Oracle Forms 10*g* – Dynamic LOVs

An Oracle Forms Community White Paper

François Degrelle
f.degrelle@free.fr
September 2005

ORACLE

INTRODUCTION

The solution described in this paper enables Oracle Forms developers to centrally manage List of Values (LOV). Changing a LOV in a Forms module doesn't require to open the module or to re-compile it.

The LOV display dialog filters the list of values with each character the user types into the LOV search field. The user can choose the column on which the filtering occurs freely.

Similar to the functionality of Java Swing tables that enable users to swap the position of columns in a table, the LOV utility promoted in this paper supports personalization of the LOV dialog. The utility sources can be downloaded from the Oracle Technology Network (OTN[1])

A global search mode enables to find a value in all the VARCHAR2, NUMBER and DATE columns of an associated table.

Implementing the solution described in this paper does provide the following benefits

Oracle Forms application developer:

- Centralized LOV handling within only one single dialog
- LOV creation, update and delete without re-compiling and re-deploying Forms modules
- Free definition of the LOV column that is used for item validation
- Configure the LOV to wait for the user to enter one or more characters before populating the list to improve LOV dialog performance

Oracle Forms application users

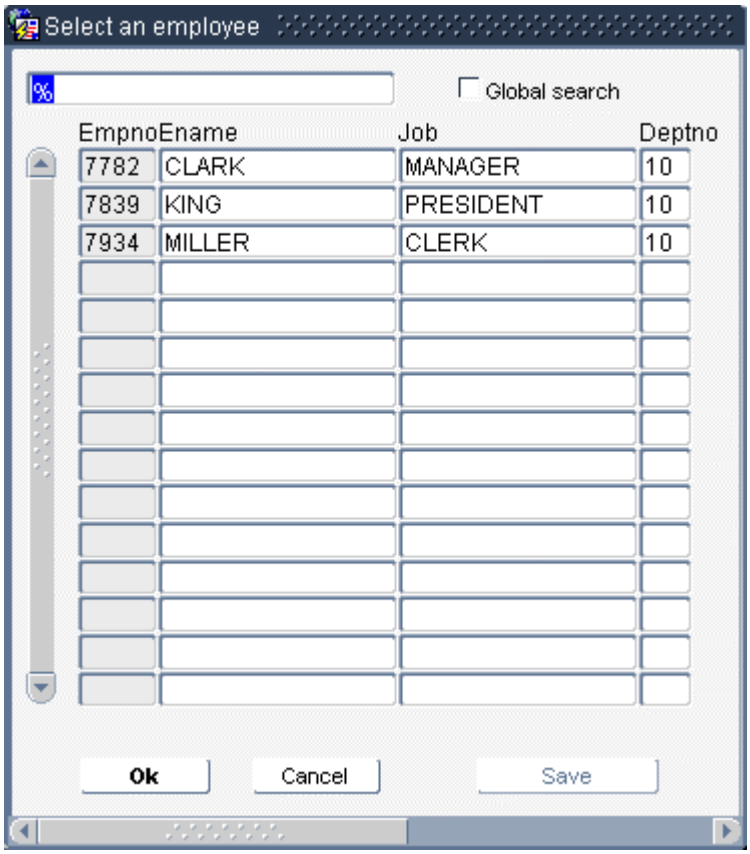
- Define the LOV column to base the search on
- Resize the width of columns in the LOV list

[1] See the Oracle Forms 10g collateral section at Otn.oracle.com/products/forms

- Swap columns
- Use the global mode to search a value in all the standard columns of the table
- Personalize the default settings for future use

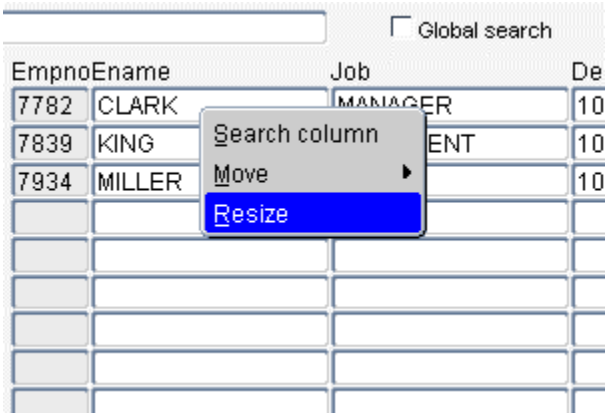
THE GENERIC LOV SCREEN

The following screenshots demonstrate different aspects of the LOV utility. Note that the LOV panel is build as a separate window that uses a Forms table to represent the list of values.

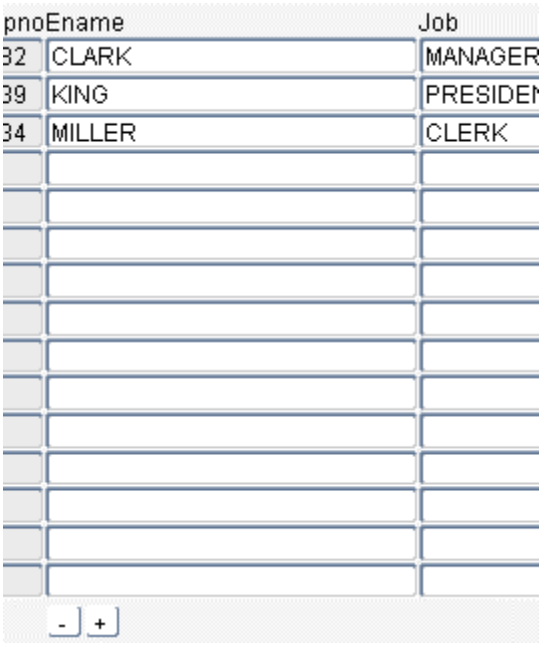


LOV column resizing

The LOV table has a context menu attached to it. To resize a column, select it and press the right mouse button to display the context menu. Choose the menu **Resize** option or press the shift+R key .

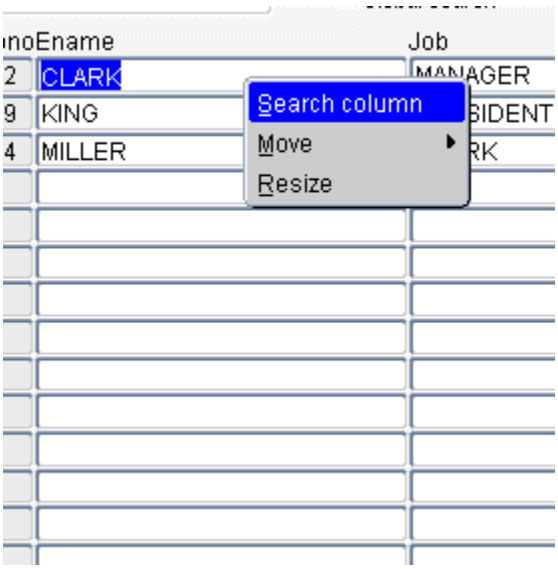


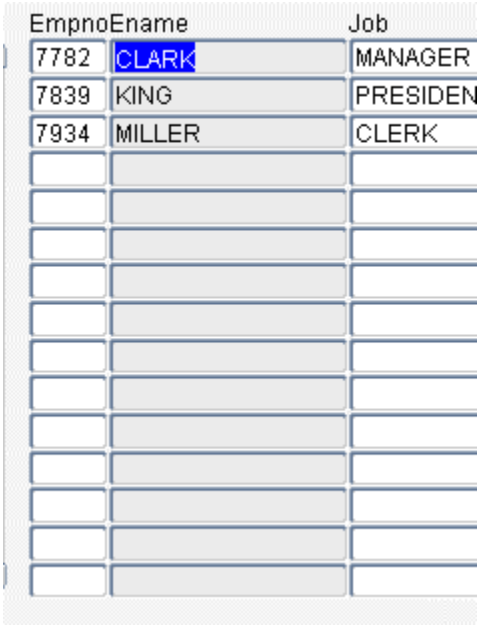
Buttons show on the bottom of the LOV table that allow the application user to either shrink or increase the column size.



Changing the search column

Select the LOV table and open the context menu. Choose **Search column** from the list of available options or press shift+S.

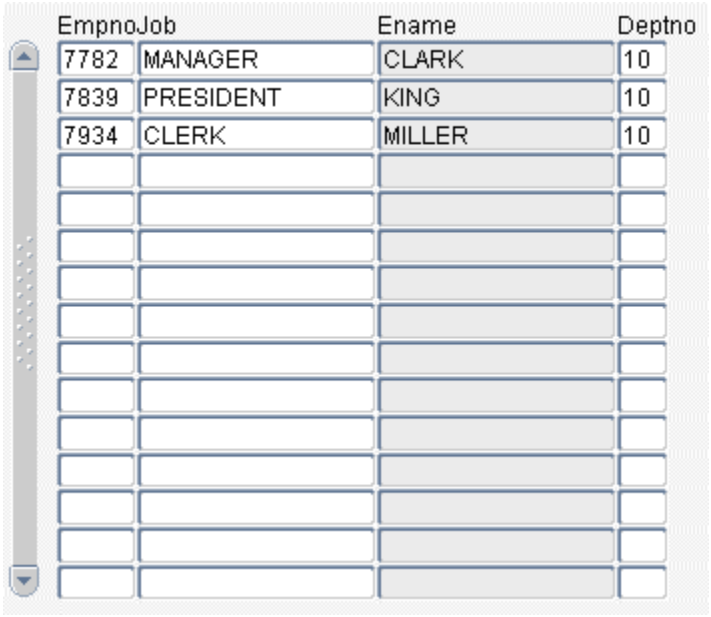




Users can see the new search column is **Ename**, because of its gray background color.

Move columns

The **Move** option of the LOV context menu allows the application user to move a selected LOV column to either the right or the left.



Empno	Job	Ename	Deptno
7782	MANAGER	CLARK	10
7839	PRESIDENT	KING	10
7934	CLERK	MILLER	10

As you can see, the Job column is, now, the second in the table.

LOV Personalization

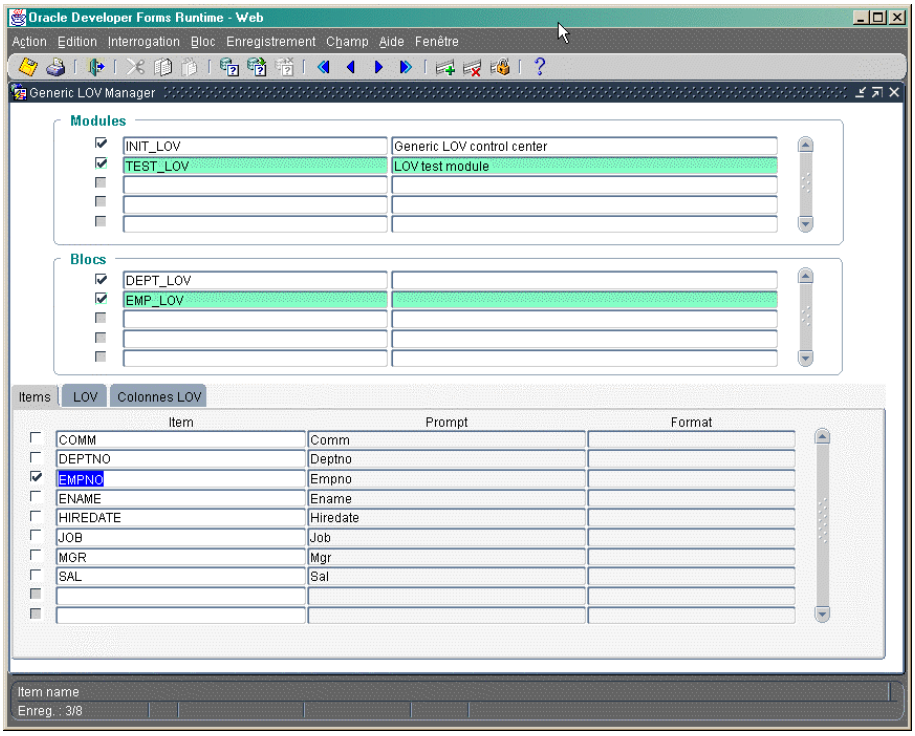
The user preferences can be stored for each LOV for later reuse.

Individual user preferences are stored in database tables that hold a table row for each user and each LOV. The application user is identified by a unique value of type NUMBER. This value is read when loading the first screen and thereafter is sent as a parameter in each subsequent call to a screen (CALL_FORM, OPEN_FORM, NEW_FORM), so you may have to adapt an existing user's table or create it.

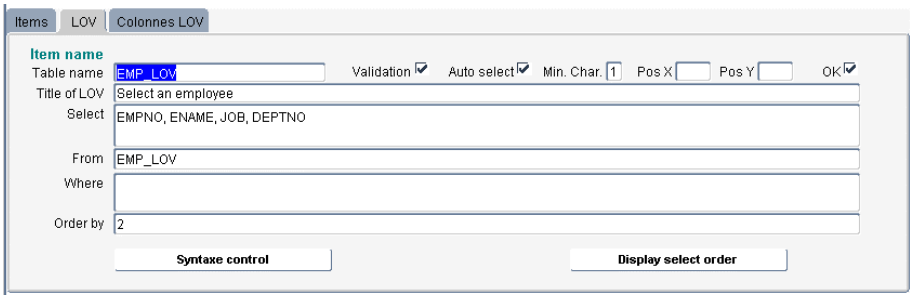
If you do not want to implement the user storage functionality, Open the LOV.pll library and set the GB$Save_Allowed variable of the PKG_VARS package to FALSE.

THE LOV MANAGEMENT

List of values are managed through a screen provided with this solution. The first screen shows module, block and items information for a particular application.

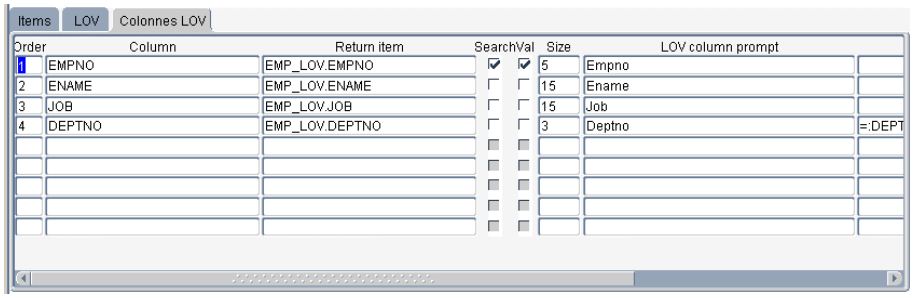


Check-boxes in front of each block notify if at least one Generic LOV is defined on this object.



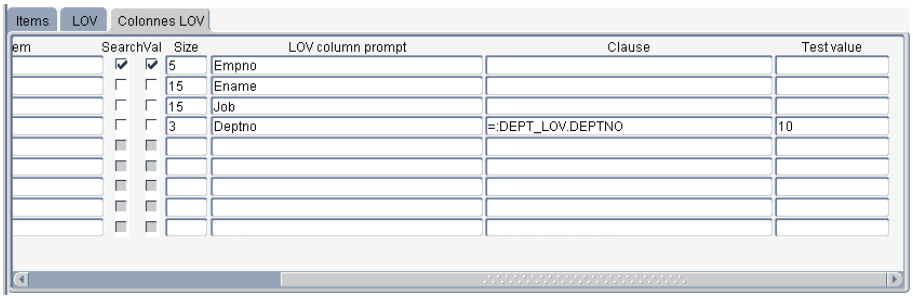
For each item, a LOV, with the following characteristics, can be defined

- Table name used by the global search functionality
- LOV for item validation
- Automatic display of values
- Min number of characters user has to enter
- X and Y positions
- Title of the LOV
- Parts of the query



For each LOV the developer can specify the following attributes

- Display order of columns
- Names of columns
- Names of target items
- The column used for searching
- The column used for validation (with native LOVs only the first column of the query can validate the item)
- Column display width
- Column display prompt
- Restrictive clause for the column that might contain a placeholder
- Test value to simulate the content of the placeholder (only for the validation phase)



Setup instructions

This tool needs tables to store the definition of generic LOVs

1. Objects creation

- Create a special schema to store them (that **INIT_LOV.fmx** screen can see to manage LOVs)
- Open the **/script/create_lov.sql** script file
- Adapt CREATE TABLE orders to suit with your tablespace and storage specifications
- With Sql*Plus execute the script (@c:\your_directory\create_lov)

➢ To grant rights to users execute the **/script/user_grants.sql** script file

➢ To create synonyms to users, connect to users schemes and execute the **/script/user_synonyms.sql** script file.

Caution:

Since the control screen (INIT_LOV.fmx) allows developer to build any LOV on any table, its schema should be able to see every tables/views of the target application.

2. Package

Open the ***PKG_GESTION_LOV*** package specification

There are 2 global variables you can adapt:

- **GN$Duree_timer NUMBER(5) := 800** , which is the delay in milliseconds of expiration timer.

  This delay allows to enter quickly 2 or 3 characters before the timer expire

- GC$Help_Path VARCHAR2(256) := 'http://machine:port/forms90/html' which is the html help files virtual directory you have copied the ***gen_lov.htm*** and ***init_lov.htm*** files and image sub-directories

  (check your httpd.conf Apache configuration file or forms90.conf and orion-web.xml Forms configuration files to know what directories are mapped)

3. Forms modules

Open, compile and generate executable files for the following modules:

- ***COLORS.PLL***
- ***LOV.PLL***
- ***GEN_LOV.FMB***
- ***INIT_LOV.FMB***
- ***TEST_LOV.FMB***

To adapt an existing module, drag the GRP_LOV Object Group from the colors.olb Object Library.

2 template forms are shipped with this package:

- ***FORM_REF_LOV.FMB*** to use generic LOV functionalities
- ***FORM_REF_LOV_COLORS.FMB*** to use both dynamic colors and generic LOV functionalities

In your own forms, attach the ***LOV.PLL*** library and drag the ***GRP_LOV*** from the ***colors.olb*** object library or attach both ***COLORS.PLL*** and ***LOV.PLL*** and drag both the ***GRP_LOV*** and ***GRP_COLORS*** object groups from the **colors.olb** object library if you want to use both dynamic colors and generic LOV

If you do not want to use the dynamic colors utility, you may have to add commentary in front of each line : ***PKG_COLORS.paint*** in the WHEN-NEW-FORM-INSTANCE trigger and suppress KEY-EXEQRY and KEY-ENTQRY form level triggers

Test of the tool

The test form ***TEST_LOV.FMB*** is shipped to test immediately the tool.

There is one Generic LOV defined on the DEPTNO item of departments block, and another one defined on the EMPNO item of employees block.

4. Translation strings

All the strings that could be translated are stored in the ***PKG_VARS*** package's specification of the **lov.pll** library.

It concerns messages displayed in both **GEN_LOV** and **INIT_LOV** Forms modules, and button labels of the **GEN_LOV** module.

5. Html contextual help files

Two html help files are shipped with the tool

- **Init_lov.htm** contains help about the INIT_LOV module (LOVs manager)
- **Gen_lov.htm** contains help about the GEN_LOV module (users)

Copy the **/help/init_lov.htm** and **/help/gen_lov.htm** files and image sub-directories in a directory which is mapped in the ***httpd.conf*** or

Forms90.conf configuration files

These html files are called from the *Display_Help()* procedure of the **LOV.PLL** library. It use the *Web.Show_Document()* Built-in.

For user:

When the Generic LOV screen (***gen_lov.fmx***) is displayed, user can press ***Ctrl+Shift+F1*** key to display the help screen

For developer:

When the Generic LOV configuration LOV screen (***init_lov.fmx***) is displayed, you can get help on topics by press :

- ***Ctrl+Shift+F1*** to get the context help about the item you are
- ***Ctrl+Shift+F2*** to get the context help about the block you are
- ***Ctrl+Shift+F4*** to get the general context help

All theses steps are explained in the install.htm file shipped with the package.

SUMMARY

The solution explained in this paper can be downloaded from the Oracle Forms section on OTN. It allows Oracle Forms developers to centrally manage all the LOVs for an application. The Forms application user can customize the LOV screens.

About the author:

François Degrelle works as a consultant for a French SSII company and is an Oracle specialist (PL/ SQL, Developer, Designer) who likes to share his Forms expertise, writing technical papers about Oracle DB, PLSQL and Forms. François is a native French speaker, as you may be able to tell from some of the variable names used in the source code of this solution. For questions regarding the sample code, please contact François at f.degrelle@free.fr.

Oracle Forms 10*g* – Dynamic Application Color Customization
Seotember 2005
Contributing Authors: Frank Nimphius, Grant Ronald (content review)

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com