

## Oracle Control Files

### Backup and Recovery Tips

Helpful statement to end the backup for the tablespaces which are in the online backup mode:

1- Create a script through the following commands:

```
svrmgrl> spool endhotbackup.sql  
svrmgrl> select 'ALTER DATABASE DATAFILE '''||name||''' END BACKUP ' FROM V$DATAFILE;
```

2- Cleanup endhotbackup.sql through vi

3- svrmgrl> @endhotbackup.sql

4- startup the database

## Oracle Forms

### Oracle Forms 4.5 FAQ

#### Can on bypass the Oracle login screen?

The first thing that the user sees when using runform is the Oracle logon prompt asking them for their username, password, and database to connect to. You can bypass this screen or customise it by displaying your own logon screen. Eg:

#### ON-LOGIN

```
declare  
    uname varchar2(10);  
    pass varchar2(10);  
begin  
    uname := 'username';  
    pass := 'password';  
    logon(uname, pass||'@connect_database');  
end;
```

---

#### Can one issue DDL statements from Forms?

DDL (Data Definition Language) commands like CREATE, DROP and ALTER are not directly supported from Forms because your Form is not suppose to manipulate the database structure.

A statement like CREATE TABLE X (A DATE); will result in error:

*Encountered the symbol "CREATE" which is an reserved word.*  
However, you can use the FORMS\_DDL built-in to execute DLL statements. Eg:  
FORMS\_DDL('CREATE TABLE X (A DATE)');

---

#### Can one execute dynamic SQL from Forms?

Yes, use the FORMS\_DDL built-in or call the DBMS\_SQL database package from Forms. Eg:

```
FORMS_DDL('INSERT INTO X VALUES (' || col_list || ')');
```

Just note that FORMS\_DDL will force an implicit COMMIT and de-synchronize Oracle Forms.

---

### **Forms won't allow me to use restricted built-in's. What should I do?**

How to get around the "can't use a restricted built-in in built-in XXX" message:

1. Create a TIMER at the point where you want the navigation to occur. Eg. `create_timer('TIMER_X', 5, NO_REPEAT);`
2. Code a WHEN-TIMER-EXPIRED trigger to handle the navigation

```
DECLARE
  tm_name VARCHAR2(20);
BEGIN
  tm_name := Get_Application_Property(TIMER_NAME);
  IF tm_name = 'TIMER_X' THEN
    Go_Item('ITEM_X');
  END IF;
END;
```

Dirty but effective (didn't Oracle promise to fix this feature?).

---

### **Can one change the mouse pointer in Forms?**

The SET\_APPLICATION\_PROPERTY build-in in Oracle Forms allow one to change the mouse pointer. Eg:

```
SET_APPLICATION_PROPERTY(CURSOR_STYLE, BUSY);
```

---

### **Why doesn't my messages show on the screen?**

Regardless of whether you call the MESSAGE() built-in with ACKNOWLEDGE, NO\_ACKNOWLEDGE, or with no mode specification at all, your message may or may not be displayed. This is because messages are displayed asynchronously. To display messages immediately, use the SYNCHRONIZE build-in:

```
message('...'); synchronize;
```

This can also be used to execute a query while the user is looking at the results of a previous query.

---

### **What happened to SQL\*Menu?**

SQL\*Menu is now fully integrated into Oracle Forms 4.5. Application menus can be added to your application by creating Menu Modules (\*.MMB) and generate it to Menu Module Executables (\*.MMX).

### **TEXT\_IO code to write data to a flat file**

```
PROCEDURE CREATE_FLAT_FILE IS
-- DESCRIPTION
-- Simple procedure to write to a text file from a form
-- NOTES
-- Call this procedure from the WHEN-BUTTON-PRESSED trigger
-- REQUIREMENTS
```

-- Must have access to local drives

--

--

=====

=====

```
in_file TEXT_IO.FILE_TYPE;
linebuf VARCHAR2(80);
your_file VARCHAR2(50) := 'C:\TEMP\your_file.dat';
text_line VARCHAR2(400);
BEGIN
    in_file := TEXT_IO.FOPEN(your_file, 'W');
    GO_BLOCK('YOUR_DATA_BLOCK');
    FIRST_RECORD;
    LOOP
    IF :your_checkbox = 'Y' THEN
        text_line := LPAD(:your_item1, 10, ' ')||
            LPAD(:your_item2, 20, ' ')||
            LPAD(:your_item3, 30, ' ')||
            LPAD(:your_item4, 50, ' ');
        --
        TEXT_IO.PUT_LINE(in_file, text_line);
    END IF;
    --
    exit when :SYSTEM.LAST_RECORD = 'TRUE';
    NEXT_RECORD;
    END LOOP;
    --
    TEXT_IO.FCLOSE(in_file);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        TEXT_IO.FCLOSE(in_file);
END;
```