



Search web ▼

Home

Learning Topic

- 1) Flexfield
- 10) How to Add Third Party Web Page Link to Oracle Application Web Page
- 11) How to Create Global User Defined Table(UDT)
- 12) How to Add Element Value in Oracle Online Payslip
- 13) How to Enable/Disable Concurrent Program Parameter Dynamically**
- 2) XML Publisher
- 3) Workflow Tutorial
- 4) Form Personalization - How to Change Field Name
- 5) iRecruitment Data Migration - How To Migrate Resume
- 6) How to Restrict Agency from entering Duplicate Candidate
- 7)How to Change User Password
- 8) How To Customise Vacancy Search in iRecruitment- Adding Subsidiary Logo
- 9)Problem With XML Parsing

Sitemap

[Learning Topic](#) >

## 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

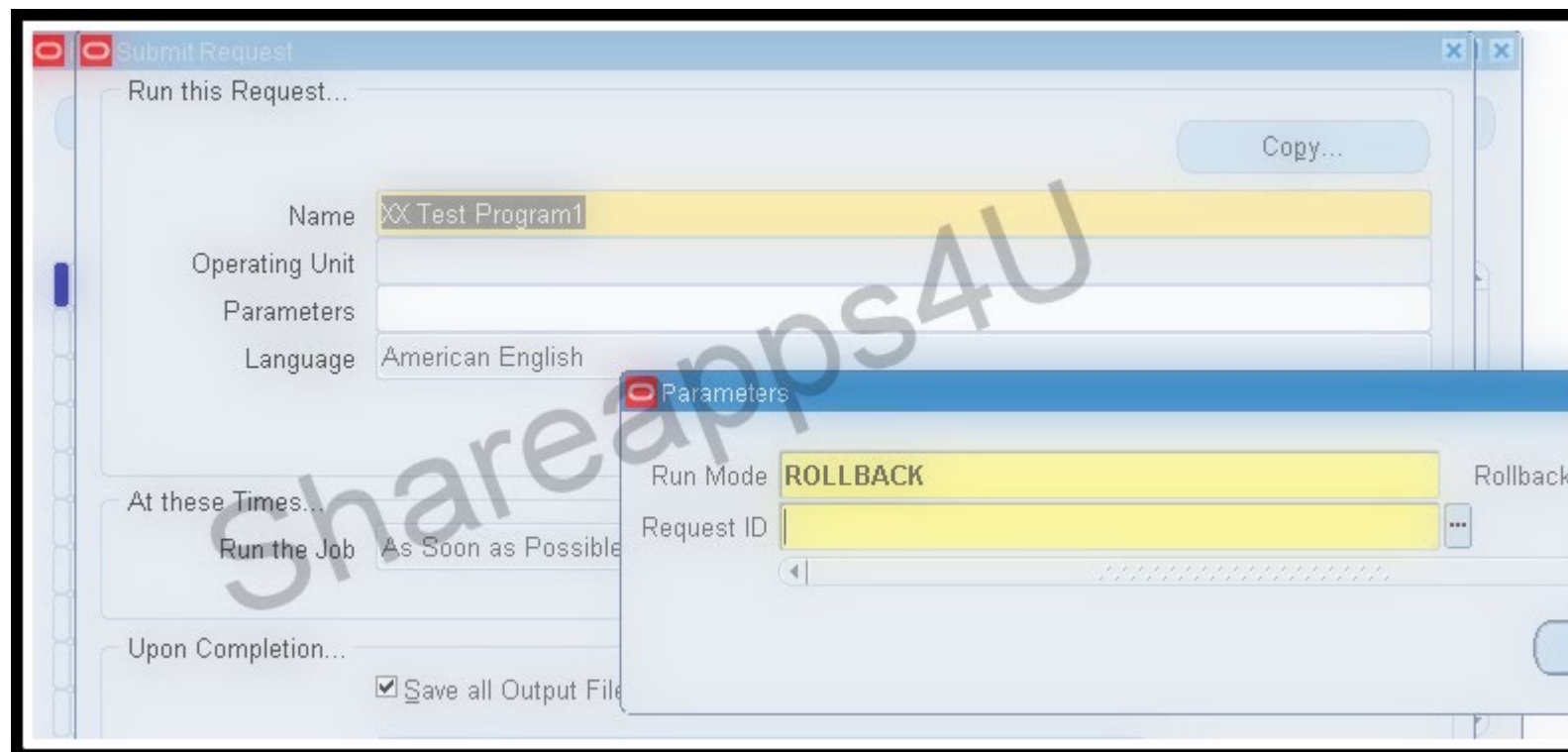
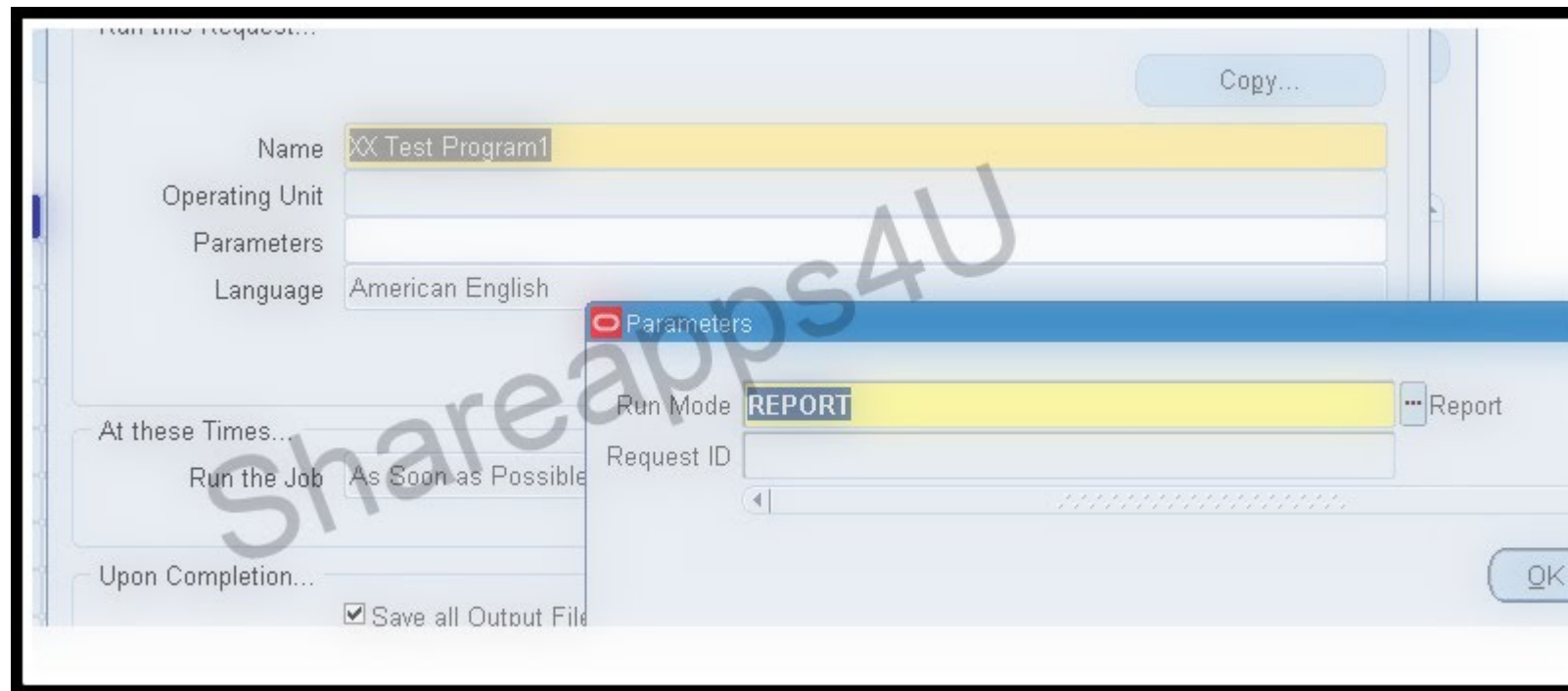
**Run Mode:-** The parameter will contain a list of values.

- 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>

Run this Request...

Copy...

Name: XX Test Program2

Operating Unit:

Parameters:

Language: American English

At these Times...

Run the Job: As Soon as Possible

Upon Completion...

☒ Save all Output File

Parameters Dialog:

From Date: 01-JAN-2012

To Date: 23-MAY-2012

Number of Years:

Submit Request

Run this Request...

Copy...

Name: XX Test Program2

Operating Unit:

Parameters:

Language: American English

At these Times...

Run the Job: As Soon as Possible

Upon Completion...

☒ Save all Output File

Layout:

Parameters Dialog:

From Date:

To Date: 23-MAY-2012

Number of Years: aa

Error Message:

not a valid number.  
value error: character

Note:

We have also attached a field validation, as it is a number field

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 retrieve 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).

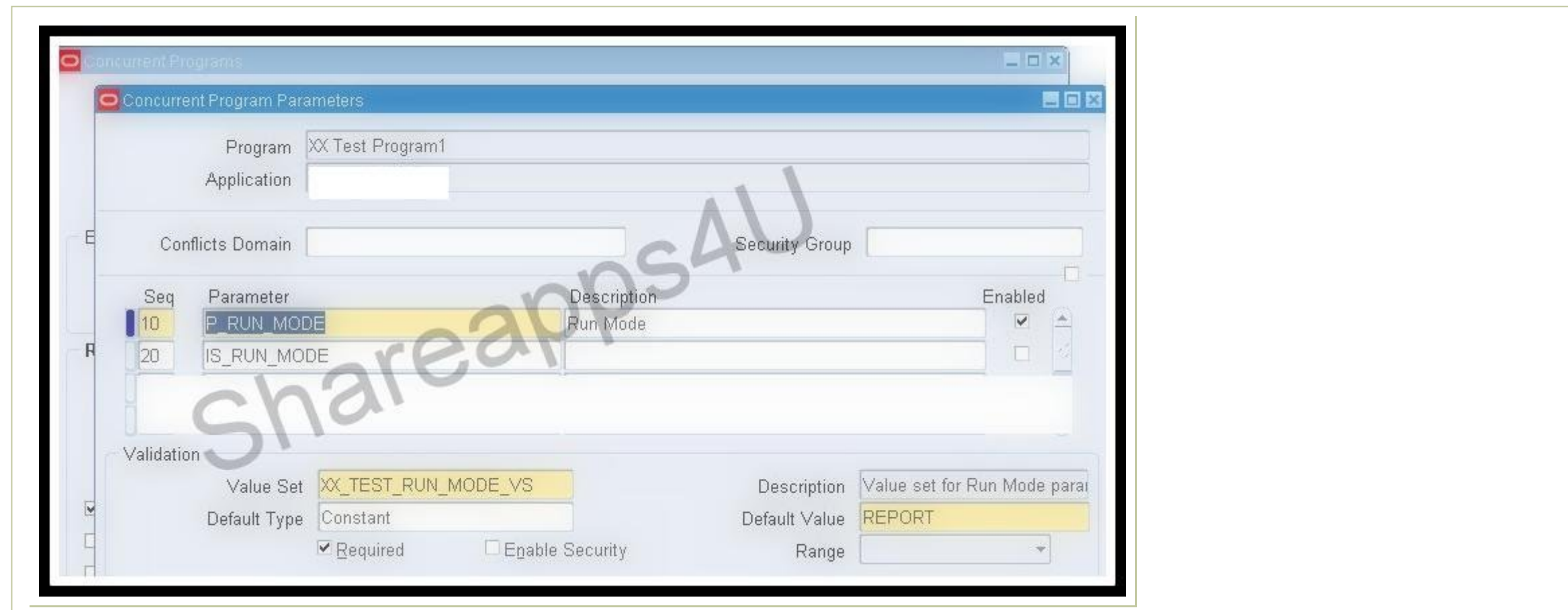
:\$FLEX\$.<Value\_SetName> is a bind variable and it is required to have a value.It means it must have a value for the statement, expression.

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.



Value Set Name: XX TEST RUN MODE VS

Description:

List Type: List of Values

Security Type: No Security

Format Validation:

Format Type: Char

☐ Numbers Only (0-9)

☐ Uppercase Only (A-Z)

☐ Right-justify and Zero-fill Numbers (0001)

Maximum Size: 100

Precision:

Min Value:

Max Value:

After defining the valueset definition, define the desired list of values via Application Developer >> Validation >> Values:

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

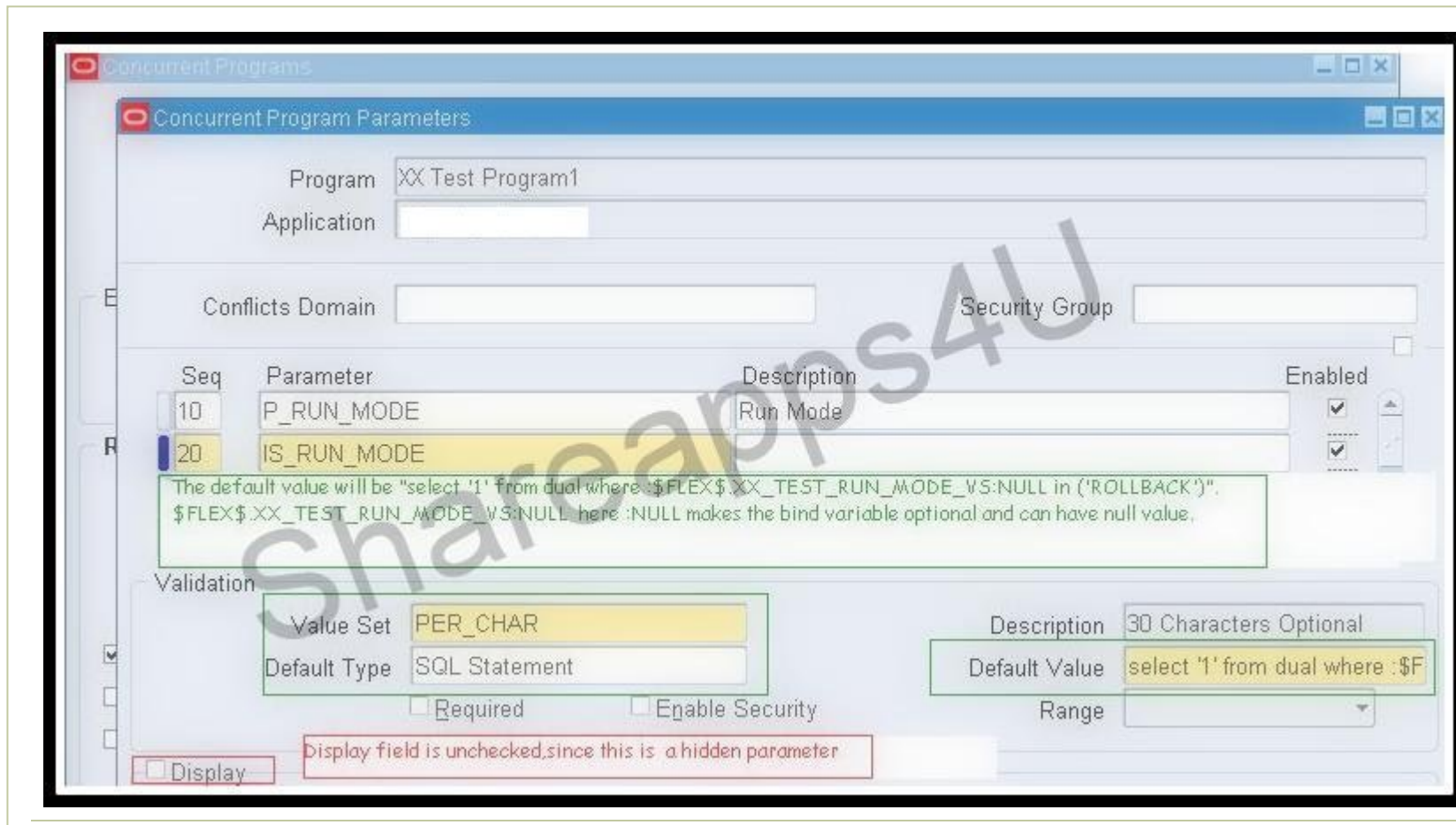
Default Type:- SQL Statement

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 list of values will be formed and disable the field.

Validation Table Information

Table Application: Application Object Library

Table Name: fnd\_concurrent\_requests FCR

☐ Allow Parent Values

Name	Type	Size
Value: FCR.request_id	Varchar2	100
Meaning:		
ID: FCR.request_id	Varchar2	100

Where/Order By: AND 1=:\$FLEX\$.PER\_CHAR|

Additional Columns:

here PER\_CHAR is the name of the valueset that we used in IS\_RUN\_MODE parameter.

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

The screenshot shows the 'Concurrent Program Parameters' window for 'XX Test Program1'. The parameters are listed in a table:

Seq	Parameter	Description	Enabled
10	P_RUN_MODE	Run Mode	<input checked="" type="checkbox"/>
20	IS_RUN_MODE		<input checked="" type="checkbox"/>
30	P_REQUEST_ID	Request ID	<input checked="" type="checkbox"/>

Below the table, the 'Validation' section is configured with:

- Value Set: XX\_TEST\_ROLLBACK\_VS
- Default Type: ☒ Required
- Enable Security: ☐
- Default Value: (empty)
- Range: (empty)
- Display: ☒
- Display Size: 25
- Description Size: 50
- Concatenated Description Size: 25
- Prompt: Request ID

A red box highlights the 'Required' checkbox and the 'Display' checkbox. A red text box states: 'Request id parameter is required if Run Mode is selected as "ROLLBACK".'

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

### Scenario II

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

The screenshot shows the 'Concurrent Program Parameters' window for 'P\_FROM\_DATE'. The parameters are listed in a table:

Seq	Parameter	Description	Enabled
10	P_FROM_DATE		<input checked="" type="checkbox"/>
20	IS_FROM_DATE		<input checked="" type="checkbox"/>
30	P_TO_DATE		<input checked="" type="checkbox"/>
40	P_NO_OF_YEARS		<input checked="" type="checkbox"/>

Below the table, the 'Validation' section is configured with:

- Value Set: FND\_STANDARD\_DATE
- Default Type: (empty)
- Required: ☐
- Enable Security: ☐
- Description: Date value set
- Default Value: (empty)
- Range: (empty)
- Display: ☒
- Display Size: 11
- Description Size: 50
- Concatenated Description Size: 25
- Prompt: From Date



2) Now create a hidden parameter IS\_FROM\_DATE, to capture whether value is entered or not. This same as we explained in Scenario I.

Seq:-20

Parameter:- IS\_FROM\_DATE

Value Set:- PER\_CHAR --- we have use PER\_CHAR here. One can use other existing 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 parameter 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 Scenario I).

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 returned 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 give us the value that is provided in the P_NO_OF_YEARS parameter
/*--- The field value is retrieved, 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.1_value:-'||l_value||sqlerrm);--pass the custom message
fnd_message.raise_error;---Raising the above error
end;
"

```

The screenshot shows the 'Special Validation Routines' configuration window for the value set 'XX\_NO\_OF\_YEARS'. The window is divided into several sections:

- Value Set Name:** XX\_NO\_OF\_YEARS
- Description:** (Empty text field)
- List Type:** List of Values
- Security Type:** No Security
- Format Validation:**
  - Format Type:** Char
  - Maximum Size:** 100
  - Precision:** (Empty text field)
  - ☐ Numbers Only (0-9)
  - ☐ Uppercase Only (A-Z)
  - ☐ Right-justify and Zero Pad
  - Min Value:** (Empty text field)
- Value Validation:**
  - Validation Type:** Special
- Special Validation Routines:**

Event	Function
Validate	<pre> FND_PLSQL " : \$FLEX\$.&lt;valueset attached with IS_FROM_DATE&gt;  Declare l_value varchar2(100); l_number number; BEGIN if '\$FLEX\$.PER_CHAR' is null then null; else l_value:= :lvalue; </pre>

A red box highlights the 'Validate' event, and a red box highlights the 'Function' column. A red box also highlights the 'Special' validation type.

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.

Seq	Parameter	Description	Enabled
10	P_FROM_DATE		<input checked="" type="checkbox"/>
20	IS_FROM_DATE		<input checked="" type="checkbox"/>
30	P_TO_DATE		<input checked="" type="checkbox"/>
40	P_NO_OF_YEARS		<input checked="" type="checkbox"/>

Validation

Value Set  ...

Default Type

☒ Required ☐ Enable Security

Description

Default Value

Range

☒ Display

Display Size  Description Size

Concatenated Description Size  Prompt

Now our ScenarioII is also ready to use.

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

**Disclaimer:-** This is a knowledge sharing site. This topic talks about a custom solution. Oracle may not provide you a support for any data corruption or any other problem in your custom code/problem arises because of the custom code. The author is not responsible for any kind of system/data problem appears because of usages of this code.Reader/implementer must do it on his/her own risk/responsibility.

Comments

You do not have permission to add comments.