

# Oracle Forms Recipes

Oracle Forms Runtime - [Hospital Software]

Masters Registration Finance Investigation Pharmacy Blood-Bank Warehouse Human-Resource Reports Window

UM Dr. OPI WD EM RB. NR DL IPH ID. CG CS CH RC RI CC C3 C4 AC M. RV Pd Cr. Ch Cp PC LB Cs Ctr Mg Ms PT XT UT MT CT MC Mc Mt MI SM MC CM CI

Find Patient:  Hospital Management System - Your Name

### I.P. Registration

I.P. No.  Add New Date 18-11-2010

C.R. No.  Name:

Empanel ID:  Empanel Name:

Ward  Bed No. 1 Category: E

### Bed and Discharge Details

Ward	Bed No.	Date Switched
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>

Save Print Exit

User: HMS

### Discharge Details

Dt. Discharge

Remark

Diagnosis

ICD Code:

Dr. Name

Update Chart

Department	IPNO
ENT	13
GENERAL	1
GYNAE/OBST.	51
MEDICINE	167
NICU	9
OPHTHALMOLOGY	9
ORTHOPAEDIC	38
PAEDIATRICS	36
PSYCHIATRY	18
SURGERY	71
TBC	19
<b>Total</b>	<b>432</b>

By Vinish Kapoor  
Developing Oracle Applications since 2001

## Contents

Map Columns From Different Tables and Create Insert and Update Statements in Oracle Forms .....	3
Checking For User Permissions Before Updating or Inserting The Records in Oracle Forms .....	5
An Example of Pre-Query and Post-Query Triggers in Oracle Forms With Using Display_Item to Highlight Dynamically.....	7
Displaying Modal Window Messages in Oracle Forms Using Alerts .....	9
Get_File_Name Usage in Oracle Forms 6i .....	11
Creating, Stopping, Re-Starting and Deleting a Timer in Oracle Forms.....	13
Writing Text File From A Tabular Block In Oracle Forms .....	16
Populating Tabular Data Block Manually Using Cursor in Oracle Forms .....	18
If Value Exists Then Query Else Allow Create New in Oracle Forms An Example .....	21
Trigger Execution Sequence Of Oracle Forms.....	23
Why And When To Use Pre-Update and Pre-Insert Triggers In Oracle Forms.....	25
Some Useful Property Settings Explained Of Oracle Forms .....	27
FRM-10001, FRM-10002, FRM-10003 Oracle Form Builder Error Solution .....	30
Example of Get_File_Name Function in Oracle Forms.....	31
Create Timer Example To Show Image Presentation in Oracle Forms .....	33
See also: Create timer to display clock in Oracle Forms .....	<b>Error! Bookmark not defined.</b>
Moving From Top To Bottom in Detailed Block in Oracle Forms.....	35
Date Picker Calendar For Oracle Forms 6i .....	36
Download Form and required PLL .....	36
How To Use .....	36
Determining Current Block and Current Item in Oracle Forms .....	38
SYSTEM.CURSOR_BLOCK .....	38
SYSTEM.CURSOR_ITEM.....	38
View Oracle Developer Handbook at Amazon.com .....	39
Adding Value To Combo List at Runtime in Oracle Forms .....	40
Using GET_GROUP_SELECTION For Record Groups in Oracle Forms.....	42
Oracle Form Data Entry Sample .....	44
Using User-Named Triggers in Oracle Forms .....	52
Giving Data Backup Option in Oracle Forms 6i .....	53
An Example of On-Error Trigger in Oracle Forms.....	55
Shifting List Item Values From One List To Another In Oracle Forms .....	58
Using SYSTEM.MOUSE_ITEM In Oracle Forms.....	60
Using Post_Query Trigger in Oracle Forms .....	62

Pre-Update and Pre-Insert Trigger Examples For Oracle Forms .....	64
Pre-Insert trigger .....	65
Handling Tab Pages in Oracle Forms .....	66
How To PLAY_SOUND in Oracle Forms.....	68
Using Find_Alert and Show_Alert in Oracle Forms .....	69
How To Use RUN_PRODUCT In Oracle Forms.....	71
How To Use FETCH_RECORDS In Oracle Forms .....	73
Writing On-Error Trigger In Oracle Forms .....	76
How To Use DBLink In Oracle Forms 6i.....	78
Solution - 1 .....	78
Solution - 2 .....	78
Highlighting Text Item On Entry In Oracle Forms.....	79
Steps to highlight current item.....	79
Changing Icon File Of Push Button At Runtime In Oracle Forms 6i.....	81
Set Icon_File property in When-Mouse-Enter trigger .....	81
Displaying Window In Center In Oracle Forms 6i.....	83
Populating Tree Item With Record Group In Oracle Forms .....	84
Formatting Excel File Using Ole2 In Oracle Forms .....	86
Adding List Item Element At Runtime In Oracle Forms .....	88
Populating Display Item Value On Query In Oracle.....	89
Creating Custom Login Screen In Oracle Forms 10g .....	90
Writing Text Files On The Client in Oracle Forms 10g .....	92
Reading An Image File Into Forms From Client In Oracle Forms 10g .....	94
Number To Indian Rupee Words in Oracle Forms / Reports .....	95
Creating Object Library OLB in Oracle D2k Form .....	98
Creating Excel File in Oracle Forms .....	101
How To Tune or Test PLSQL Code Performance in Oracle D2k Forms .....	105
Creating Dynamic LOV in Oracle D2k Forms .....	107
Using GET_APPLICATION_PROPERTY in Oracle D2k Forms .....	112
Upload Files To FTP in Oracle Forms D2k .....	114

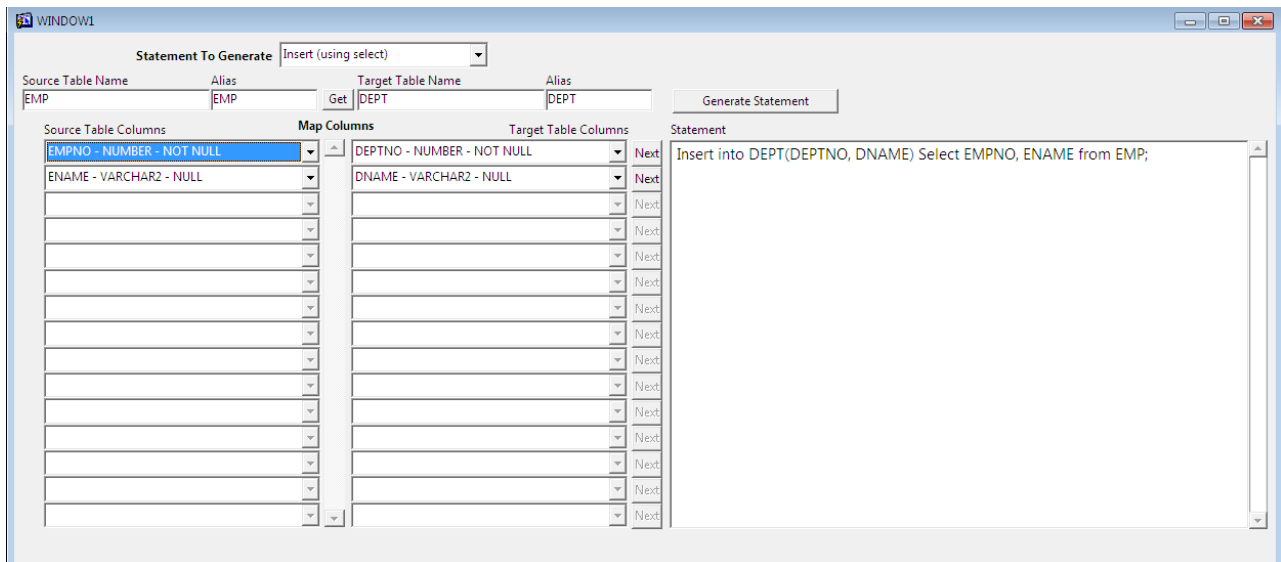
## Map Columns From Different Tables and Create Insert and Update Statements in Oracle Forms

This is one of my most needed tool to create Insert and Update statements using select or alias from different tables where column mapping is difficult.

What this utility actually does is, suppose you have two table with too many columns and you want to update or insert in one table from another, you know column names but you want some visual interface so that your task can be easier. This tool provide the same facility to create insert or update statement by visually mapping fields from two tables.

The utility is created using Oracle Forms and can be used easily. Free download it from [Tablemap.fmx](http://Tablemap.fmx)

Below is the screen shots of this tool:



WINDOW1

Statement To Generate: Insert (using alias)

Source Table Name: EMP    Alias: EMP    Target Table Name: DEPT    Alias: DEPT    Generate Statement

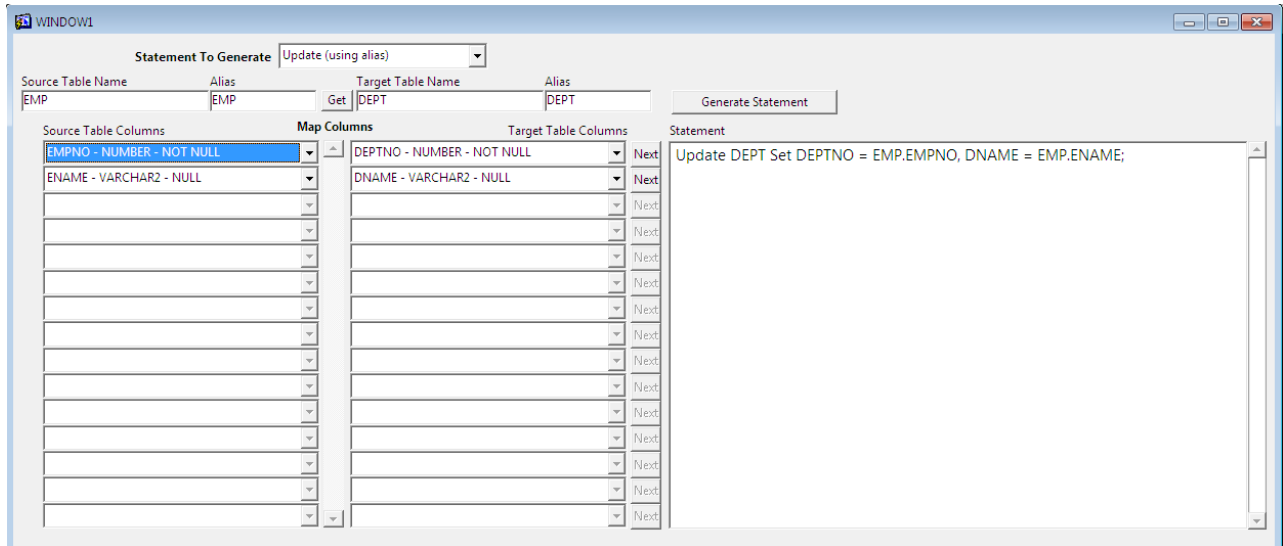
Source Table Columns	Map Columns	Target Table Columns	Statement
EMPNO - NUMBER - NOT NULL		DEPTNO - NUMBER - NOT NULL	Insert into DEPT(DEPTNO, DNAME) Values (EMP.EMPNO, EMP.ENAME);
ENAME - VARCHAR2 - NULL		DNAME - VARCHAR2 - NULL	

WINDOW1

Statement To Generate: Update (using select)

Source Table Name: EMP    Alias: EMP    Target Table Name: DEPT    Alias: DEPT    Generate Statement

Source Table Columns	Map Columns	Target Table Columns	Statement
EMPNO - NUMBER - NOT NULL		DEPTNO - NUMBER - NOT NULL	Update DEPT Set (DEPTNO, DNAME) = (Select EMPNO, ENAME from EMP);
ENAME - VARCHAR2 - NULL		DNAME - VARCHAR2 - NULL	



## Checking For User Permissions Before Updating or Inserting The Records in Oracle Forms

Suppose you want to check the user permissions on inserting or updating the records in Oracle Forms, then you can use [Pre-Insert and Pre-Update](#) triggers for that particular data block to check whether user is having proper [permission](#) or not.

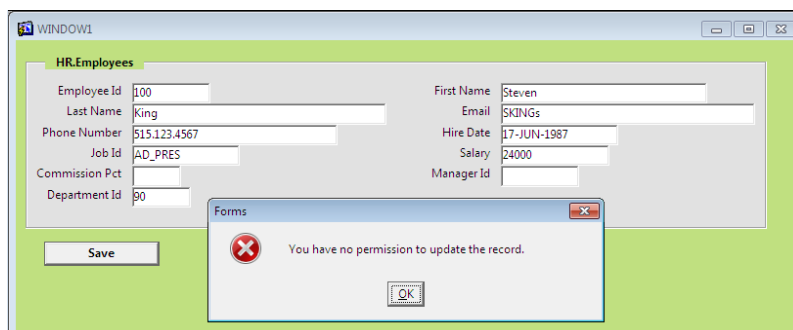
The example is given for HR schema and the following demo table is created for this:

Create Table User\_Permissions (User\_Name varchar2(50) Primary Key,  
Can\_Insert Varchar2(1) default 'N',  
Can\_Update Varchar2(1) default 'N')

Insert into User\_Permissions values ('FORMUSER1', 'Y', 'N');

Commit;

Below is the screen shot for the example:



You can download this form from the following link: [Pre-Update-Insert.fmb](#)

The following is the code written in Pre-Update trigger to check the permissions from the database:

```
Declare
n_allow number;
begin
Select 1 into n_allow
  from hr.user_permissions
  where user_name = 'FORMUSER1'
  and can_update = 'Y';
  --- all is well if no exception raised else stop in exception area

exception
  when others then
    message('You have no permission to update the record.');
```

The following is the code written in Pre-Insert trigger to check the permissions from the database on Insert:

```
Declare
n_allow number;
begin
Select 1 into n_allow
  from hr.user_permissions
  where user_name = 'FORMUSER1'
  and can_insert = 'Y';
  --- all is well if no exception raised else stop in exception area

exception
  when others then
    message('You have no permission to insert the record.');
```

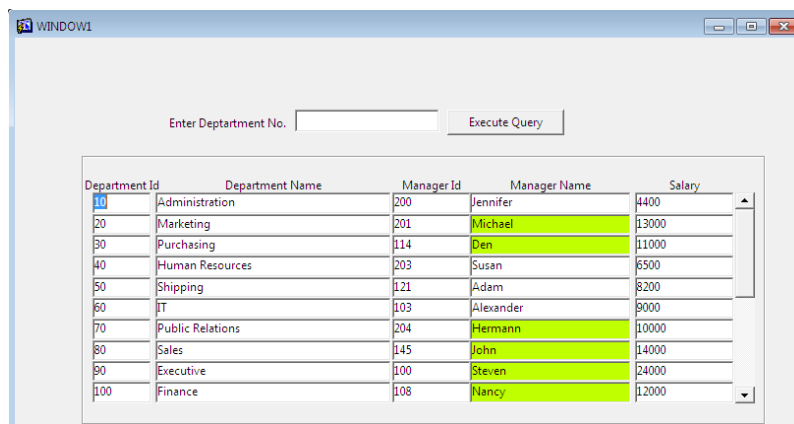
The code is written for Save button:

```
Commit_form;
```

## An Example of Pre-Query and Post-Query Triggers in Oracle Forms With Using Display\_Item to Highlight Dynamically

Example is given for [Pre-Query](#) and [Post-Query](#) triggers in Oracle Forms, with using [Display\\_Item](#) built-in in Post-Query triggers to highlight fields dynamically.

This is the screen shot below for this example:



Department Id	Department Name	Manager Id	Manager Name	Salary
10	Administration	200	Jennifer	4400
20	Marketing	201	Michael	13000
30	Purchasing	114	Den	11000
40	Human Resources	203	Susan	6500
50	Shipping	121	Adam	8200
60	IT	103	Alexander	9000
70	Public Relations	204	Hermann	10000
80	Sales	145	John	14000
90	Executive	100	Steven	24000
100	Finance	108	Nancy	12000

You can also download this form from the following link: [Query.fmb](#)

The example is based on HR schema departments table. In this example Department No. and Execute Query push button is in upper block named "Ctrl" block and below block is the departments block. User will be asked to enter department no. in above



block and then to click on Execute Query button to filter the records below. The filtration is handled in Pre-Query trigger and after execution of query the Manager Name and Salary will be populated in Post-Query trigger by dynamically highlighting the Manager Name using Display\_Item built-in.

The following code written in Pre-Query trigger of Departments block to filter the records:

```
if :ctrl.deptno is not null then
  -- set default_where property of the block to the ctrl block item to filter the records
  before query
  set_block_property('departments', default_where, 'department_id = :ctrl.deptno');
end if;
```

The following code written in Post-Query trigger of Departments block to populate non-database item fields and dynamically highlighting the Manager Name field:

```
begin
  select first_name, salary into :departments.empname, :departments.sal
  from hr.employees where employee_id = :departments.manager_id;
  -- highlight as per your criteria
  if :sal >= 10000 then
    -- create highlight visual attribute with color of your choice to highlight
    display_item('departments.empname', 'highlight');
  else
    -- create default1 visual attribute to restore to normal view
    display_item('departments.empname', 'default1');
  end if;
exception
  when others then
    null;
end;
```

The following code written in When-Button-Pressed trigger of Execute Query push button in Ctrl block to execute query in Departments block:

```
go_block('departments');
set_block_property('departments', default_where, "");
execute_query;
```

## Displaying Modal Window Messages in Oracle Forms Using Alerts

You can display modal windows in Oracle Forms to display normal messages, [error message](#) or asking for confirmation eg. on deleting a record or saving a record etc.

These modal window messages can be shown using [Alert option in Oracle forms](#).

This is the screen shot below for this example:



You can download this form from the following link: [Modal Msgt.fmb](#)

For this example I have created three alerts with the following names:

1. Good\_Msg

2. Error\_Msg
3. Ask\_Alert

The following code is written for "Show Good Message" button to display a normal message, you can use this code in any PLSQL block:

```
Declare
-- create a numeric variable to hold show_alert return value
nalertbutton number;
Begin
-- set the message for alert
set_alert_property('good_msg', alert_message_text, 'Records saved successfully. ');
-- after below statement the execution will hold till you click on ok.. because it is an modal
window
nalertbutton := show_alert('good_msg');
:alertblock.result := 'That was a good message.';
-- after this you can perform any task...
End;
```

The following code is written for "Show Error Message" button to display an Error message:

```
Declare
-- create a numeric variable to hold show_alert return value
nalertbutton number;
Begin
-- set the message for alert
set_alert_property('error_msg', alert_message_text, 'An error occurred. ');
-- after below statement the execution will hold till you click on ok.. because it is an modal
window
nalertbutton := show_alert('error_msg');
:alertblock.result := 'That was an ERROR message.';
-- after this you can perform any task...
End;
```

The following code is written for "Ask Confirmation" button to ask for a confirmation:

```
Declare
-- create a numeric variable to hold show_alert return value
nalertbutton number;
Begin
-- set the message for alert
set_alert_property('ask_alert', alert_message_text, 'Confirm Yes or No? ');
-- after below statement the execution will hold till you click on ok.. because it is an
modal window
```

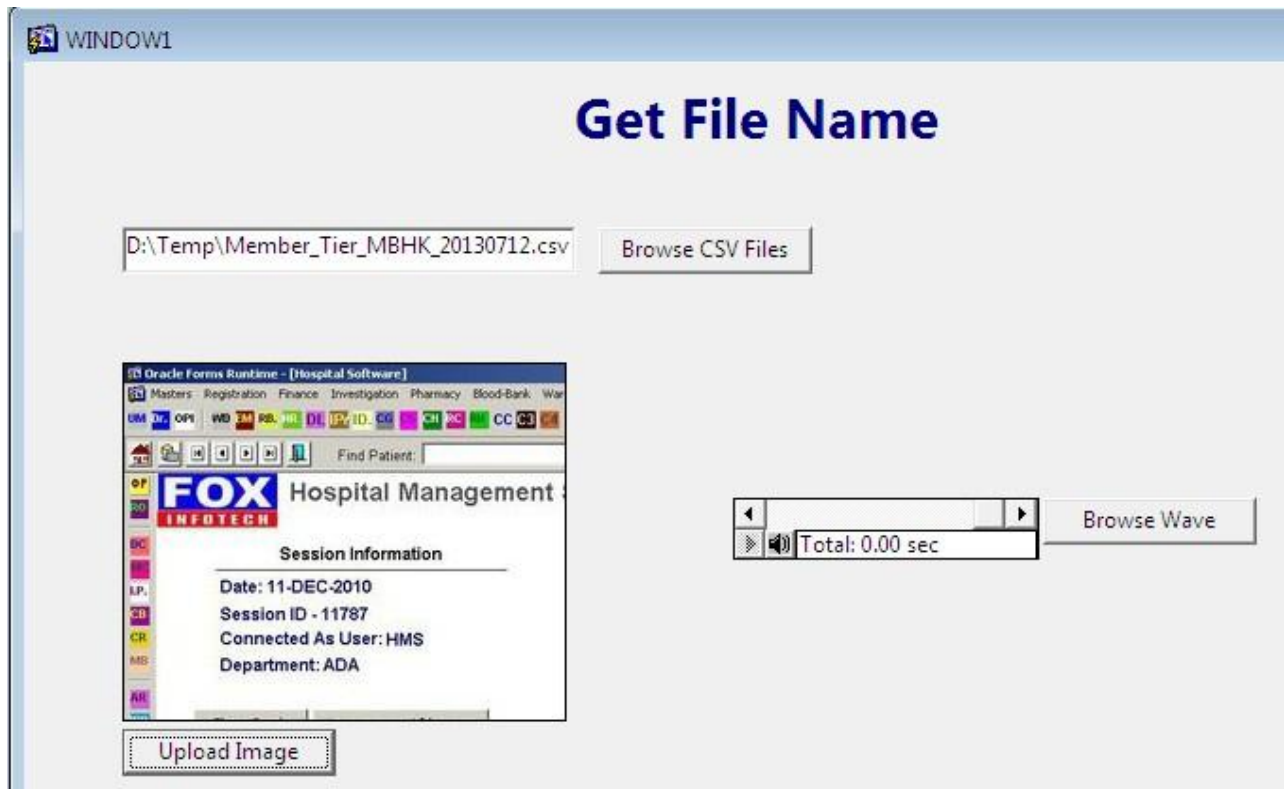
```
nalertbutton := show_alert('ask_alert');
-- now check which button or answer have been chosen
if nalertbutton = alert_button1 then
:alertblock.result := 'You choose Yes.';
else
:alertblock.result := 'You choose No.';
end if;
-- after this you can perform any task...
End;
```

## Get\_File\_Name Usage in Oracle Forms 6i

[Get\\_File\\_Name is built-in function](#) of Oracle Forms 6i, used to get the file name with address by browsing the file. You can browse a specific extension name file or with multiple extensions using wild cards.

In this example I am showing three different usage of Get\_File\_Name function, one is given for to get any [Csv file](#) name into a display field, second is given to [read image files](#) into an image item and third is for [sound item](#).

Below is the screen shot for this example:



You can also download this form from the following link: [GetFileName.fmb](#)

Write the following code on When-Button-Pressed trigger for Browse CSV Files button:

```
declare
  filename varchar2(500);
begin
  filename := GET_FILE_NAME(File_Filter=> 'CSV Files
(*.Csv)|*.Csv|');

:block2.txtfile := filename;
end;
```

Write the following code on When-Button-Pressed trigger for Upload Image button:

```
declare
  filename varchar2(500);
begin
  filename := GET_FILE_NAME(File_Filter=> 'Jpg Files
(*.jpg)|*.jpg|Gif Files (*.gif)|*.gif|All Files (*.*)|*.*)|');
  READ_IMAGE_FILE(filename, 'JPEG', 'block2.image9');
```

```
end;
```

Write the following code on When-Button-Pressed trigger for Browse Wave button:

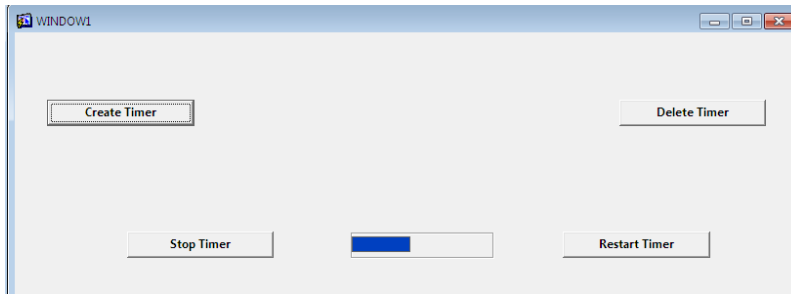
```
declare
  filename varchar2(500);
begin
  filename := GET_FILE_NAME(File_Filter=> 'Wav Files
(*.wav)|*.wav|All Files (*.*)|*.*|');

  READ_sound_FILE(filename, 'wave', 'block2.sound_item11');
  play_sound('block2.sound_item11');
end;
```

## Creating, Stopping, Re-Starting and Deleting a Timer in Oracle Forms

I have written many posts previously on Timers in Oracle Forms like [how to change images randomly with timers](#) and [how to display a clock using timer](#), but in this post I am simply describing to how to create a timer, stop a timer, re-start a timer and deleting a timer.

The following is the screen shot for this example showing a progress bar based on a display item:



You can also download this form from the following link [Timer.fmb](#)

Create a display item with the following properties:

Name: Prgbar  
 Width: 5  
 Bevel: Plain  
 Background Color: blue

Write the following code for the "Create Timer" button:

When-Button-Pressed trigger

```

Declare
  v_timer timer;
Begin
  -- find timer first if already exists.
  v_timer := find_timer('PrgBarTmr');
  if id_null(v_timer) then
    -- Creating timer for one second... one second = 1000 millisecond
    v_timer := Create_Timer('PrgBarTmr', 1000, Repeat);
  else
    message('already exists.');
```

-- will handle this timer in form level when-timer-expired trigger

```

End;
```

Write the following code for the "Stop Timer" button:

When-Button-Pressed trigger

```

Declare
  v_timer timer;
Begin
  -- find the timer first
  v_timer := find_timer('PrgBarTmr');
  if not id_null(v_timer) then
    -- this will stop the timer after one millisecond
    Set_Timer(v_timer, 1, No_Repeat);
```

```
end if;
-- will handle this timer in form level when-timer-expired trigger
End;
```

Write the following code for the "Re-Start Timer" buton:

When-Button-Pressed trigger

```
Declare
v_timer timer;
Begin
-- find the timer first
v_timer := find_timer('prgbartmr');
if not id_null(v_timer) then
-- this will re-start the timer after one second
Set_Timer(v_timer, 1000, Repeat);
else
v_timer := create_timer('prgbartmr',1000, Repeat);
end if;
-- will handle this timer in form level when-timer-expired trigger
End;
```

Write the following code for the "Delete Timer" buton:

When-Button-Pressed trigger

```
Declare
v_timer timer;
Begin
-- find the timer first
v_timer := find_timer('PrgBarTmr');
if not id_null(v_timer) then
-- this will delete the timer
Delete_Timer(v_timer);
end if;
End;
```

Then finally write the code for When-Timer-Expired trigger at form level to handle the timer and to do specific task:

When-Timer-Expired trigger

```
Declare
v_timer_name varchar2(30);
v_width number;
Begin
-- get the timer name first.. to know which timer has expired.. if multiple timer are
running
v_timer_name := get_application_property(timer_name);
```



```
-- check if the same timer with capital letters
if v_timer_name = 'PRGBARTMR' then
  v_width := get_item_property('blKtmr.prgbar', width);
  if v_width < 100 then
    v_width := v_width + 5;
  else
    v_width := 0;
  end if;
  set_item_property('blktmr.prgbar', width, v_width);
end if;

-- will handle this timer in form level when-timer-expired trigger
End;
```

## Writing Text File From A Tabular Block In Oracle Forms

The example given below for writing text file or [CSV](#) using [Text\\_IO](#) package from a tabular block in Oracle Forms.

## Thank You for previewing this eBook

You can read the full version of this eBook in different formats:

- HTML (Free /Available to everyone)
- PDF / TXT (Available to V.I.P. members. Free Standard members can access up to 5 PDF/TXT eBooks per month each month)
- Epub & Mobipocket (Exclusive to V.I.P. members)

To download this full book, simply select the format you desire below

