

Oracle Forms 10g – Dynamic Color Customization

An Oracle Forms Community White Paper

*François Degrelle
June 2004*

Oracle Forms 10g – Dynamic Color Customization

Introduction.....	3
Color definition table	3
Colors.pll.....	4
Color selection screens	5
PL/ SQL-based dialog.....	6
Java Bean-based dialog.....	7
Changing Forms menu colors.....	8
Forms test module.....	8
Setup Instructions	9
Summary	9

Oracle Forms 10g – Dynamic Color Customization

INTRODUCTION

The solution introduced in this paper allows Oracle Forms application users to customize the colors used in a form. A color selection screen is available for the following Forms items:

- Canvas color
- Tab canvas color
- Prompts (and tooltips) color
- Background buttons color
- Text buttons color
- Background current record color
- Text current record color

Individual user preferences are stored in a database table which holds a table row for each user. 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).

The customized color defined for each item is applied during Forms startup by calling a PLSQL function in the WHEN-NEW-FORM-INSTANCE trigger

The PLSQL function is located in a Forms library, *colors.pll*, and loops through all visible items, setting the colors through visual attributes.

Though this paper focuses on Oracle Forms 10g, the provided PL/SQL code examples also work with earlier releases of Forms. All source examples are available at otn.oracle.com/products/forms.

COLOR DEFINITION TABLE

The table that holds the customized colors for a user must contain at least one row, with COD_UTIL = 0, which contains the default values (RGB format) for users who don't use a customized color scheme.

```

CREATE TABLE UTIL_PREFS
(
  COD_UTIL    NUMBER(5) PRIMARY KEY, -- User code
  C_FOND      VARCHAR2(20 BYTE),      -- color of canvas
  C_LIBELLE   VARCHAR2(20 BYTE),      -- color of prompts
  C_CADRE     VARCHAR2(20 BYTE),      -- color of tab canvas
  C_BOUTON    VARCHAR2(20 BYTE),      -- color of background buttons
  C_TBOUTON   VARCHAR2(20 BYTE),      -- color of text buttons
  C_CURREC    VARCHAR2(20 BYTE),      -- color of background current
  C_TCURREC   VARCHAR2(20 BYTE)       -- color of text current
) ;

INSERT INTO UTIL_PREFS ( COD_UTIL, C_FOND, C_LIBELLE,
C_CADRE, C_BOUTON, C_TBOUTON, C_CURREC,C_TCURREC) VALUES (
0, 'r128g128b192', 'r255g255b64', 'r64g128b255', 'r255g192b255',
'r64g64b192', 'r128g255b255', 'r128g128b255');

```

A Forms color selection dialog is provided in the sample code of this solution and handles the user specific table inserts and updates.

COLORS.PLL

The colors.pll library contains the PL/ SQL code to customize the color of visual objects in Forms at startup, at enter query (which color all queryable items with the color specified in the VA_QUERY visual attribute), and after execute query to recall previous colors.

The colors.pll library needs to be attached to the Forms modules used in a Forms application. Copy the *GRP_COLORS* group from the object library *colors.olb*.

The *GRP_COLORS* group contains the following objects that need to be added to the Forms module:

- A form level trigger WHEN-NEW-FORM-INSTANCE for the colorization function call
- A form level trigger KEY-ENTQRY for the colorization in ENTER-QUERY mode
- A form level trigger KEY-EXEQRY to re-set the initial colors after execute query
- 3 alert boxes
- A parameter *UTI_ID* for handling the ID that uniquely identifies the user
- Visual attributes for colorization of objects

The color.pll library contains 2 PLSQL procedures and 1 PLSQL package:

- **Start_query()** - procedure to color queryable items in ENTER-QUERY mode

- **End_query()** - procedure to re-set the initial colors after EXECUTE-QUERY
- **PKG_COLORS package**
 - Global variables for the colors read from the UTIL_PREFS table
 - Global variables to translate Strings used in the COLORS.FMB and COLORS_J.FMB screen
 - Procedure “Paint”, which is the main procedure called during the Forms instance startup
 - Procedure Set_colors

Items that are not navigable are colored with a light gray background. All mandatory items are indicated by a bold prompt. You may want to customize the PLSQL package to use your preferred indicators.

To preserve the initial colors of items, start the TOOLTIP_TEXT with the NCC (No Change Color) characters. These 3 characters are removed when the program is executed to insure a correct tooltip display

COLOR SELECTION SCREENS

Two Forms screens are provided with this whitepaper, as well as samples that allow the user to customize the application color for that instance:

- **PL/SQL based dialog (COLORS.FMB)** - This screen uses native Forms code to render the color selection dialog.
- **Java Bean based dialog (COLORS_J.FMB)** – This screen uses the Color Picker Java Bean sample from the Oracle Forms demos. Using the Java Bean does allows the user to choose from more colors than the PL/SQL based screen, but requires the bean JAR files to be deployed with the application.

All screens can be accessed through a menu option or buttons. All that needs to be done is to pass the user code in the UTI_ID parameter. Never transmit the 0 code which is the pre-populated default code

PL/SQL-based dialog



Figure 1: Colors.fmb

To customize the color of a visual component in Forms, click the corresponding button of the object to change the color of and click on one of the 64 color cells. Save the screen (Ctrl+S) to store the new colors or just exit (F4) to cancel any modification

Java Bean-based dialog

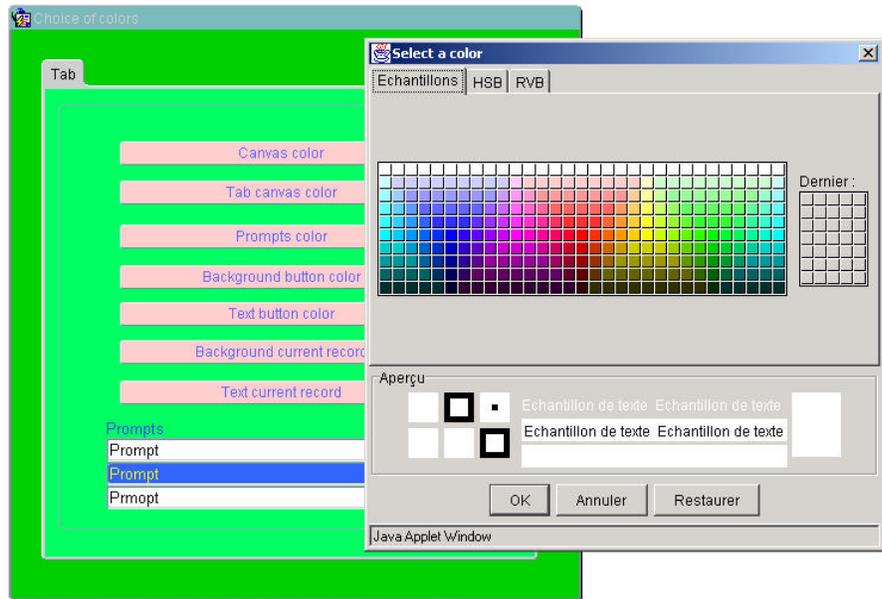


Figure 2: Colors_J.fmb, requiring colorpicker.jar

To customize the color of a visual component in Forms, click the corresponding button of the object to change the color, then select the color from the Bean selection dialog. Save the screen (Ctrl+S) to store the new colors or just exit (F4) to cancel any modification.

In order to run the screen with the Java Bean component, you need to download the colorpicker.jar file, which is a part of the Oracle Forms 10g demos available on the Oracle Technology Network (OTN)¹

- To configure the Bean, copy the file colorpicker.jar into the <ORACLE_HOME>/forms90/java directory
- Edit the Forms formsweb.cfg configuration file located in <ORACLE_HOME>/forms90/server/ and add the colorpicker.jar file to the JInitiator archive tag

```
[myColorfulApp]
form = myForm.fmx
...
archive_jini=f90all_jinit.jar,colorpicker.jar
...
```

¹ http://download.oracle.com/otn/other/general/forms10gdemos9_0_4_2.zip

CHANGING FORMS MENU COLORS

The Forms menu module (MENU_COLORS.MMB), which is part of the associated example code, either calls the PLSQL based color selection screen, or the Java Beans-based screen.

FORMS TEST MODULE

The code sources for the solution introduced in this paper also contain a Forms test module for you to try.



Figure 3: Test Forms

The Test Forms module demonstrates the following functionality:

- Mandatory items have a bold prompt
- Display items have a grey background
- Items for which you don't want to alter the color remain as defined by the design value (because the first three characters of the tooltip text are NCC)

Setup Instructions

1. Install the Forms sources, into a directory which is contained in the FORMS90_PATH path
2. To use the JavaBean color screen:

Copy the file colorpicker.jar into your
<ORACLE_HOME>/forms90/java server directory

Edit the formsweb.cfg configuration file in the <ORACLE_HOME>
/forms90/server/ directory and add

```
archive_jini=f90all_jinit.jar,colorpicker.jar
```

to the application configuration that uses this solution.

3. Create the UTIL_PREFS table in your own scheme and insert the default row
4. Launch the Forms Builder, log into the scheme in which the UTIL_PREFS table was created, and open the source files
5. Compile and generate executables for all sources files (colors.pll, colors.fmb, colors_j.fmb, test_colors.fmb, menu_colors.mmb)

To use this solution in your own Forms modules:

- Open your Form (use the template form)
- Attach the colors.pll library
- Open the colors.olb object library
- Drag the grp_colors group in the object group of your form
- Compile all and generate

SUMMARY

The solution described in this whitepaper allows Forms application users to apply a customized color scheme to their applications. The PL/SQL and Forms source codes shipped with the paper are free to be used in your own Forms applications.

About Forms Community Whitepapers

Oracle Forms Community White Papers is a new section on OTN and provides a forum for Oracle experts to share their product expertise with others by writing technical articles about Oracle Forms.

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. Francois 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 Francois at f.degrelle@free.fr.



Oracle Forms 10g – Dynamic Application Color Customization
June 2004

Author: François Degrelle, f.degrelle@free.fr
Contributing Authors: Frank Nimphius, Oracle

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

Copyright © 2004, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.