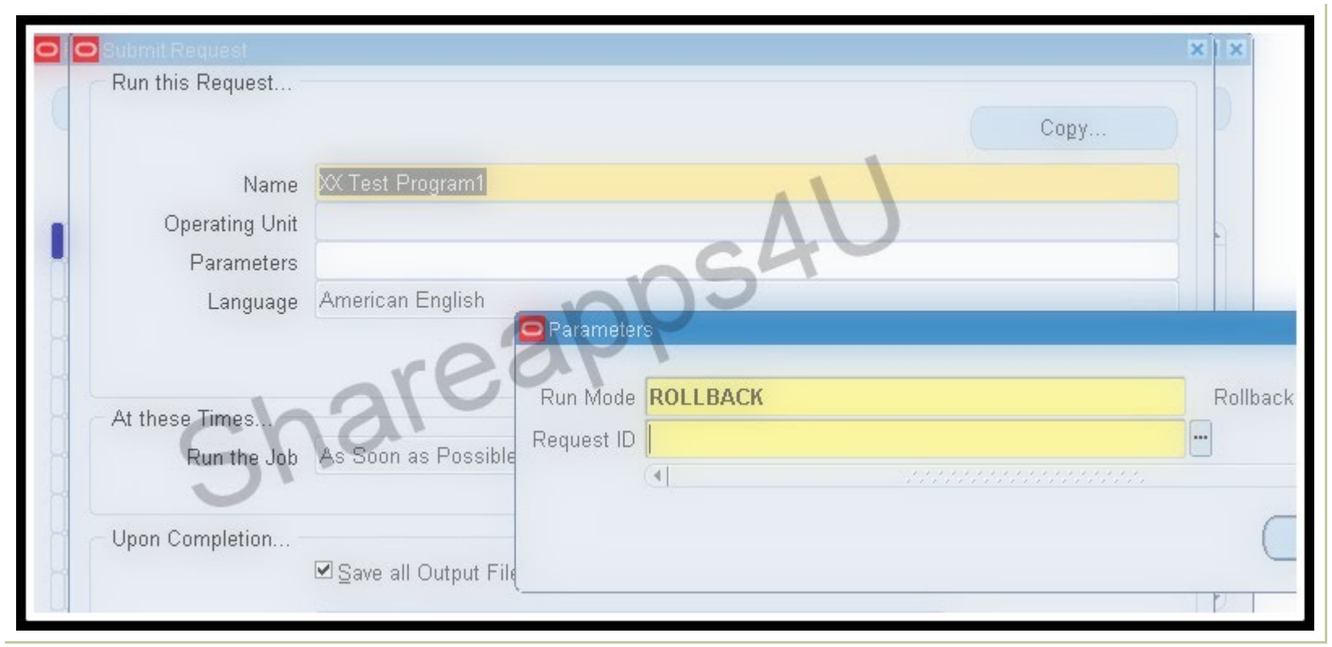
Run Mode:- The parameter will contain a list of values.

- REPORT :- Report mode will generate a report.(as per requirement)
- ROLLBACK :- Rollback mode will rollback/delete the data changes done by previously executed request

Request Id:- This parameter will allow the user to select any of the previously executed concurrent request.

Request Id parameter will be enabled and mandatory if the Run Mode is selected as "ROLLBACK".
If the Run Mode is selected as "REPORT", then Request Id parameter will remain disable.





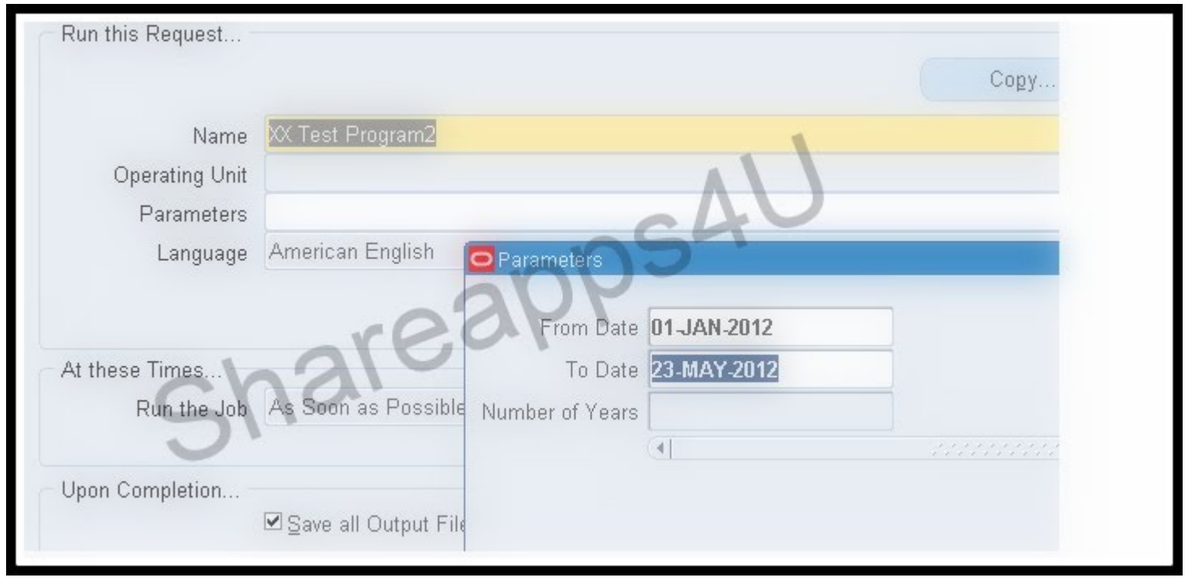
Scenario II

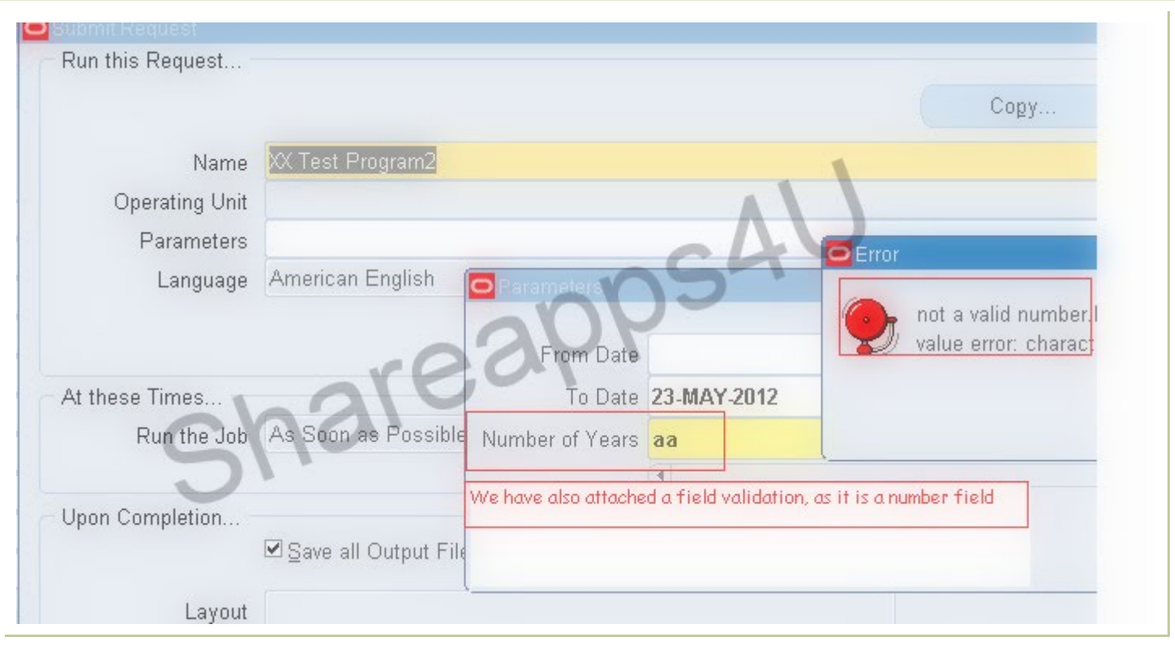
We have to define a concurrent program with the following parameters

From Date :- The From date parameter is a date field. The data in the report will be shown from this date.
To Date :- The To date parameter is a date field. If it is kept as null, program will take system date as its value.
No of Years :- This is a number field.
If the From Date is blank, this field will be enabled and will become mandatory. If "From Date" is entered, then it will be disabled.

This parameter will be use to construct from date

From Date:- <To Date parameter value> - <No. of Years>





Note:- a) Program may have other parameters(as per the business logic) but we are not discussing about those. We will concentrate on these two parameters.

b) The difference between ScenarioI and ScenarioII is as follows

In ScenarioI the dependent field(Request Id) is a custom table validated valueset

Whereas in ScenarioII the dependent field (No. of Years) is a number field(not a table validated value set).

Solution Approach

Before we proceed with the solution first we will discuss about ":$FLEX$.<Value_Set_Name>:NULL" functionality meaning

:$FLEX$.<Value_Set_Name>:NULL

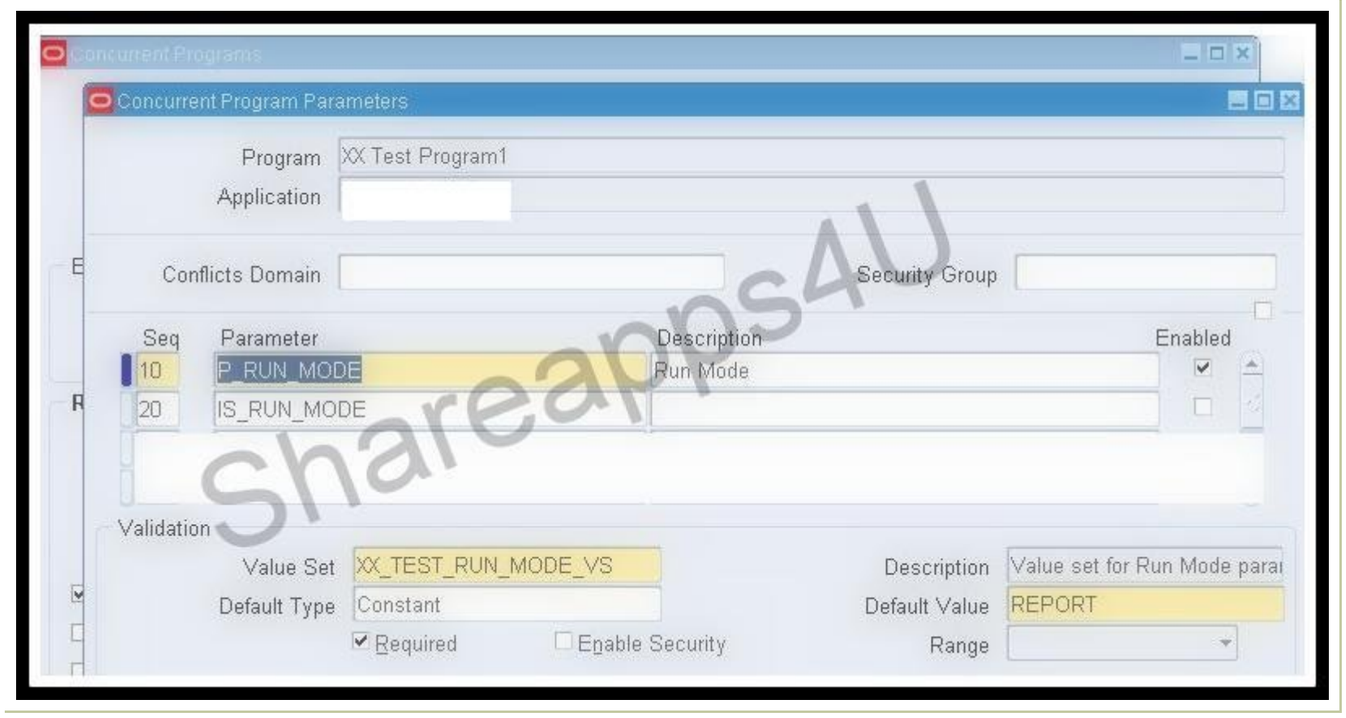
This is a special arguments(bind variables) that can be use in the where clause to retireve a value based on the other field/segment/parameter values.
Value_Set_Name is the name of the value set of a prior parameter that we want our validation table-based values to depend on.
The $FLEX$ mechanism uses the "**closest**" prior parameter with a matching value set name."Value_Set_Name" is case-sensitive, so we have to ensure that it exactly matches with the name of the value set that we use in our prior parameter(parameter with the lower sequence number).

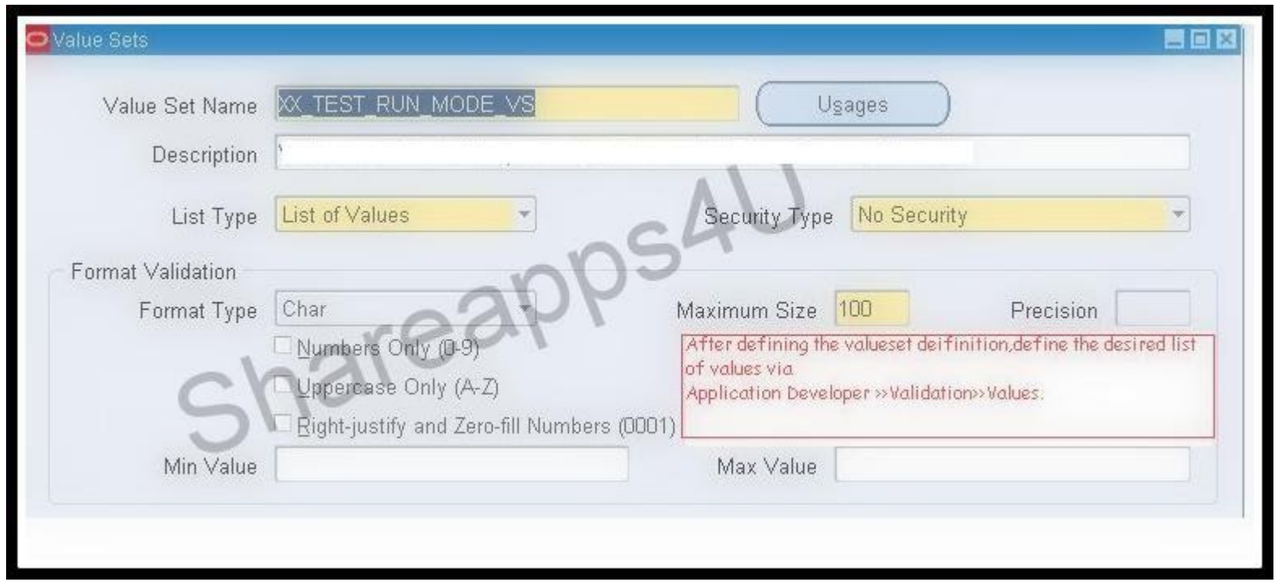
:$FLEX$<Value_SetName> is a bind variable and it is required to have a value.It means it must have a value for the statement, expression.
This can be made optional by using the :NULL suffix.

We will use the above logic to implement our requirement.

Scenario I

1) **Run Mode** :- Create a independent value set with the with desired list of values and attach the same with this parameter in the concurrent program parameter window.

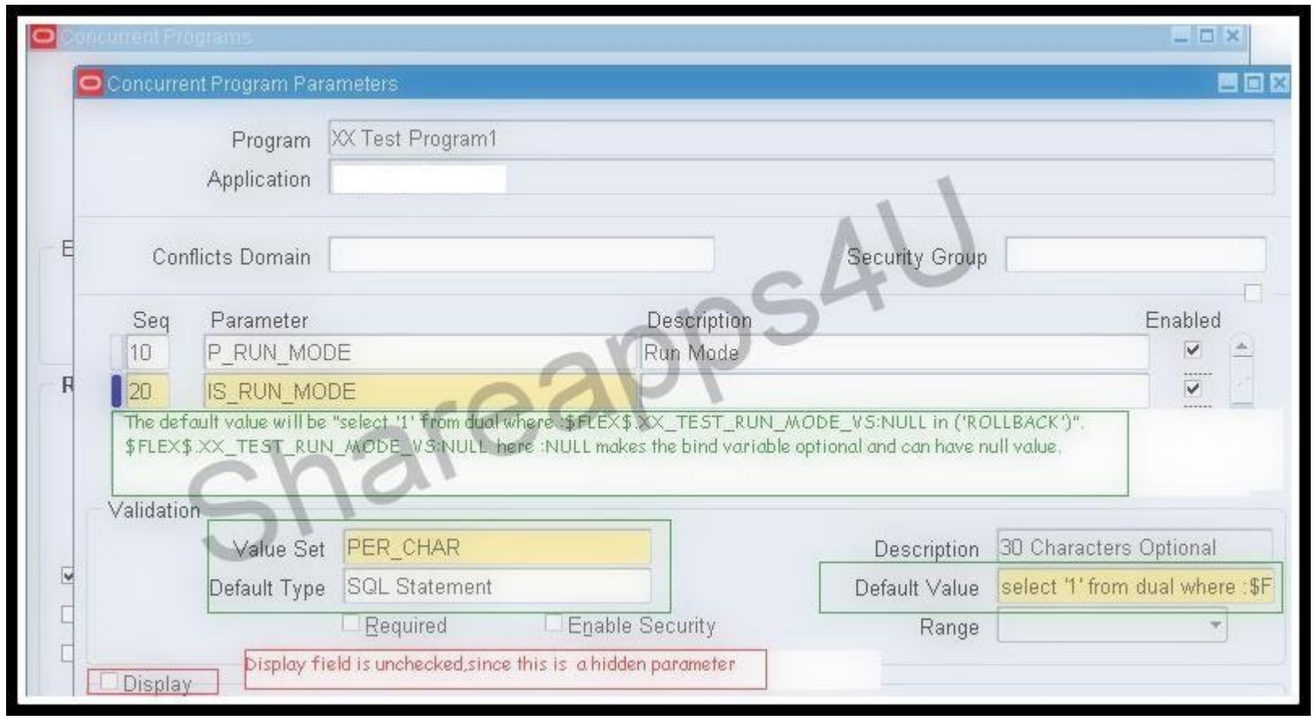




2) Create a hidden parameter with the following details

Seq:-20
Parameter:- IS_RUN_MODE
Value Set:- PER_CHAR *--- we have use PER_CHAR here. One can use other exsisting seeded value set*
Default Type:- SQL Statement
Default Value:- Select '1' from dual where :$FLEX$.XX_TEST_RUN_MODE_VS:NULL in ('ROLLBACK')

The above Default value will return 1 Run Mode is selected as "ROLLBACK". Otherwise it will return NULL.
If it is NULL then it will disable the next paramter where it is use(P_REQUEST_ID).

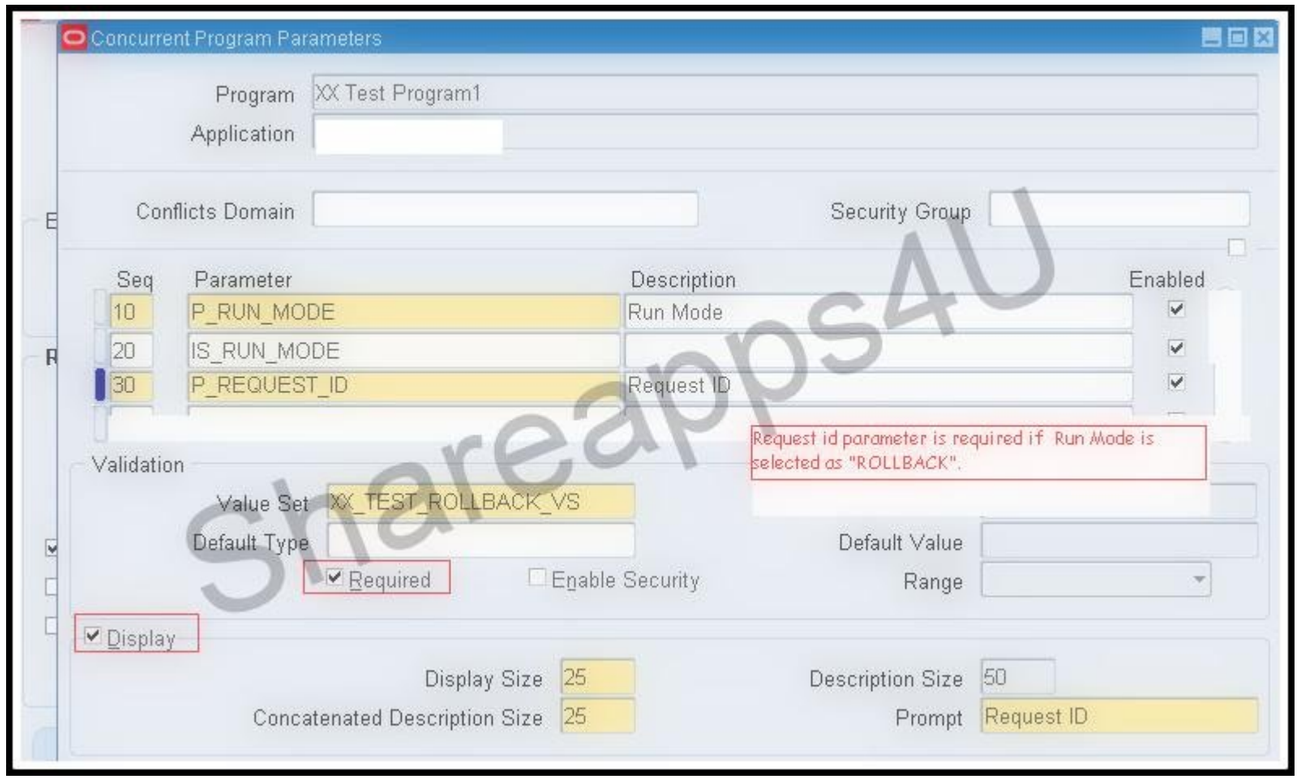


3) **Request Id** :- Create a table validated valueset that will retrieve all the concurrent request(for this program) ran previously. and append the following clause in the table validated valueset

```
and 1=:$FLEX$.PER_CHAR
```

The PER_CHAR value set was used in IS_RUN_MODE hidden parameter.If it is 1 then list of values will be created. Otherwise, no lost of values will be formed and disable the field.

attach the table validated valueset and make the parameter as required.Since for ROLLBACK mode the parameter is a required.



Now our **scenarioI** is ready to use.

Scenario II

1) Create a parameter for From Date using standard date field.

2) Now create a hidden parameter IS_FROM_DATE, to capture whether value is entered or not.This same as we expalined in ScenarioI.
Seq:-20
Parameter:- IS_FROM_DATE
Value Set:- PER_CHAR *--- we have use PER_CHAR here. One can use other exsisting seeded value set*
Default Type:- SQL Statement
Default Value:- Select '1' from dual where :$FLEX$.FND_STANDARD_DATE:NULL is NULL

The above Default value will return 1, if the no value is entered in From Date parameter.(also when the program paramter will open).

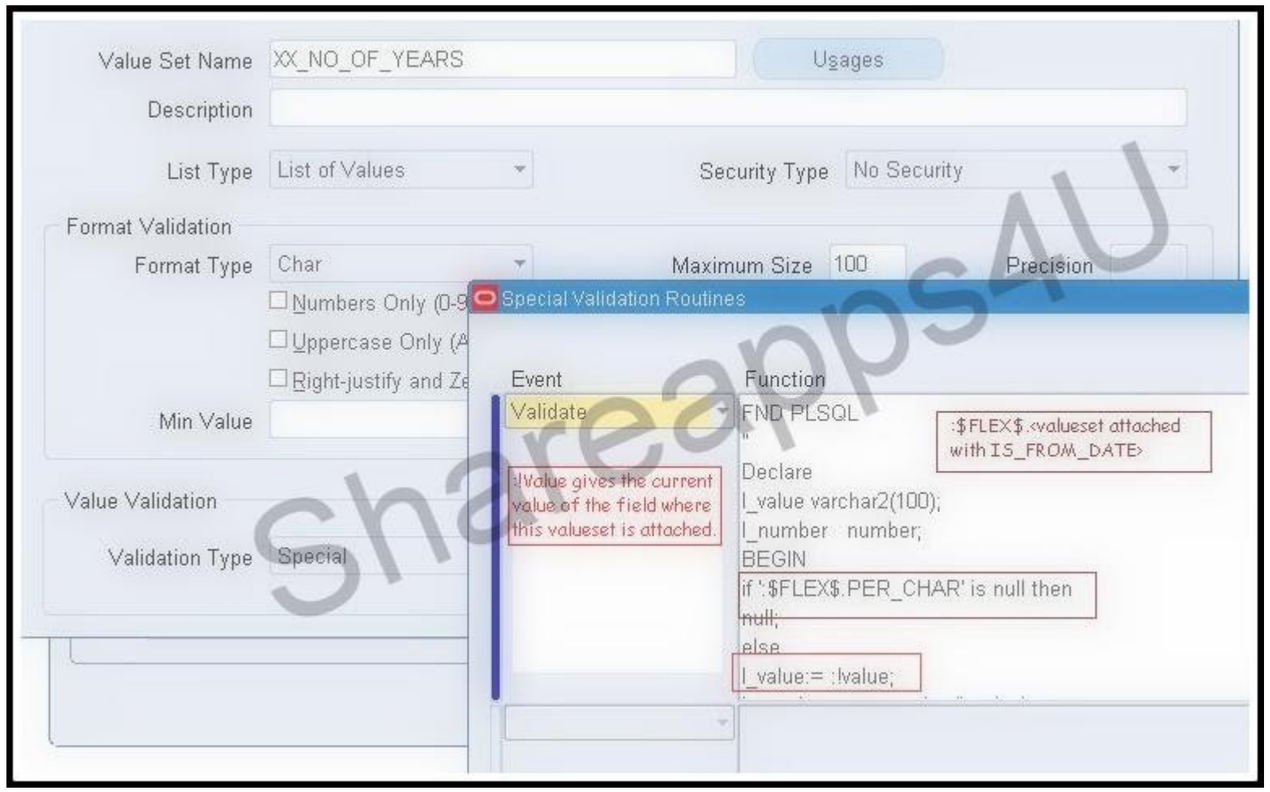
3) Now we will discuss about, how we can construct the P_NO_YEARS parameter. As this is not a table validated valueset, so we cannot use the IS_FROM_DATE directly in any queries (as we did in ScenarioI).

To achieve our goal, we have to use "**Special**" Type value set. In Special valueset we can attach our custom validation depending on the value returened by IS_FROM_DATE parameter.

The validation code is pasted below.

```
FND PLSQL   ------ Starting of the validation
"           --- Your pl/sql block must be with the "<your custom validation>"
Declare
l_value varchar2(100);
l_number   number;
BEGIN
if ':$FLEX$.PER_CHAR' is null then   ----When the IS_FROM_DATE returns null,it means value is entered in "From Date" field
null;                                 --- returning NULL will disable the field.
else
l_value:= :!value;                   --- :!value will gives us the value that is provided in the P_NO_OF_YEARS parameter
/*--- The field value is retrieve, so that we can check value entered in the field must be a numeric value
      You can add your own custom validation depending on the requirement.
 */
l_number :=to_number(l_value);
end if;
exception
when others then
/*
-- FND_GENERIC_MESSAGE is a fnd message that helps us to generate own message as per requirement.
This will raise the exception(as shown the basic requirement section).
```

```
*/
fnd_message.set_name( 'FND', 'FND_GENERIC_MESSAGE' ) ; -
fnd_message.set_token( 'MESSAGE', 'not a valid number.l_value:-'||l_value||sqlerrm);--pass the custom message
fnd_message.raise_error;---Raising the above error
end;
"
```



4) Attach your newly created "Special" type valueset with the P_NO_OF_YEARS parameter.Click the check box "Required", as this is required when "From Date" parameter is not entered.

Now our ScenarioII is also ready to use.

References:-1) https://metalink.oracle.com
2) Oracle Applications Flexfields Guide(Release 12) B31456-01

Comments

You do not have permission to add comments.