



Upgrade Developer Forms 4.5 to Oracle Forms 6

An Oracle Technical White Paper
March 2000

WHY UPGRADE?

In today's demanding business world, operational systems are relied upon more than ever. User communities are growing in number, and they are demanding higher levels of application performance, reliability, and availability. Business environments are more dynamic than ever before, requiring applications that adapt to changing business requirements in shorter timeframes.

Oracle Forms Builder and Oracle Forms Server provide a scalable, flexible architecture that automatically delivers high-performance, enterprise-class applications to all of your users. Today, Oracle customers are using Oracle Forms Server to support hundreds of users around the world. Benchmarks have proven exceptional performance by running thousands of concurrent users. In fact, Oracle's ERP business suite, Oracle Applications Release 11, supports thousands of users around the world using the Oracle Forms Server technology. Oracle Business OnLine, a hosting service developed by Oracle for mid-sized markets, is also powered by Oracle Forms Server.

Oracle Forms has responded to technological advances, enabling customers to preserve their investment and their business logic while exploiting the benefits of each technological wave. The underlying platform and services built inside the deployment engine itself are the keys that allows you to readily exploit the advantages of each new technology. It is critical for an organization to keep its applications running on current technology in order to exploit the benefits provided by Oracle Forms.

Upgrading your existing applications will give you great benefit in enabling your applications to run over the web, as well as giving you the capability to incorporate new features into your applications to provide the best possible environment for your customers. The goal of this paper is to help you upgrade your existing Oracle Developer Forms 4.5 applications smoothly and easily to Oracle Forms Release 6.

ORACLE APPLICATIONS MANUFACTURING AND FINANCIALS FORMS UPGRADE

Oracle is a company that believes in its own technology. Oracle's ERP business suite, Oracle Applications Release 11i, are written with the Oracle Forms product line. These applications were written using Oracle Developer Forms 4.5. With the next major Release of Oracle's ERP Applications, Release 11i, all Forms have been upgraded to Oracle Forms 6.

By upgrading Applications Release 11i to the Oracle Forms Release 6 technology stack, Oracle is able to offer business suites that run on the web. Internet computing has replaced client/server computing because it's simpler, cheaper, and provides better information. Oracle has the only e-business suite that runs on corporate intranets and the World Wide Web. With its third-generation internet release, Oracle has hundreds of customers who are running Oracle's 100% Internet Applications in production today!

In order to ensure that the upgrade process for Oracle Forms Release 6 would be smooth, Oracle used its Manufacturing and Financial Applications as test cases. This upgrade consisted of 1,178 Forms, 359 Libraries, and 1 Menu. Due to the ease of use and functionality provided by the Upgrade facilities provided by Oracle Forms Release 6, it took only two man weeks of work in order to upgrade all of these modules. Most modules required no manual intervention for the upgrade. 77% of the Form modules, 78% of the Library modules and 100% of the Menu modules required no manual changes.

Let's first take a look at what the Upgrader will do to a module. Once we are aware of the things that can be upgraded automatically, we will take a look at all the manual changes required on the 12% - 13% modules that were not 100% automatically upgraded.

UPGRADING PL/SQL

Oracle Forms Release 6 uses PL/SQL Version 8. When you upgrade your Forms, Reports and Libraries, all your PL/SQL code will have to be upgraded. You must be aware of the following changes in PL/SQL. Please be sure to upgrade your libraries first, as your forms and reports may reference them. Upgrading the PL/SQL is optional when upgrading your Oracle Forms module, however your modules will not generate into a runtime until the PL/SQL upgrade is complete.

- All character variables must be VARCHAR2 and must have a length. The following demonstrates how character variable definitions will be converted by default.

CHAR(N)	Becomes..	VARCHAR2(N)
CHAR	Becomes..	VARCHAR2(N)
VARCHAR	Becomes..	VARCHAR2(N)
VARCHAR2	Becomes..	VARCHAR2(N)

- Functions may no longer have implicit NULL returns. The following demonstrates how implicit NULL returns will be converted by default.

<pre>If a = b Then Return; End If;</pre>	Becomes..	<pre>If a = b Then return NULL; End If;</pre>
--	-----------	---

- The function TO_PLS_INTEGER no longer exists. All references to this function should be replaced with the function TO_NUMBER. The default functionality of the upgrader is to make this change automatically.
- Variables with the datatype of BOOLEAN can no longer have a length, by default the upgrader will remove any length specification from a Boolean datatype.
- In PL/SQL version 1 the LENGTH function would return 0 (zero) if the length of the character string was null. In PL/SQL version 8 the LENGTH function will return a NULL if it is given a character string of NULL length. In order for your existing code to work the same all calls to the LENGTH function will be embedded within the NVL function by default.

LENGTH(NULL);	Becomes..	NVL(LENGTH(NULL),0);
---------------	-----------	----------------------

INITIATING THE UPGRADE

You have two options for upgrading to Oracle Forms Release 6, interactive and batch. You can upgrade your Forms with the batch executable "ifcmp60.exe", or open your old version FMB file in the Oracle Forms Builder Release 6. For Oracle Reports you have the same two options, in batch mode using the "rwcon60.exe" or opening the report in the Oracle Reports Builder Release 6. Either case will create a log file with a PLG extension. This log file will document all the changes that have been made to the Oracle Forms module during the upgrade process. The log file will also list cases where manual intervention is needed in order for the module to be fully upgraded and functional.

INTERACTIVE

In order to convert your Form interactively simply open your old FMB file in the Oracle Forms Builder Release 6. If there is any PL/SQL code in your Form that requires a syntax change in order to PL/SQL 8 compliant the screen in Figure 1 will appear:

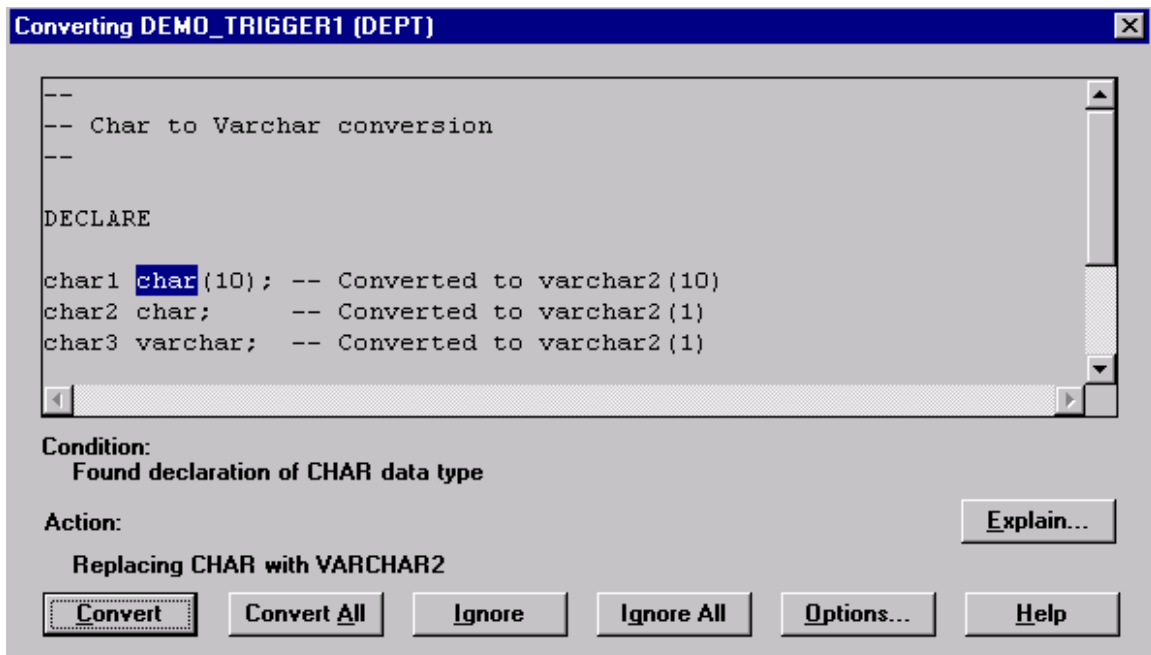


Figure 1

You have the option to convert the code or ignore it. If you ignore and do not convert your PL/SQL code your form will not compile in the Oracle Forms Builder, and you will be unable to generate a FMX file. Pressing the Options button will show you the window in the Figure 2 below. This window will allow you to filter the code that you wish to manually change.

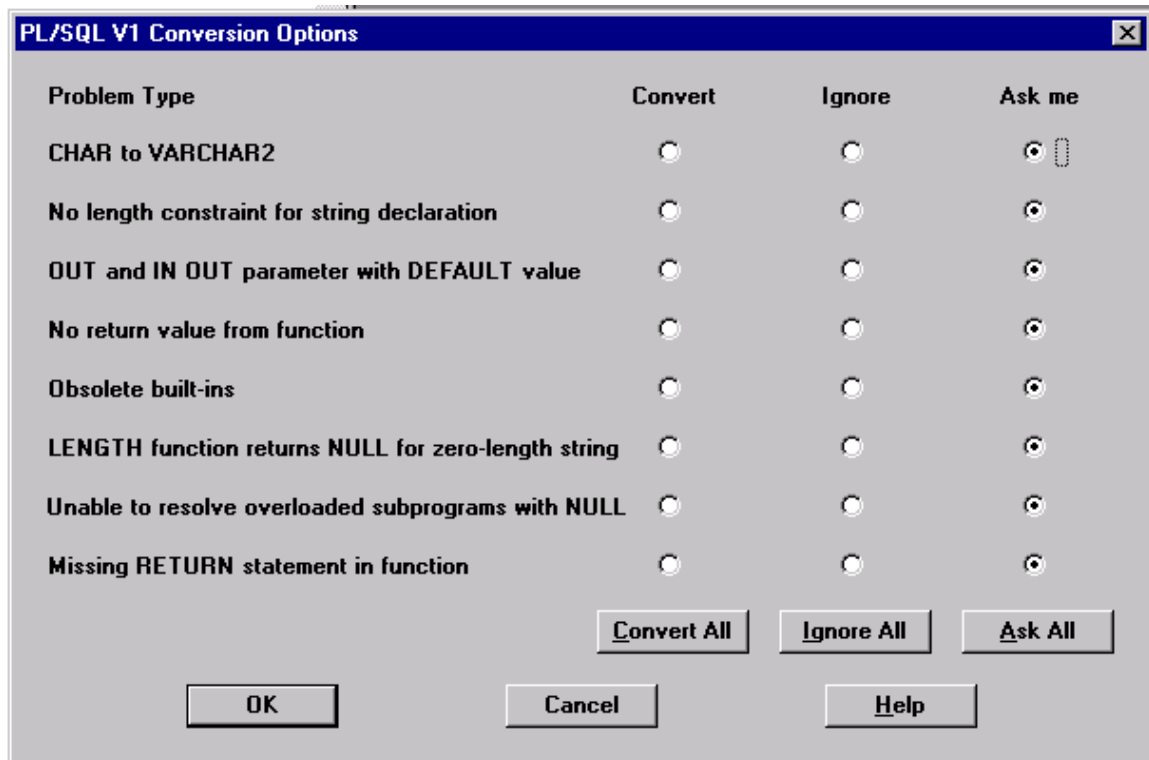


Figure 2

The same PL/SQL conversion will take place for your reports RDF files that you open in the Oracle Reports Builder Release 6.

BATCH:

If you choose to convert your Oracle Forms and Oracle Reports in Batch mode, all of the default actions discussed in the Interactive section will be made automatically. In order to convert a form in batch mode from the command line you need to call:

```
ifcmp60 module=old.fmb userid=scott/tiger@db upgrade_plsql=yes
```

batch=yes

For Oracle Reports use the command line

```
rwcon60 source=myrep upgrade_plsql=yes
```

LOG FILE:

A log file of all PL/SQL conversion changes will be made regardless of whether you convert you Oracle Form or Oracle Report interactively or in batch mode. The log file will have a PLG extension, and will contain the details of what was converted and what was not converted. The things that were not converted may require manual intervention. An example of a change that will require manual intervention is the use PL/SQL Version 8 reserved word VARIANCE demonstrated in figure 3, the sample file below.

```
PL/SQL V1 Conversion Summary
DEMO_TRIGGER1 (DEPT)
  Converted: CHAR to VARCHAR2 at line 7 column 7
  Converted: CHAR to VARCHAR2 at line 8 column 7
  Converted: No length constraint for string declaration at line 8 column 7
  Converted: No length constraint for string declaration at line 9 column 7
  Converted: No length constraint for string declaration at line 10 column 7
DEMO_TRIGGER2 (DEPT)
  ERROR 103 at line 9, column 1
    Encountered the symbol "VARIANCE" when expecting one of the following:

    begin function package pragma procedure subtype type use
    <an identifier> <a double-quoted delimited-identifier> cursor
    form current
    The symbol "begin" was inserted before "VARIANCE" to continue.
  ERROR 103 at line 2, column 4
    Encountered the symbol "end-of-file" when expecting one of the following:

    begin declare end exception exit for goto if loop mod null
    pragma raise return select update while <an identifier>
    <a double-quoted delimited-identifier> <a bind variable> <<
    close current delete fetch lock insert open rollback
    savepoint set sql commit <a single-quoted SQL string>
  Converted: Removing length constraint on type DATE at line 7 column 11
  Converted: Removing length constraint on type BOOLEAN at line 8 column 17
  Not Converted: VARIANCE is a new PL/SQL reserved word at line 9 column 1
  Converted: Replacing LENGTH(...) with NVL(LENGTH(...), 0) at line 10 column 6
```

Figure 3

MANUAL INTERVENTION NEEDED DURING ORACLE MANUFACTURING AND FINANCIAL APPLICATION UPGRADE

This section list all known places where manual intervention may be required in your Form and Library modules.

DUPLICATE PROCEDURES IN THE SAME PACKAGE DEFINITION

It is illegal to have two procedures with exactly the same name and exactly the same parameter definition in a Package.

```
PACKAGE BODY my_example_pack IS
    PROCEDURE foo IS
    BEGIN
        Null;
    END;

    PROCEDURE foo IS
    BEGIN
        Null;
    END;
END my_example_pack;
```

Unfortunately the above syntax would compile under the PL/SQL V1 compiler even though the code itself was illegal. With the PL/SQL V8 compiler the above code will cause ERROR-305. The solution to this problem is to delete the duplicate procedure.

CANNOT READ PARAMETERS OF OUT TYPE

Example:

```
PROCEDURE foo(p_outvar OUT NUMBER) IS
    V_locvar NUMBER;
BEGIN
    V_locvar := p_outvar;
END;
```

Again the PL/SQL V1 compiler did not catch this illegal code. This will not compile with the

PL/SQL V8 compiler. In order to fix the compile error, make the variable that is of type OUT type IN OUT.

VARIABLES AND PROGRAM UNITS WITH SAME NAME IN SAME PACKAGE

It is no longer legal to have program units and variables with the same name in the same package. The compiler cannot resolve which to reference. With PL/SQL V1 you may have gotten unexpected results at runtime, meaning that while coding you may have meant to reference the value of a variable, but instead got the return value of a function with the same name. Oracle recommends prefixing all variable with a 'v_'.

PL/SQL V8 RESERVED WORDS

PL/SQL version 8 has new reserved words. If these new reserved words were used in your PL/SQL version 1 code as variable names, procedure names, or function names then manual intervention is required to change these names.

- VARIANCE
- VALUE
- MAIL

MASTER/DETAIL RELATIONSHIPS

In versions of Forms previous to 6 it was possible to create forms with master/detail relationships where one of the blocks was a control block. This is no longer allowed in Forms 6. When you upgrade your PL/SQL code will be converted successfully, but when you compile your form to FMX you will get the error shown in the diagram below. You must manually change the blocks involved in the relationship to both be data blocks, or remove the relationship.

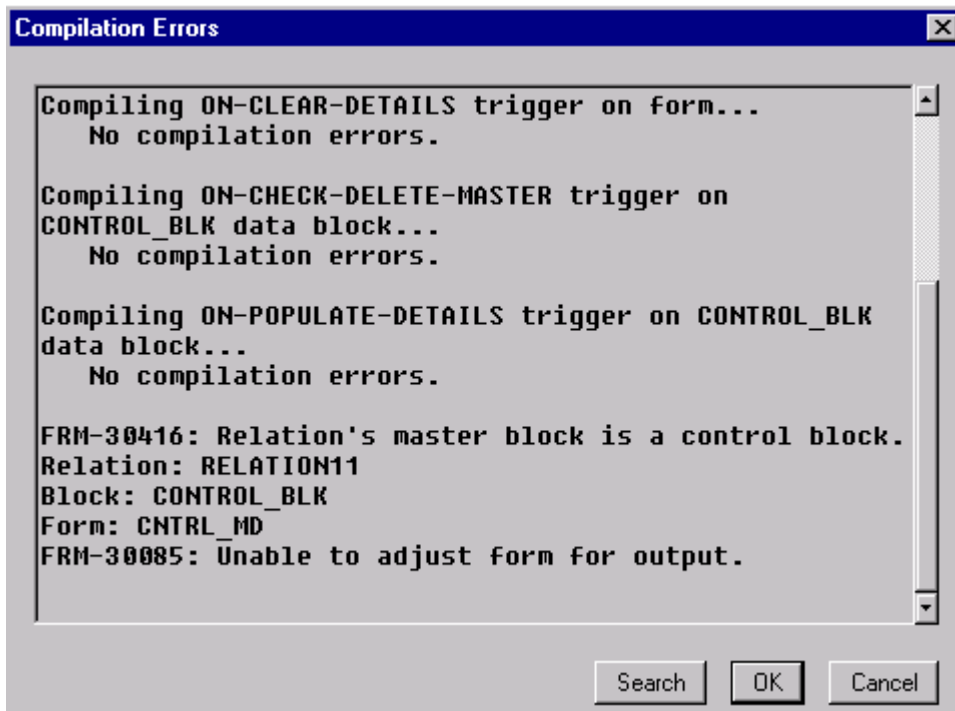


Figure 4

REFERENCING NON-EXISTENT MIRROR ITEMS

In Oracle Developer Forms 4.5 Items had a property called *Mirror Item*. In Oracle Forms Release 6 this property has been renamed to *Synchronize with Item*. Using the example in figure 5, Item5 is specified to *Synchronize with* Item4. Item4 no longer exists. This will not compile in Oracle Forms Release 6. You must remove the reference to Item4 in the Property Palette as this item no longer exists in the form module.

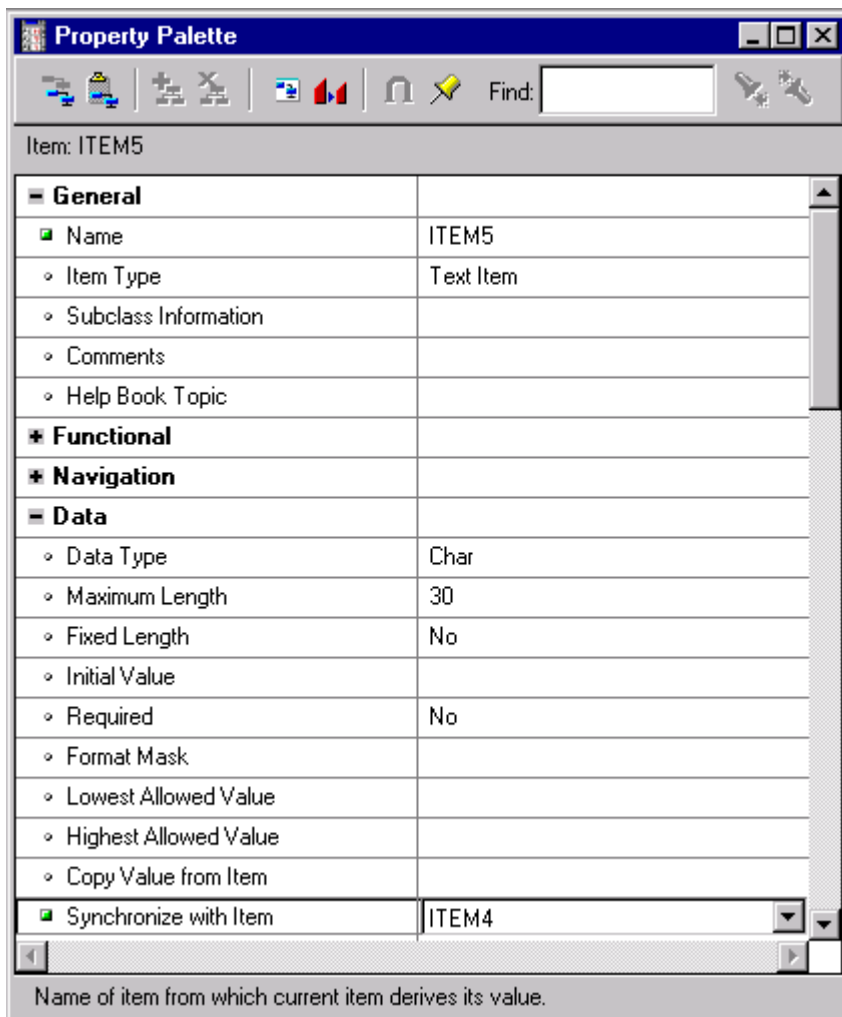


Figure 5

INVALID FORMAT MASKS ON DISPLAY ITEMS

Oracle Developer Forms 4.5 did not verify that the format mask specified in the Property Palette of an item matched the datatype of the item. This check is now done in Oracle Forms Release 6. You must either remove the incorrect format mask, or use a format that matches the datatype of the item.

SCROLLBARS ON NON-EXISTENT CANVASES

If a Data Block's scrollbar is placed on a canvas, and then that canvas is deleted, in Oracle Developer

Forms 4.5 no error was generated. You would not see the scrollbar at runtime. Oracle Forms Release 6 now checks all property values to make sure the reference still exists before it allows the Forms to be generated. If this has happened, you must manually remove the reference to the non-existent canvas in the block properties.

OVERLOADED FUNCTIONS THAT ALLOW NULL ARGUMENTS

It is no longer valid for overloaded functions to allow null arguments. You must modify your code when calling an overloaded function to put a place holder in for each argument. Use the keyword NULL if necessary.

PROGRAM UNITS IN A PACKAGE WITH SAME NAME AS FORMS BUILT-INS

It is legal to name a procedure or function in a package to have the same name as a Forms Built-In. For example:

```
My_package.SET_ITEM_PROPERTY( _ ) ;
```

In Oracle Developer Forms 4.5 if the procedure or function name that matched a Forms Built-In did not exist in the package, it would fall through and call the Forms Built-In instead. This will no longer happen in Oracle Forms Release 6. You will now get an error saying the procedure or function does not exist in the package. You must either code the procedure or function in the package, or remove the package name prefix from the call so that you are calling the Forms Built-In.

FONTS

Be aware that the default value for the environment variable FORMS60_DEFAULTFONT changed. This may cause your layout on your canvases to change slightly if upgrading with Oracle Forms Release 6. In order to alleviate any redraw problems that may occur in Release 6, simply set the environment variable to the same value as your forms were created with in prior versions.

DBMS_* PACKAGES

One major difference that you may encounter in the PL/SQL upgrade is that DBMS_* package calls are not supported on the client. All DBMS_* packages are server-side PL/SQL only. In Oracle Developer Forms 4.5 calls to DBMS_* were compiled and executed successfully. After the upgrade to Oracle Forms Release 6, your Forms may no longer compile, or they may cause a runtime error. This is not a bug as DBMS_* calls are not supported in client-side PL/SQL code. The work around for this problem is to create a server-side procedure that in turn calls the DBMS_* package.

BUGS AND FIXES

Oracle is always striving to improve its products. Based on information learned from the Oracle ERP Applications upgrade as well as feedback from customers a number of migration bugs have been identified and fixed. If you run into any problems during your migration from Oracle Developer Forms 4.5 to Oracle Forms Release 6 please check the following list for your problem. You may just need to ensure the correct Patch has been applied to your Oracle Forms installation

PATCH 1

BUG #	Fixed In	Description
707437	6	APPS992:GEN:ORDERING OF BLOCKS REFERENCED VIA OBJECT GROUPS IS LOST IN 4.5 UPGRADE
689209	6	APPS992:AR:GETTING ERROR FRM-41104: CANNOT FIND RELATION WHILE QUERYING RECORDS
568616	6.0.0.9	VERTICAL BOILERPLATE-LINES GETTING LOST AFTER UPGRADE
705338	6.0.2.0.0	4.5 TO 6.0 UPGRADE : QUERY LENGTH BECOMES 0 IN PROPERTY CLASS
683221 465251	6.0.4	Block property 'base table', which is referenced from a property class,is not transferred correctly when migrating
851153	6.0.4	frm-40209 on execute_query if you enter in a non-base-table-item '#' and has a pre-query-trigger
824443	6.0.4	UPGRADING WITH REFERENCED OBJECTS BRINGS CRASH IN IFBLD32.EXE
798886	6.0.4.0.10	CANNOT UPDATE DETAIL BLOCK IN THE FORM
784695 754308	6.0.4.1.0	GMTICS:UPGRDATION: GETTING FRM-30064:UNABLE TO PARSE STMT RECORD GROUPS
753815	6.0.4.1.0	IF~/.DEFAULT.ORA IS CORRUPT THEN UPGRADE SHOULD CONTINUE

745489 734737	6.0.4.12	V1/V8 CONVERSION DOES NOT HAPPEN WHILE UPGRADING IN BATCH MODE
540833	6.0.4.14.0	ONLINE HELP IS NOT AVAILABLE FOR PICK LANGUAGE DIALOGS.
710424	6.0.4.4.1	NLS: CANNOT CHANGE FONT SIZE FROM SET_ITEM_PROPERTY BUILT-IN
159851	6.0.4.3.0	INCONSISTENCY IN DESC. BETWEEN THE GENERATOR OPTION WINDOW AND HELP=YES OPTION
1001562 962726	6.0.4.5.0	BUILDER:4.5 - 6.0 UPGRADE:TRYING TO OPEN ANY 4.5 FORM DUMPS CORE.
908625	6.0.4.7.1	MENU SUBCLASSING IS LOST WHILE UPGRADING 4.5 MENUS TO 6.0
899724	6.0.4.8.0	FIELDS AND PROMPTS ARE NOT DISPLAYED IN WEB-FORMS
882532	6.0.5	FORMS COMPILER STILL HAS OPTION TO UPGRADE 3.0 FORM OR 5.0 MENU
877905 701248	6.0.5	BUGOUT60 CANNOT HAVE A PROGRAM UNIT NAMED FILE
874116	6.0.5	DBMS_ERROR_CODE GIVES WRONG VALUES IN THE ON-ERROR TRIGGER
858863	6.0.5	PROBLEM WITH TELEPHONE NUMBERS ITEM WITH FORMAT MASKS "("999") "999"- "9999
857219	6.0.5	SOME 4.5 FORMS ARE CAUSING A CORE DUMP WHEN OPENED WITH F60DESM
813035	6.0.5	GFP ON USING FORMS WITH FULL-SCREEN MENUS
809894	6.0.5	PROBLEM WITH MENU OPTIONS : 4.5 TO 6.0 BEHAVIOUR DIFFERENCE.
809614	6.0.5.0.0	BACKGROUND COLOR FOR TEXT ITEMS/DISPLAY ITEMSCHANGES FROM 4.5 TO 6.0
791543	6.0.5.0.0	FORMS BUILDER: DATE VALUE CONVERSION PROBLEM IF OPEN FORM CREATED WITH FORMS 4.5
751478	6.0.5.29	PLS-103 IF COMPILE MIGRATED MENU CONTAIN. REFERENCED ITEM OF COMMAND TYPE MENU
749456	6.0.5.29.4	ORDERING OFSUBCLASSED BLOCKS REFERENCED VIA OBJECT GROUPS IS NOT PROPER

PATCH 3

BUG #	Fixed In	Description
790617	6.0.5.30.2	BUTTONS APPEAR TO HAVE A BORDER, MAKING IT NECESSARY TO INCREASE WIDTH AND HEIGHT FOR ALL BUTTONS
747145	6.0.5.31.0	AUTOSKIP DOES NOT WORK CORRECTLY IN CHARACTER MODE

PATCH 4A

BUG #	Fixed In	Description
746167 754395	6.0.5.32.0	BUILDER:UPGRADE:UPGRADE SUMMARY IS NOT GETTING DISPLAYED

PATCH 5A

BUG #	Fixed In	Description
746038	6.0.5.34.0	UPGRADE:NOT ABLE TO UPGRADE LIBRARIES IF ATTACHED LIBRARIES ARE MISSING

RELEASE 6i

BUG #	Fixed In	Description
718392 696935	6.0.6	V1V8:CONVERTER FAILS TO CONVERT CODE AFTER FIRST ERROR IN A PROCEDURE
716838	6.0.6	TEXT ITEM FIELD IS NOT SELECTED/HIGHLIGHTED WHEN NAVIGATING FIELD IN WEBFORMS
715409 689209	6.0.6.3.0	RECORDS NOT QUERIED AGAIN WHILE OPENING THE SAME FORM WITH DIFFERENT CRITERIA
700959	6.0.6.4	FORMS UPGRADED FROM 4.5 TO 6.0 WITH NON-RENDERED FIELDS CANNOT DISPLAY TEXT
694465 557637	6.0.6.4.0	LOV DISPLAYING ONLY WINDOW TITLE WHEN HEIGHT AND WIDTH SET TO ZERO
690632 683423	6.0.6.4.0	UPGRADE:DUMPS CORE WHILE UPGRADING
650728	6.0.6.5.0	WEBFORMS: HORIZONTAL TOOLBAR SIZE ADDED TO CONTENT CANVAS SIZE
628710	6.0.6.5.0	LOV SELECTION CRITERIA BEHAVING DIFFERENTLY
612537 619876	6.0.8.0.0	ACCESS VIOLATION USING DBMS_TRANSACTION.COMMIT IN FORMS
608768 251381	6.0.8.1	VISUAL ATTRIBUTES CAN INHERIT PROPERTY CLASSES AFTER FORM UPGRADE
605998	6.0.8.2.0	SYSDATE RETURNS THE CLIENT DATE AND TIME RATHER THAN SERVER

ADDITIONAL INFORMATION

Additional information may be obtained from the Oracle Technology Network (OTN). OTN is a free technical resource site for all Oracle products.

<http://technet.oracle.com/>

In the Developer/Developer Server section in OTN you will find the following papers of interest:

- Upgrade Existing Applications to Developer – This paper is designed to help with the conversion of character based Forms to a graphical interface.
- Deploying Applications on the Web with Oracle Developer Server: Intranet, Extranet, Internet



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

Worldwide Inquiries:
+1.650.506.7000
Fax +1.650.506.7200
<http://www.oracle.com/>

Copyright © Oracle Corporation 1999
All Rights Reserved

This document is provided for informational purposes only, and the information herein is subject to change without notice. Please report any errors herein to Oracle Corporation. Oracle Corporation does not provide any warranties covering and specifically disclaims any liability in connection with this document.

Oracle is a registered trademark, and Oracle8i, Oracle8, PL/SQL, and Oracle Expert are trademarks of Oracle Corporation. All other company and product names mentioned are used for identification purposes only and may be trademarks of their respective owners.