

Oracle Forms

A dynamic Report launcher

[Home page](#)

1. Purpose

This is a tutorial and an Oracle Forms tool that show how, in a single dialog, to launch any report with 0 up to 10 parameters.

It needs 2 Oracle tables to store definition of the reports and their associated parameters.

Each parameter support the following properties:

- Required
- Datatype
- Hint text
- Default value
- Format mask
- LOV select order
- Validation rule

Two dialogs are provided to run the demonstration

- DYN_REP.FMB to launch the reports
- DYN_REP_BUILD.FMB to manage report and parameters definition

2. The Oracle tables

The table that store the reports properties

```
CREATE TABLE REP
(
  REP_NUM          NUMBER                NULL,
  REP_NAME         VARCHAR2(100 BYTE)    NOT NULL,
  REP_TITLE        VARCHAR2(256 BYTE)    NOT NULL,
  REP_SERVER       VARCHAR2(100 BYTE)    NULL,
  REP_DESTYPE      VARCHAR2(20 BYTE)     DEFAULT 'CACHE'          NOT NULL,
  REP_DESNAME      VARCHAR2(256 BYTE)    NULL,
  REP_DESFORMAT   VARCHAR2(20 BYTE)     DEFAULT 'PDF'            NOT NULL,
  REP_EXEC_MODE    VARCHAR2(30 BYTE)     DEFAULT 'ASYNCHRONOUS'   NOT NULL,
  REP_COMM_MODE    VARCHAR2(15 BYTE)     DEFAULT 'BATCH'          NOT NULL
)
```

The table that store the report parameters properties

```
CREATE TABLE REP_PARAM
(
  REP_NUM          NUMBER                NULL,
  REP_PAR_REP_NAME VARCHAR2(30 BYTE)     NULL,
  REP_PAR_ORDER    NUMBER(2)             DEFAULT 1                NULL,
  REP_PAR_TITLE    VARCHAR2(256 BYTE)    NOT NULL,
  REP_PAR_TYPE     VARCHAR2(20 BYTE)     DEFAULT 'CHAR'          NOT NULL,
  REP_PAR_LOV      VARCHAR2(2000 BYTE)   NULL,
  REP_PAR_VALIDATION VARCHAR2(512 BYTE)  NULL,
  REP_PAR_DEFAULT  VARCHAR2(256 BYTE)    NULL,
  REP_PAR_HINT     VARCHAR2(128 BYTE)    NULL,
  REP_PAR_ERRMSG   VARCHAR2(256 BYTE)    NULL,
  REP_PAR_CASE     VARCHAR2(1 BYTE)      DEFAULT 'M'              NOT NULL,
  REP_PAR_FORMAT_MASK VARCHAR2(50 BYTE)  NULL,
  REP_PAR_REQUIRED VARCHAR2(1 BYTE)      DEFAULT 'N'              NOT NULL
)
```

3. The launcher dialog

The list item (Report title) allows to choose the report to launch and display dynamically the associated parameters.

The form contains 10 CHAR items, 10 NUMBER items and 5 DATE items.

When you choose a specific report, the table that contain the parameter description is read, and the corresponding items are showed on the screen.

This operation is performed in the *INIT_PARAMS* program unit called by the When-List-Changed trigger:

```

PROCEDURE Init_Params IS
  LC$Req Varchar2(512) ;

  Cursor C_PARAMS IS
  Select      *
  From        REP_PARAM
  Where       REP_NUM = :REP.REP_LIST
  Order by   REP_PAR_ORDER ;

  LR$Rec C_PARAMS%ROWTYPE ;
  LN$I   Pls_Integer := 1 ;
BEGIN

  -- Hide all parameters --
  For i IN 1..10 Loop
    Set_Item_Property( 'PARAMS.L' || Ltrim( To_char( i,'00' ) ), VISIBLE, PROPERTY_FALSE ) ;
    Set_Item_Property( 'PARAMS.C' || Ltrim( To_char( i,'00' ) ), VISIBLE, PROPERTY_FALSE ) ;
    Set_Item_Property( 'PARAMS.N' || Ltrim( To_char( i,'00' ) ), VISIBLE, PROPERTY_FALSE ) ;
  End loop ;
  For i IN 1..5 Loop
    Set_Item_Property( 'PARAMS.D' || Ltrim( To_char( i,'00' ) ), VISIBLE, PROPERTY_FALSE ) ;
  End loop ;

  If :REP.REP_LIST IS NOT NULL Then

    -- Populate the internal collection with parameter properties --

```

```

PKG_VARS.TB_Params.DELETE ;
Open C_PARAMS ;
Loop
    Fetch C_PARAMS Into LR$Rec ;
    Exit When C_PARAMS%NOTFOUND ;
    PKG_VARS.TB_Params(LN$I) := LR$Rec ;
    LN$I := LN$I + 1 ;
End loop ;
Close C_PARAMS ;

If PKG_VARS.TB_Params.COUNT > 0 Then
    -- Show corresponding items --
    Display_Params ;
End if ;

End if ;

END;

PROCEDURE Display_Params IS
    LN$PosY    Pls_integer ;
    LC$Item    Varchar2(61) ;
    LC$First   Varchar2(61) ;
    LC$Prec    Varchar2(61) ;
    LN$I       Pls_Integer ;
    LN$C       Pls_Integer := 1 ;
    LN$N       Pls_Integer := 1 ;
    LN$D       Pls_Integer := 1 ;
    LN$Height  Pls_Integer := 0 ;
BEGIN

    LN$PosY := Get_Item_Property( 'PARAMS.L01', Y_POS ) ;

    -- Show parameters --
    LN$I := 1 ;
    For i IN PKG_VARS.TB_Params.First .. PKG_VARS.TB_Params.Last Loop

        LC$Item := 'L' || ltrim(to_char(LN$I,'00')) ;
        Copy( PKG_VARS.TB_Params(i).REP_PAR_TITLE, LC$Item ) ;
        Set_Item_Property( LC$Item , VISIBLE, PROPERTY_TRUE ) ;
        Set_Item_Property( LC$Item, Y_POS, LN$PosY ) ;

        If PKG_VARS.TB_Params(i).REP_PAR_TYPE = 'CHAR' Then
            LC$Item := 'C' || ltrim(to_char( LN$C, '00' ) ) ;
            LN$C := LN$C + 1 ;
        ElseIf PKG_VARS.TB_Params(i).REP_PAR_TYPE = 'NUMBER' Then
            LC$Item := 'N' || ltrim(to_char( LN$N, '00' ) ) ;
            LN$N := LN$N + 1 ;
        Else
            LC$Item := 'D' || ltrim(to_char( LN$D, '00' ) ) ;
            LN$D := LN$D + 1 ;
        End if ;

        PKG_VARS.TB_Items(i) := LC$Item ;

        If LN$I = 1 Then
            LC$First := 'PARAMS.' || LC$Item ;
        End if ;
        Set_Item_Property( LC$Item, VISIBLE, PROPERTY_TRUE ) ;
        Set_Item_Property( LC$Item, ENABLED, PROPERTY_TRUE ) ;
        Set_Item_Property( LC$Item, INSERT_ALLOWED, PROPERTY_TRUE ) ;
        Set_Item_Property( LC$Item, UPDATE_ALLOWED, PROPERTY_TRUE ) ;
        Set_Item_Property( LC$Item, Y_POS, LN$PosY ) ;
        -- Default value --
        If PKG_VARS.TB_Params(i).REP_PAR_DEFAULT IS NOT NULL Then
            Copy( PKG_VARS.TB_Params(i).REP_PAR_DEFAULT, LC$Item ) ;
        End if ;
        -- Parameter name --
        Set_Item_Property( LC$Item, TOOL TIP_TEXT, PKG_VARS.TB_Params(i).REP_PAR_REP_NAME ) ;
        -- Hint text --
        Set_Item_Property( LC$Item, HINT_TEXT, PKG_VARS.TB_Params(i).REP_PAR_HINT ) ;
        -- Format mask --
        Set_Item_Property( LC$Item, FORMAT_MASK, PKG_VARS.TB_Params(i).REP_PAR_FORMAT_MASK ) ;
        -- Case restriction --
        If PKG_VARS.TB_Params(i).REP_PAR_CASE = 'U' Then
            Set_Item_Property( LC$Item, CASE_RESTRICTION, UPPERCASE ) ;
        ElseIf PKG_VARS.TB_Params(i).REP_PAR_CASE = 'L' Then
            Set_Item_Property( LC$Item, CASE_RESTRICTION, LOWERCASE ) ;
        End if ;
        -- Required ? --
        If PKG_VARS.TB_Params(i).REP_PAR_REQUIRED = 'Y' Then
            Set_Item_Property( LC$Item, REQUIRED, PROPERTY_TRUE ) ;
        End if ;

        If LN$I > 1 Then
            Set_Item_Property( LC$Prec, NEXT_NAVIGATION_ITEM, LC$Item ) ;
            Set_Item_Property( LC$Item, PREVIOUS_NAVIGATION_ITEM, LC$Prec ) ;
        End if ;

        LC$Prec := LC$Item ;
        LN$I := LN$I + 1 ;
        LN$PosY := LN$PosY + Get_Item_Property( LC$Item, HEIGHT ) ;
    End loop ;

```

```

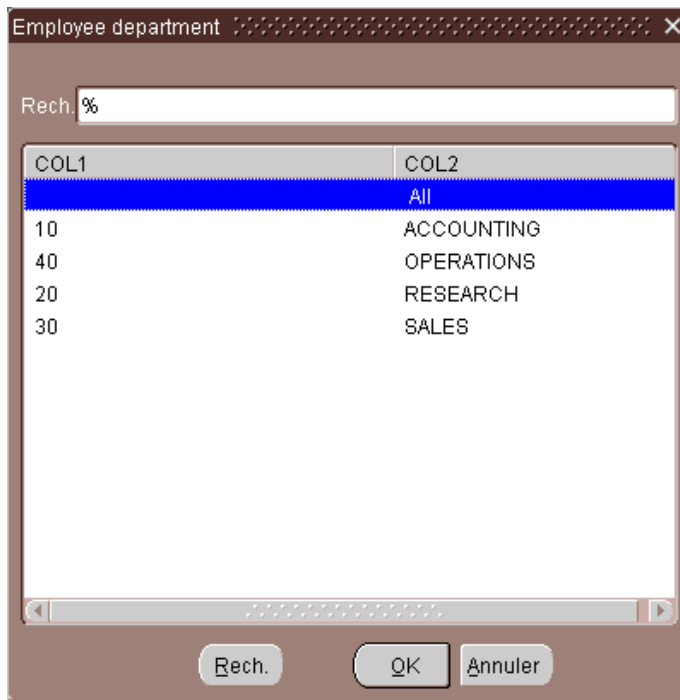
If LN$I > 1 Then
    Set_Item_Property( LC$Item, NEXT_NAVIGATION_ITEM, LC$First ) ;
    Set_Item_Property( LC$First, PREVIOUS_NAVIGATION_ITEM, LC$Item ) ;
End if ;

If LN$I > 1 Then
    Go_Item( LC$First ) ;
Else
    Go_Block( 'REP' ) ;
End if ;

END;

```

If a LOV select order is setted on a parameter, you can display the associated LOV



There are 5 LOVs defined in the module (for 1, 2, ..., 5 columns)

The value returned is always the first column of the LOV, and the target item is :CTRL.RECEPT

This is the code that constructs the LOV, called in the When-New-Item-Instance trigger:

```
Display_Lov( :REP.REP_LIST, Get_Block_Property( 'PARAMS', CURRENT_RECORD ) ) ;
```

```

PROCEDURE Display_Lov
(
    PN$Num    IN NUMBER,
    PN$Col    IN NUMBER
) IS
    LC$Select  Varchar2(2000) ;
    LC$title   Varchar2(256) ;
    rg_name    Varchar2(20) := 'RG_GROUP' ;
    rg_id      RecordGroup ;
    err        Number ;

    c NUMBER;
    d NUMBER;
    col_cnt INTEGER;
    rec_tab dbms_sql.desc_tab2;
    col_num NUMBER;

    LC$Lov     Varchar2(10) ;
    LC$Col     Varchar2(100) ;

    LC$Item    Varchar2(61) := :System.Trigger_Item ;
    LN$width   Pls_Integer ;
    LN$TotWidth Pls_integer := 0 ;

BEGIN

    LC$Col := Get_Item_Property( LC$Item, TOOLTIP_TEXT ) ;

```

```

-- Get The Select order --
Begin
  Select  REP_PAR_LOV, REP_PAR_TITLE
  Into    LC$Select, LC$Title
  From    REP_PARAM
  Where   REP_NUM = PN$Num
  And     REP_PAR_REP_NAME = LC$Col
  ;

Exception
  When NO_DATA_FOUND Then
    goto the_end ;
End ;

If LC$Select IS NULL Then
  goto the_end ;
End if ;

BEGIN
  c := dbms_sql.open_cursor;
  dbms_sql.parse(c, LC$Select, 1);
  d := dbms_sql.EXECUTE(c);
EXCEPTION
  WHEN OTHERS THEN
    dbms_sql.close_cursor(c);
    Raise form_trigger_failure ;
END ;
dbms_sql.describe_columns2(c, col_cnt, rec_tab);
dbms_sql.close_cursor(c);

col_num := rec_tab.last ;

-- LOV name --
LC$Lov := 'LV' || ltrim( to_char( col_num ) ) || 'C' ;

rg_id := Find_Group( rg_name ) ;
If not ID_NULL( rg_id ) Then
  Delete_Group( rg_id ) ;
End if ;

-- Create and populate the record group --
rg_id := Create_Group_From_Query( rg_name, LC$Select ) ;
err := Populate_Group( rg_name ) ;

-- Set the LOV column properties --
For i In rec_tab.first .. rec_tab.last Loop
  -- Title --
  Set_Lov_Column_Property( LC$Lov, i, TITLE, rec_tab(i).col_name ) ;
  -- Width --
  IF rec_tab(i).col_type = 1 THEN
    LN$Width := rec_tab(i).col_max_len * 11 ;
  ELSIF rec_tab(i).col_type = 2 THEN
    LN$Width := rec_tab(i).col_precision * 11 ;
  ELSIF rec_tab(i).col_type = 12 THEN
    LN$Width := 80 ;
  END IF ;
  If LN$Width > 200 Then
    LN$Width := 200 ;
  End if ;
  Set_Lov_Column_Property( LC$Lov, i, WIDTH, LN$Width ) ;
  LN$TotWidth := LN$TotWidth + LN$Width + 20 ;
End loop ;

-- LOV properties --
Set_Lov_Property( LC$Lov, LOV_SIZE, LN$TotWidth, 400 ) ;
Set_Lov_Property( LC$Lov, TITLE, LC$Title ) ;
Set_Lov_Property( LC$Lov, GROUP_NAME, rg_name ) ;

Set_Item_Property( LC$Item, LOV_NAME, LC$Lov ) ;

<<the_end>>
null ;

END;

```

4. The builder dialog

The screenshot shows the 'Report parameter builder' dialog in Oracle Forms 9i/10g. The window title is 'Oracle Forms 9i/10g - Dynamic Report Launcher'. The menu bar includes 'Action', 'Edition', 'Interrogation', 'Bloc', 'Enregistrement', 'Champ', 'Aide', and 'Fenêtre'. The toolbar contains various icons for file operations and navigation.

The 'List of Reports' section contains the following fields:

- Num: 1
- Name: EMP_LIST.rdf
- Title: List of employees
- Rep Server: rep00server
- Desttype: Cache File Printer Mail
- Desname: c:\emplist.pdf
- Desformat: PDF
- Exec Mode: Asynchronous Synchronous
- Comm Mode: Batch Runtime

The 'List of parameters' section contains a table with the following columns: Order, Name, Req., Title, and Datatype.

Order	Name	Req.	Title	Datatype
1	P_ENAME	<input type="checkbox"/>	Employee name	CHAR
2	P_DEPTNO	<input type="checkbox"/>	Employee department	NUMBER
3	P_DATE	<input type="checkbox"/>	Employee hiredate	DATE
		<input type="checkbox"/>		
		<input type="checkbox"/>		
		<input type="checkbox"/>		

At the bottom of the dialog, there is a status bar showing 'Enreg. : 1/1'.

It allows to set the reports and parameters properties.

- Enter in the first block the basic report specifications
- Enter in the second block the parameters specifications

Name is the name of the parameter defined in the report module

Req allows to set the parameter required or not

Title is the parameter label

	Datatype	Case	LOV order
<input type="checkbox"/>	CHAR	U	
<input type="checkbox"/>	NUMBER	M	select NULL COL1, 'All' COL2 From Dual <input type="checkbox"/> Union <input type="checkbox"/> select to_char(c
<input type="checkbox"/>	DATE	M	
<input type="checkbox"/>			
<input type="checkbox"/>			
<input type="checkbox"/>			

Datatype can be CHAR, NUMBER or DATE.

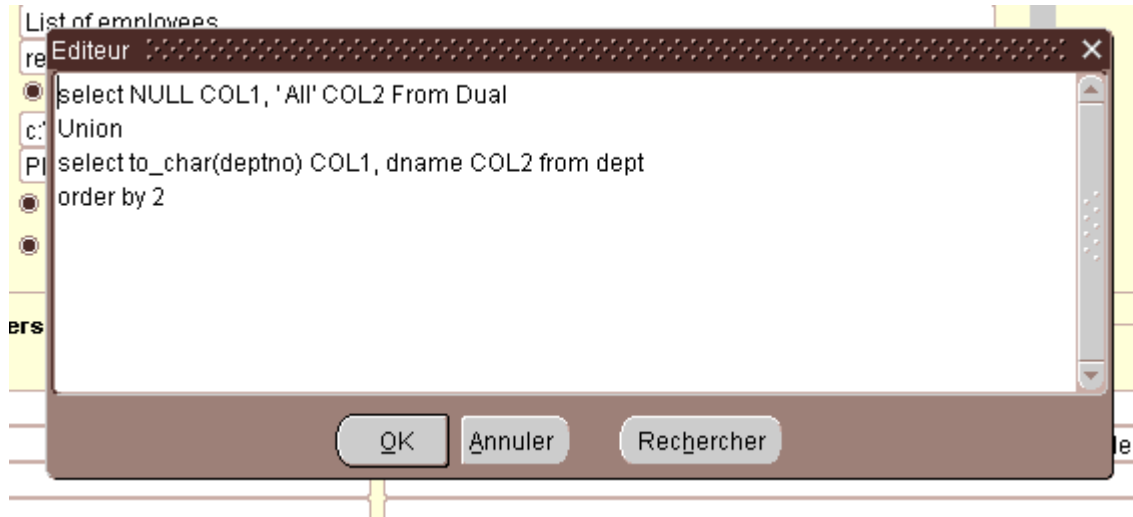
Case allows to set the restriction case of the parameter.

Allowed values are:

- (U)ppercase
- (L)owercase
- (M)ixt

LOV order allows to define the SELECT order for the lov attached on the parameter.

You can define SELECT order with 1 up to 5 columns.
 Each column must have an alias (COL1 to COL5)



Validation rule

IN('SMITH','SCOTT')
> '31/12/1980'

Validation rule allows to set a valid SQL to validate the parameter.

Default value	Hint
	Enter an employee name
10	Enter a department
	Enter a hiredate

Error message

Employee name must be 'SMITH' or 'SCOTT'
Hiredate must be greatest than '31/12/1980'

Error message allows to set an error message displayed to the end user when the validation failed.

Format Mask

	DD/MM/YYYY

You can also define a format mask for the parameter.

5. The sample dialogs

- [Download the dynrep.zip file](#)
- Unzip the **dynrep.zip** file
- Under Sql*Plus, run the **/scripts/install.sql** to create the following tables:
 1. EMP
 2. DEPT
 3. REP
 4. REP_PARAM
- Open the **/reports/EMP_LIST.rdf** and **/reports/DEPT_LIST.rdf** modules (Oracle Reports 9.0.2)
- Open the **/forms/DYN_REP.FMB** and **/forms/DYN_REP_BUILD.FMB** modules (Oracle Forms 9.0.2)
- Compile all and run the modules