Oracle® Forms

Migrating Forms Applications from Forms 6*i*

10*g* (9.0.4)

For Windows and UNIX

Part No. B10469-01

September 2003

ORACLE®

Oracle Forms Migrating Forms Applications from Forms 6*i*, 10*g* (9.0.4)

Part No. B10469-01

Copyright © September 2003 Oracle Corporation. All rights reserved.

Primary Author: Orlando Cordero

Contributors: Ashwin Baliga, Emerson deLaubenfels, Arthur Housinger, David Klein, Duncan Mills, Girish Nagaraj, Frank Nimphius

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and JInitiator, Oracle SQL/Services, Oracle8, Oracle8i, Oracle9i, and PL/SQL are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Intended Audience xiii
Structure xiii
Related Documents xv

1 Why Migrate?

Forms 6i Features Removed from Oracle Forms 1-1
Components Removed from the Developer Product Suite 1-2
How are Obsolete Item Types Handled when Migrating Forms 6i Applications? 1-3
Are Tools Available to Assist with the Migration of Obsolete Features? 1-4

2 Using the Oracle Forms Migration Assistant

What Does the Oracle Forms Migration Assistant Do? 2-1
Mutiple Log Support 2-2
Editing the converter.properties File 2-3
Editing the search_replace.properties File 2-4
Adding Search-and-Replace Strings 2-4
Modifying Warnings for Obsolete Built-Ins 2-5
Starting the Oracle Forms Migration Assistant 2-6
About the Migration Assistant in Batch Mode 2-7
Starting the Migration Assistant in Batch Mode 2-8
Running the Wizard Version of the Forms Migration Assistant 2-9
Starting the Wizard Version of the Forms Migration Assistant 2-9
Setting Advanced Converter Options 2-10

3 Converting Forms 6i FMTs to Oracle Forms FMBs

Converting a Forms 6i FMT to an Oracle Forms FMB 3-1

4 Built-ins, Packages, Constants, and Syntax

Obsolete Menu Built-ins 4-1
Other Obsolete Built-ins 4-3
Obsolete Built-in Packages 4-4
Obsolete Constants 4-5
Obsolete Syntax 4-6

5 Triggers

Obsolete Triggers 5-1
Stricter Enforcement of Triggers 5-1

6 Properties

Obsolete Properties 6-1

7 Changes to Client-Server Deployment and Forms Runtime

Effect on Forms Development 7-1
Obsolete Forms Runtime Command Line Options 7-2
Obsolete Character Mode Runtime 7-2

8 Logical and GUI Attributes

Use Visual Attributes to Replace Logical and GUI Attributes 8-1
Obsolete Logical and GUI Attributes 8-1

9 Item Types

Item Types Specific to Operating Systems 9-1

10 List of Values (LOVs)

Obsolete List of Values (LOVs) 10-1

11 User Exits

Obsolete V2 User Exits 11-1

12 Menu Parameters

Predefined Menu Parameters 12-1
User-Defined Menu Parameters 12-2

13 Java-Related Issues

Using oracle.ewt Classes in Pluggable Java Components 13-1
JDK Versions and Font-Rendering Issues 13-2

14 Integrating Reports and Graphics

Oracle Graphics6i 14-1
Displaying Existing Charts in Oracle Forms 14-1
Editing Existing Chart Items 14-2
About Integration with Oracle Reports 14-3
Displaying Reports in Oracle Forms 14-4
Example 14-4
Notes About the Example 14-5
Using Parameter Lists in RUN_REPORT_OBJECT 14-6
Migration Steps 14-7

15 Migrating Client-Server Applications to the Web

Client-Server-Based Architecture 15-1
Web-Based Architecture 15-2
Who Should Read this Chapter? 15-3
Guidelines for Migration 15-4

16 Upgrading from Pre-Forms 6i Applications to Oracle Forms

About Upgrading Forms 16-1
Upgrading a Form 16-1
PL/SQL 9 Support 16-2
Compatibility with Earlier Versions of PL/SQL 16-2

Forms Developer Runtime Behavior..... 16-2

Index

List of Figures

15–1 Legacy Forms Server client-server-based architecture 15-2
15–2 Forms Services Web-based architecture 15-3

Send Us Your Comments

Oracle Forms

Migrating Forms Applications from Forms 6*i*

10*g* (9.0.4)

For Windows and UNIX

Part No. B10469-01

September 2003

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: formsdoc_us@oracle.com
- Postal service:
  Oracle Corporation
  Forms Developer and Forms Services Documentation
  200 Oracle Parkway, 2op1224
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

List of Tables

1–1	Components Removed from the Developer Suite	1-2
2–1	Oracle Forms Migration Assistant converter.properties File Options	2-3
2–2	Oracle Forms Migration Assistant Command Line Parameters	2-7
2–3	Converter Properties	2-11
4–1	Obsolete Menu Built-ins	4-1
4–2	Other Obsolete Built-ins	4-3
4–3	Obsolete Built-in Packages	4-5
4–4	Obsolete Constants	4-5
5–1	Obsolete Triggers	5-1
5–2	Triggers with Restricted Usage	5-2
6–1	Obsolete Properties	6-2
8–1	Obsolete Logical and GUI Attributes	8-1
9–1	Obsolete Item Types	9-1
12–1	Obsolete Predefined Menu Parameters	12-1
15–1	Font Support for Web-based Applications	15-4
16–1	Version Number for Forms 4.5 Generate Command	16-2

Preface

Welcome! This manual describes:

- Features and functionality that have been removed from Forms Developer and Forms Services
- Information about migration events that automatically occur when you open or deploy a Forms 6*i* application in Oracle Forms.
- Information about the Oracle Forms Migration Assistant, a tool to help you convert your applications.
- Information about steps that developers, system administrators, and DBAs need to take to migrate Forms applications from Forms 6*i* to Oracle Forms Forms.

Intended Audience

This manual is intended for developers, system administrators, and DBAs who develop and deploy Oracle Forms applications.

Structure

This manual contains the following chapters:

Chapter 1, "Why Migrate?"

This chapter explains why Oracle Forms is desupporting a number of features and replacing them with new functionality. It also lists the Forms 6*i* features and the Developer suite components that have been removed for the 9*i* release.

Chapter 2, "Using the Oracle Forms Migration Assistant"

This chapter describes the Oracle Forms Migration Assistant, a command line tool or wizard that will help you migrate Forms applications from Forms 6*i* to Oracle Forms. This tool will help you detect and migrate obsolete features.

Chapter 3, "Converting Forms 6i FMTs to Oracle Forms FMBs"

This chapter describes how to convert Forms 6*i* FMTs and MMTs to Oracle Forms FMBs and MMBs.

Chapter 4, "Built-ins, Packages, Constants, and Syntax"

This chapter describes the PL/SQL Built-ins, packages, constants, and syntax that have been removed for the 9*i* release and information that will help you replace the obsolete functionality.

Chapter 5, "Triggers"

This chapter describes the triggers that have been removed for the 9*i* release and information that will help you replace the obsolete functionality.

Chapter 6, "Properties"

This chapter describes the properties that have been removed for the 9*i* release and information that will help you replace the obsolete functionality.

Chapter 7, "Changes to Client-Server Deployment and Forms Runtime"

This chapter describes the runtime changes for the 9*i* release and information that will help you replace the obsolete functionality.

Chapter 8, "Logical and GUI Attributes"

This chapter describes the logical and GUI attributes that have been removed for the 9*i* release and information that will help you replace the obsolete functionality.

Chapter 9, "Item Types"

This chapter describes the items that have been removed for the 9*i* release and information that will help you replace the obsolete functionality.

Chapter 10, "List of Values (LOVs)"

This chapter describes the LOVs that have been removed for the 9*i* release and information that will help you replace the obsolete functionality.

Chapter 11, "User Exits"

This chapter describes the user exits that have been removed for the 9*i* release and information that will help you replace the obsolete functionality.

Chapter 12, "Menu Parameters"

This chapter describes the menu parameters that have been removed for the 9*i* release and information that will help you replace the obsolete functionality.

Chapter 13, "Java-Related Issues"

This chapter describes migration steps to take if your Forms applications use Java-related components.

Chapter 14, "Integrating Reports and Graphics"

This chapter describes changes that you need to make to Forms applications in order to integrate existing Graphics6*i* and Reports applications.

Chapter 15, "Migrating Client-Server Applications to the Web"

If you deployed Forms applications in a client-server environment in the past, use the information in this chapter to understand the changes you need to make in order to deploy your Forms applications on the Web. Additional information is available in the manual *Oracle Application Server Forms Services Deployment Guide (Part No. A92175-01)*.

Chapter 16, "Upgrading from Pre-Forms 6i Applications to Oracle Forms"

This chapter contains hints to help you convert pre-Forms 6*i* applications to Forms 6*i*. After converting applications to Forms 6*i*, you can migrate to Oracle Forms.

Related Documents

For more information, see the following manuals and online help:

- *Oracle Forms Developer and Forms Services: Release Notes for Windows (Part No. B10472-01)*
- *Oracle Forms Developer and Forms Services: Release Notes for UNIX (Part No. B10471-01)*
- *Oracle Application Server Forms Services Deployment Guide (Part No. B10470-01)*
- Oracle Forms Developer Online Help, available from the Help menu in Oracle Forms Developer.

In addition, you will find white papers and other resources at http://otn.oracle.com.

1

Why Migrate?

Oracle Forms Developer and Forms Services have been upgraded to simplify the development and deployment of Forms applications on the Web. A number of new features have been added. In restructuring the Oracle Forms product, some legacy features have been dropped or scaled back. The future of Forms includes improving the Java-based Web user interface and extending product "openness" by allowing Java integration on all three tiers.

Forms 6*i* Features Removed from Oracle Forms

The following features have been dropped from Oracle Forms:

- Client-server runtime
- Character mode runtime
- Various runform command line options
- Character mode properties and logical attributes
- Item types that are operating-system specific
- Various Built-ins
- Various properties
- Various menu features including:
  - Character mode menu properties
  - Obsolete types from the Menu-Items command type property
  - Menu parameters
  - Menu Built-ins

  - Full screen menu style
  - Bar (Lotus) menu style
- Forms version 2 style triggers and list of values (LOVs)
- Graphics Chart Wizard

In addition, rules enforcing trigger usage have become more strict.

Components Removed from the Developer Product Suite

The following components have been removed.

Table 1–1 Components Removed from the Developer Suite

Obsolete Component	Migration Notes
Oracle Graphics	If your applications use the Graphics Web Cartridge or Oracle Graphics Runtime, you should remain on Forms 6*i*. You can upgrade and deploy existing Forms applications that contain embedded Oracle Graphics using the RUN_PRODUCT call, or by displaying them as chart objects. To do this, you must install Oracle Graphics6*i* on the same machine as Forms Services, but in a separate ORACLE_HOME.
Oracle Forms Listener and Load Balancing Components	Use the Forms Listener Servlet to manage Forms sessions on the Web. The Forms Listener Servlet provides: ■ Improved security because all traffic is directed through standard Web server HTTP or HTTPS ports, with no extra ports open through the firewall. ■ Use of standard load balancing techniques. ■ Broader firewall and proxy support. ■ Less administration because the listener and load balancing processes do not need to be managed. ■ Simplified HTTPS support because a separate Web server SSL certificate for the Forms listener is not required. ■ HTTPS support for Internet Explorer 5.x with native JVM.
Oracle Forms Server Cartridge and CGI	Use the Forms Servlet. The functionality available with the Oracle Forms Server cartridge and CGI was incorporated into the Forms Servlet, which was first available in Oracle Forms Release 6*i* patchset 2.

Table 1–1 Components Removed from the Developer Suite

Obsolete Component	Migration Notes
Oracle Procedure Builder	Use the facilities for editing and debugging local and server-side PL/SQL code in Forms Developer, which has been considerably improved for this release.
Oracle Project Builder	No migration path or replacement functionality at this time.
Oracle Translation Builder	Use TranslationHub to translate resource strings in Forms modules in order to deploy the modules in multiple languages.
Oracle Query Builder/Schema Builder	No migration path or replacement functionality.
Oracle Terminal	The resource files used by Web-deployed forms are text based and can be edited using a conventional text editor. As a result, Oracle Terminal is no longer required for the product.
Open Client Adapters (OCA)	In order to provide platform-independent access to a wider range of non-Oracle data sources, use the Oracle Transparent Gateway and Generic Connectivity solutions in place of OCA.
Tuxedo Integration	No migration path or replacement functionality.
Performance Event Collection Services (PECS)	No migration path. Use Forms Trace and Oracle Trace, which are documented in the book *Oracle Application Server Forms Services Deployment Guide (Part No. A92175-01)*.

How are Obsolete Item Types Handled when Migrating Forms 6*i* Applications?

When you open a Forms application, obsolete item types are listed at the end of the item-type poplist in the Property Palette. The property values for obsolete items are indicated as obsolete. For example, the property value for VBX would be "VBX Control (Obsolete)".

Note: If you are migrating from older releases of Forms to Oracle Forms, you must first migrate your applications to Forms 6*i*, and then migrate them to Oracle Forms. See Chapter 16, "Upgrading from Pre-Forms 6i Applications to Oracle Forms" for more information.

You can use the Oracle Forms Migration Assistant, described in Chapter 2, "Using the Oracle Forms Migration Assistant", to resolve many migration issues.

Are Tools Available to Assist with the Migration of Obsolete Features?

The Oracle Forms Migration Assistant is provided with Oracle Forms to help you migrate your Forms 6*i* applications. See Chapter 2, "Using the Oracle Forms Migration Assistant" for details.

2

Using the Oracle Forms Migration Assistant

Oracle Forms provides a tool that updates obsolete usage in your PL/SQL code in order to migrate your Forms *6i* applications to Oracle Forms. The tool issues warnings when it cannot make the required changes automatically. This tool has a command line and a wizard version.

This chapter contains the following sections:

- "What Does the Oracle Forms Migration Assistant Do?"
- "Editing the converter.properties File"
- "Editing the search_replace.properties File"
- "Starting the Oracle Forms Migration Assistant"

Go to http://otn.oracle.com for the latest version of the tool.

What Does the Oracle Forms Migration Assistant Do?

The Oracle Forms Migration Assistant does the following for all Forms module types (including object libraries and PL/SQL libraries):

- Updates PL/SQL code where possible, for example:
  - Updates RUN_PRODUCT to the RUN_REPORT_OBJECT Built-in when used to call Reports.
  - Updates CHANGE_ALERT_MESSAGE to the SET_ALERT_PROPERTY Built-in.
- Provides a list of obsolete code usage, including code that the tool cannot change when there is not a straight-forward equivalent for migration, for example:

  - Provides warnings when specific obsolete Built-ins are used at runtime, such as ITEM_ENABLED.

    Note: The Oracle Forms Migration Assistant replaces Built-ins and issues warnings about Built-ins that exist within code comments.

- Provides warnings when obsolete functionality is encountered, such as when obsolete item types are included in the code.
- Provides warnings about triggers defined at incorrect levels.
- Replace simple one-for-one code strings such as OHOST to HOST, MENU_CLEAR_FIELD to CLEAR_ITEM, and MENU_FAILURE to FORM_FAILURE.
- Performs more complex substitutions such as CHANGE_ALERT_MESSAGE to SET_ALERT_PROPERTY Built-in, DISABLE_ITEM to SET_MENU_ITEM_PROPERTY Built-in, ITEM_ENABLED to GET_ITEM_PROPERTY Built-in, and ENABLE_ITEM to SET_MENU_ITEM_PROPERTY Built-in.
- Raises a warning in the Form module's log if V2-style triggers are found.
- Raises a warning in the Form module's log if Built-ins related to obsolete object types are found such as VBX.FIRE_EVENT, VBX.GET_PROPERTY, and VBX.GET_VALUE_PROPERTY.

You run the Oracle Forms Migration Assistant in batch mode. You can re-enter the utility as needed to run the migration process on a Forms application more than once.

By editing the converter.properties file, you can set options before you start the batch migration. By editing the search_replace.properties file you can specify the strings that the Oracle Forms Migration Assistant will search for and replace, and edit the warnings that are issued when an obsolete Built-ins is encountered.

The tool creates a log file so that you can navigate to problem areas in the application and make modifications manually.

Mutiple Log Support

The Forms Migration Assistant allows you either to write all log information into a single log file or span mutiple log files. If the Forms Migration Assistant spans multiple log files, the Forms Migration Assistant generates individual log files for each module that is processed.

You'll need to specify the directory in which the Forms Migration Assistant writes the log files. The name of the log file which is generated is `modulename_moduletype.log`.

For example, if you process a module by name, such as test.fmb, the name of the log file is test_fmb.log. If you select test.fmb *and* test.mmb for migration, the Forms Migration Wizard does not overwrite the log files since the type of log file was specified in the converter.properties. The Forms Migration Assistant generates two log files: test_fmb.log and test_mmb.log.

Editing the converter.properties File

To change migration options, edit the converter.properties file in a text editor. You can set the following migration options:

Table 2–1 Oracle Forms Migration Assistant converter.properties File Options

Option	Description
Log File Name (default.logfilename)	Specifies the file name and location for log information.
Reports Queue Table Installed (default.usequeuetables)	When using web-based reports, these queue tables can be used to monitor queued and processed reports. When used with the Oracle Forms Migration Assistant, the queue table provides detailed error messages when installed in the application schema. (For example, if a report cannot run due to uncompiled PL/SQL, you can use the queue table to query for full error messages.) The resulting report is automatically printed. See the Reports Services documentation for more information about the Reports Queue Table.
Reports Servlet Directory (default.servletdir)	Specifies the name that has been defined for the virtual path used for the Reports Servlet, which is used for running reports on the web. This setting is required when converting Run_Product calls to Run_Report_Object.
Reports Servlet Name (default.servletname)	Specifies the name for the Reports Servlet used for running reports on the web. This setting is required when converting Run_Product calls to Run_Report_Object.
Reports Server Host (default.reports_servername)	The name or IP address of the machine running the Reports Server. This setting is required when converting Run_Product calls to Run_Report_Object.
DESTYPE (default.destype)	The type of destination device that will receive the report output. See the *Reports Developer Reference* for details.

Table 2–1 Oracle Forms Migration Assistant converter.properties File Options

Option	Description
DESFORMAT (default.desformat)	The printer driver to be used when DESTYPE is FILE. See the *Reports Developer Reference* for details.
DESNAME (default.desname)	The name of the file, printer, e-mail ID, or distribution list to which the report output will be sent. See the *Reports Developer Reference* for details.
Reports Server Host (default.reportshost)	The name or IP address of the machine running the Reports Server. This setting is required when converting Run_Product calls to Run_Report_Object.

Editing the search_replace.properties File

The search_replace.properties file contains the strings that the Oracle Forms Migration Assistant will search for and replace. It also contains a list of obsolete Built-ins that will generate warnings.

Adding Search-and-Replace Strings

You can edit this file to add your own search-and-replace strings, as follows:

1. Open the search_replace.properties file in a text editor.
2. Go to the end of the list of search-and replace strings.
3. Add a string to search for and replace using the following syntax:

   ```
   SearchString|ReplaceString
   ```

4. Save the search_replace.properties file.

   Note: Do not delete the last two commands in the search_replace.properties file. See the release notes for more information.

Modifying Warnings for Obsolete Built-Ins

The warnings for Built-ins have the following syntax:

```
<class>.Message=<WarningMessage>
<class>.Warning1=<BuiltIn1>
<class>.Warning2=<BuiltIn2>
<class>.Warning3=<BuiltIn3>
etc.
```

For example, for the class `obsoleteMenuParam`, the warnings are coded as follows:

```
obsoleteMenuParam.Message=Menu Parameters are no longer supported, the parameter
and usage of %s should be replaced using a Forms parameter or global variable.
obsoleteMenuParam.Warning1=MENU_PARAMETER
obsoleteMenuParam.Warning2=QUERY_PARAMETER
obsoleteMenuParam.Warning3=TERMINATE
```

`<class>` is a group of Built-ins that have a common warning. `<WarningMessage>` can contain one variable string (`%s`).

When the Oracle Forms Migration Assistant finds a Built-in for which a warning should be issued, it logs the warning and replaces the variable string (`%s`) with the Built-in name.

Classes that have already been created in the search_replace.properties file are:

- `obsoleteItemTypeBuiltin`
- `obsoleteBuiltin`
- `obsoleteMenuParam`
- `obsoleteItemTypeConstantProp`
- `obsoleteConstantProp`
- `obsoleteConstant`
- `obsoleteHardCodedUserExit`
- `obsoleteComplexBuiltin`
- `obsoleteCallBuiltin`
- `DataParameterWithReports`

You can add more Built-in warnings to an existing class, or create new classes and warnings:

1. Open the search_replace.properties file in a text editor.
2. Go to the end of the list of warning messages.
3. Add a warning to an existing class or create new classes and warnings using the syntax described above.
4. Save the search_replace.properties file.

   Note: Do not delete the last two commands in the search_replace.properties file. See the release notes for more information.

Starting the Oracle Forms Migration Assistant

You should upgrade the common modules (such as OLB files) and Libraries (PLL files) that a Form relies upon and ensure that these are available in the FORMS90_PATH, before running the migration assistant.

Note: For UNIX, an xterm display is required to run the Oracle Forms Migration Assistant.

Note: The forms90\rp2rro.pll and forms90\EnableDisableItem.pll files should be in the FORMS90_PATH in order to convert RUN_PRODUCT to RUN_REPORT_OBJECT.

Note: In UNIX, set the variable FORMS90_PATH, such as

```
setenv FORMS90_PATH $Oracle_Home/forms90
```

To start the Oracle Forms Migration Assistant on Windows, enter:

```
ifplsqlconv90 mode=batch module=<modulename> log=<log>
```

To start the Oracle Forms Migration Assistant on UNIX, enter:

```
f90plsqlconv mode=batch module=<modulename> log=<log>
```

You can also specify additional command line parameters, as described in Table 2–2, "Oracle Forms Migration Assistant Command Line Parameters":

Table 2–2 Oracle Forms Migration Assistant Command Line Parameters

Parameter	Description
modulename (required)	Specifies the module to migrate. The modulename parameter can take only one value. See "About the Migration Assistant in Batch Mode" if you want to migrate more than one file at a time.
log (optional)	Specifies the log file to which the migration results are written. If not specified, the default values from the converter.properties file are used.
Mode (optional)	This option has two values (batch and wizard). Use mode=batch to run Converter in Batch mode.

Information about the progress of the migration process is displayed on the screen. It is also saved to the log file that you specify in the converter.properties file. (To change migration options, see Editing the converter.properties File.)

Check the log file for information about required migration steps that the tool did not modify. Manually make these changes to your application.

About the Migration Assistant in Batch Mode

You can convert multiple applications by running the Forms Migration Assistant in batch mode from the command line. Batch mode is useful for converting multiple Forms applications. For example, on Windows NT, create a batch file (for example upgrade.bat) that contains the following:

```
for %%ff in (%1) do call ifplsqlconv90 module=%%ff
```

Then run the batch file as follows:

```
upgrade *.fmb
```

or

```
upgrade foo*.mmb
```

On UNIX, create a shell script (for example upgrade.sh) that contains the following:

```
for file in $*
```

```
do
   f90plsqlconv.sh module=$ff
done
```

Then run the shell script as follows:

```
upgrade.sh *.fmb
```

or

```
upgrade.sh foo*.mmb
```

Starting the Migration Assistant in Batch Mode

To start the Migration Assistant in batch mode in Windows:

```
ifplsqlconv90 mode=batch module=<filename> log=<logname>
```

To start the Migration Assistant in batch mode in UNIX:

```
f90plsqlconv  mode=batch module=<filename> log=<logname>
```

where <filename> is the name of the file you want to convert, and <logname> is the name of the log file that will get generated.

In the following example:

```
ifplsqlconv90 mode=batch module=d:\temp\test.fmb log=d:\temp\test.log
```

Forms Migration Assistant will look for the file “test.fmb” in the d:\temp directory and will name the generated log file “test.log” and place it in the d:\temp directory. You can give the log file any name you want and specify any location you want. By default the log file is written in $ORACLE_HOME\bin. The module parameter can take only one value.

The output of the conversion is displayed on the screen. It is also saved to the default log file. Note that in batch mode, all converter options are taken from the converter.properties file. To change the converter options, see "Editing the converter.properties File".

Running the Wizard Version of the Forms Migration Assistant

You can run the wizard version of the Forms Migration Assitant and edit converter options.

Starting the Wizard Version of the Forms Migration Assistant

To start the wizard version of the Forms Migration Assistant:

1. In UNIX systems, set the variable FORMS90_PATH as in the example below:

   ```
   setenv FORMS90_PATH $Oracle_Home/forms90
   ```

2. From the command line, start the conversion utility by typing `ifplsqlconv90 mode=wizard` on Windows

   or

   `f90plsqlconv mode=wizard` in UNIX.

   The Conversion Wizard **Welcome** dialog displays.

   > **Note:** You can get help by typing -h after the command, for example, ifplsqlconv90 -h.

3. Click **Next**.
4. In the **Modules** dialog, click the **Add Module(s)** button.
5. Select the modules that you want to convert.
6. Click **Next**.
7. In the **Converter options** dialog, enter the location and name of the log file that will be generated. A browse button is provided to assist in choosing a location.
8. If you have Forms that include embedded reports, fill in the remaining fields. Otherwise, you may ignore these fields.

9. For more information about the Reports fields, see *Reports Developer Reference.* For more information about advanced converter options, see the following section, "Setting Advanced Converter Options".
10. Click **Next**.
11. In the **Finish** dialog, you will see the modules you selected. Click **Finish** to start the conversion.

> **Note:** A dialog window will appear recommending that you backup your files before converting. Select the **Show me this again** check box to enable this warning in the future.

12. Progress displays in the Log window. (Log output is written to the log file name(s) that you specify in the Options dialog. See the following section for information about setting options.)
13. Check the log file(s) for information about required conversion steps that the tool did not modify. Manually make these changes to your application.

Setting Advanced Converter Options

> **Note:** For information about setting the search_replace.properties file, see "Editing the converter.properties File".

You can set conversion options before running the wizard:

1. In the **Converter options** dialog (step 2 of the wizard), click **Advanced Options**.
2. Converter properties display in the left column of the dialog. Edit the property values as needed, as described in Table 2–3, "Converter Properties":

Table 2–3 Converter Properties

Display Backup Warning	Determines whether the warning dialog displays to back up files when the application starts.
Log File Name	Specifies the log filename in Single Log mode.
Log Dir	The destination directory to write the log files in multilog mode
Reports Servlet Virtual Directory	Specifies the name that has been defined for the virtual path used to define the Reports Servlet, which is used for running reports on the web. This setting is required when converting Run_Product calls to Run_Report_Object.
Reports Servlet	Specifies the name for the Reports Servlet used for running reports on the web. This setting is required when converting Run_Product calls to Run_Report_Object.
Reports Server	The name or IP address of the machine running the Reports Server. This setting is required when converting Run_Product calls to Run_Report_Object.
Default DESFORMAT	Printer driver to be used when DESTYPE is PRINTER (XML, HTML, HTMLCSS, PDF, RTF, delimited) .
Default DESTYPE	Type of destination device that will receive the report output (cache, printer, file).
Default DESNAME	Name of the file, printer, e-mail ID, or distribution list to which the report output will be sent
Default Browser	For UNIX, the browser used to display help for the Migration Assistant. This setting is either *netscape* or *iexplore*. In Windows, the system default browser is used.

3. Click **OK** to save the configuration. The data is saved to the converter.properties file, and the settings you specify will be used in the current *and* future sessions of the utility.

3

Converting Forms 6*i* FMTs to Oracle Forms FMBs

Because some properties are obsolete in Oracle Forms, you cannot directly convert Forms 6*i* FMTs and MMTs to Oracle Forms FMBs and MMBs using Forms Developer.

Converting a Forms 6*i* FMT to an Oracle Forms FMB

To convert a Forms 6*i* FMT or MMT to an Oracle Forms FMB or MMB, do the following:

1. Use the Forms 6*i* Builder or Compiler to convert the Forms 6*i* FMT or MMT to a Forms 6*i* FMB or MMB.
2. Then, use Forms Developer to convert the Forms 6*i* FMB or MMB to an Oracle Forms FMB or MMB.

4

Built-ins, Packages, Constants, and Syntax

In order to streamline the tools available and simplify the development process for building Forms applications for the Web, Built-ins, constants, packages, and some syntax that are not applicable to Web deployment have been removed.

Obsolete Menu Built-ins

Menus associated with full-screen display and character mode have been removed. Code that contains these Built-ins will not compile and should be recoded, although exceptions are mentioned in the following table. Built-ins that are equivalent to obsolete Built-ins are also noted in the table.

Table 4–1 Obsolete Menu Built-ins

Obsolete Menu Built-in	Migration Notes
Application_Menu	No migration path or replacement functionality.
Application_Parameter	No migration path or replacement functionality. See Chapter 12, "Menu Parameters".
Background_Menu<n>	No migration path or replacement functionality.
Debug_Mode	No migration path or replacement functionality. Code containing this Built-in will compile but will not provide functionality.
Disable_Item	Use SET_MENU_ITEM_PROPERTY().
Enable_Item	Use SET_MENU_ITEM_PROPERTY().
Exit_Menu	No migration path or replacement functionality.
Hide_Menu	No migration path or replacement functionality.

Table 4–1 Obsolete Menu Built-ins

Obsolete Menu Built-in	Migration Notes
Item_Enabled	Use GET_MENU_ITEM_PROPERTY(<name>, ENABLED). Item_Enabled will work in Oracle Forms, but will be removed in a future release.
Main_Menu	No migration path or replacement functionality.
Menu_Clear_Field	Use CLEAR_ITEM.
Menu_Failure	Use FORM_FAILURE flag.
Menu_Help	No migration path or replacement functionality.
Menu_Message	Use MESSAGE.
Menu_Next_Field	Use NEXT_ITEM.
Menu_Parameter	No migration path or replacement functionality. See Chapter 12, "Menu Parameters".
Menu_Previous_Field	Use PREVIOUS_ITEM.
Menu_Redisplay	No migration path or replacement functionality.
Menu_Show_Keys	Use SHOW_KEYS. The upgrade process will make this change automatically.
Menu_Success	Use FORM_SUCCESS flag.
New_Application	No migration path or replacement functionality.
New_User	Use LOGOUT and LOGON.
Next_Menu_Item	No migration path or replacement functionality.
OS_Command	Use HOST.
OS_Command1	Use HOST.
Previous_Menu	No migration path or replacement functionality.
Previous_Menu_Item	No migration path or replacement functionality.
Query_Parameter	No migration path or replacement functionality. See Chapter 12, "Menu Parameters".
Set_Input_Focus	No migration path or replacement functionality.
Show_Background_Menu	No migration path or replacement functionality.
Show_Menu	No migration path or replacement functionality.

Table 4–1 Obsolete Menu Built-ins

Obsolete Menu Built-in	Migration Notes
Terminate	No migration path or replacement functionality. See Chapter 12, "Menu Parameters".
Where_Display	No migration path or replacement functionality.

Other Obsolete Built-ins

The following Built-ins have been removed. Code that contains these Built-ins will not compile and should be recoded, although exceptions are mentioned in the following table. Built-ins that are equivalent to the obsolete Built-ins are also noted in the table.

Table 4–2 Other Obsolete Built-ins

Obsolete Built-in	Migration Notes
BLOCK_MENU	No migration path or replacement functionality.
BREAK	Migrate to DEBUG.SUSPEND.
CALL	Use CALL_FORM.
CHANGE_ALERT_MESSAGE	Use SET_ALERT_PROPERTY(..., ALERT_MESSAGE_TEXT,...);
DISPATCH_EVENT	Applied only to OLE and OCX items. Therefore, no migration path or replacement functionality.
(FORMS_OLE.) ACTIVATE_SERVER (FORMS_OLE.)CLOSE_SERVER (FORMS_OLE.)EXEC_VERB (FORMS_OLE.)FIND_OLE_VERB (FORMS_OLE.)GET_INTERFACE_POINTER (FORMS_OLE.)GET_VERB_COUNT (FORMS_OLE.)GET_VERB_NAME (FORMS_OLE.)INITIALIZE_CONTAINER (FORMS_OLE.)SERVER_ACTIVE	No migration path or replacement functionality.

Table 4–2 Other Obsolete Built-ins

Obsolete Built-in	Migration Notes
MACRO	No migration path or replacement functionality.
OHOST	Use HOST.
PLAY_SOUND	No migration path or replacement functionality.
READ_SOUND_FILE	No migration path or replacement functionality.
ROLLBACK_FORM	CLEAR_FORM(NO_COMMIT,FULL_ ROLLBACK)
ROLLBACK_NR	CLEAR_FORM(NO_COMMIT,FULL_ ROLLBACK)
ROLLBACK_RL	CLEAR_FORM(NO_COMMIT,FULL_ ROLLBACK)
ROLLBACK_SV	CLEAR_FORM(NO_COMMIT,FULL_ ROLLBACK)
RUN_PRODUCT	Valid only for integration with Oracle Graphics. Use RUN_REPORT_OBJECT for integration with Oracle Reports. For all other uses, code will compile but will generate run-time errors.
VBX.FIRE_EVENT VBX.GET_PROPERTY VBX.GET_VALUE_PROPERTY VBX.INVOKE_METHOD VBX.SET_PROPERTY VBX.SET_VALUE_PROPERTY	No migration path or replacement functionality.
WRITE_SOUND_FILE	No migration path or replacement functionality.

Obsolete Built-in Packages

The following Built-in packages have been removed. Code that contains these packages will not compile and should be recoded, although exceptions are mentioned in the following table. Packages that are equivalent to the obsolete packages are also noted in the table.

Table 4–3 Obsolete Built-in Packages

Obsolete Package	Migration Notes
DEBUG	No migration path or replacement functionality because there is a new debugger. DEBUG.ATTACH and DEBUG.SUSPEND are still supported.
PECS	No migration path. Use Forms Trace and Oracle Trace, which are documented in the book *Oracle Application Server Forms Services Deployment Guide (Part No. A92175-01).*

Obsolete Constants

The following constants used in the GET_ITEM_PROPERTY and SET_ITEM_PROPERTY Built-ins have been removed. Code that contains these constants will not compile and should be recoded, although exceptions are mentioned in the following table.

Table 4–4 Obsolete Constants

Obsolete Constant	Migration Notes
DATE_FORMAT_COMPATIBILITY mode	Used by GET_APPLICATION and SET_APPLICATION properties. This constant is ignored.
COMPRESSION_OFF COMPRESSION_ON HIGHEST_SOUND_QUALITY HIGH_SOUND_QUALITY LOW_SOUND_QUALITY LOWEST_SOUND_QUALITY MEDIUM_SOUND_QUALITY MONOPHONIC ORIGINAL_QUALITY ORIGINAL_SETTING	No migration path or replacement functionality.

Table 4–4 Obsolete Constants

Obsolete Constant	Migration Notes
POPUPMENU_CUT_ITEM	No migration path or replacement functionality.
POPUPMENU_COPY_ITEM	
POPUPMENU_DELOBJ_ITEM	
POPUPMENU_INSOBJ_ITEM	
POPUPMENU_LINKS_ITEM	
POPUPMENU_OBJECT_ITEM	
POPUPMENU_PASTE_ITEM	
POPUPMENU_PASTESPEC_ITEM	
SHOW_FAST_FORWARD_BUTTON	
SHOW_PLAY_BUTTON	
SHOW_POPUPMENU	
SHOW_RECORD_BUTTON	No migration path or replacement functionality.
SHOW_REWIND_BUTTON	
SHOW_SLIDER	
SHOW_TIME_INDICATOR	
SHOW_VOLUME_CONTROL	
STEREOPHONIC	

Obsolete Syntax

Using the ampersand (&) as a functional equivalent to NAME_IN() is now obsolete.

5

Triggers

In order to streamline the tools available and simplify the development process for building Forms applications for the Web, triggers that are not applicable to Web deployment have been removed. In addition, the functionality of some triggers is being more strictly enforced.

Obsolete Triggers

Table 5–1 Obsolete Triggers

Obsolete Triggers	Migration Notes
ON-DISPATCH-EVENT	Applies only to OLE and OCX items. Therefore, no migration path or replacement functionality.
All V2-style triggers	When you open FMBs that contain V2-style triggers, the triggers are dropped and a warning message lists the names of the dropped triggers. You should recode V2-style triggers into PL/SQL in Forms 6*i*, before upgrading to 9*i*.

Stricter Enforcement of Triggers

The use of the following triggers is more strictly enforced. These triggers will not execute if they are used incorrectly.

Table 5–2 Triggers with Restricted Usage

Trigger	Restricted Usage
WHEN-CLEAR-BLOCK	Allowed at the Block and Form level only. No longer allowed at the Item level.
WHEN-CREATE-RECORD	
WHEN-DATABASE-RECORD	
WHEN-NEW-RECORD-INSTANCE	
WHEN-REMOVE-RECORD	
WHEN-NEW-FORM-INSTANCE	Allowed at the Form level only. No longer allowed at the Block and Item level.

6

Properties

In order to streamline the tools available and simplify the development process for building Forms applications for the Web, properties that are not applicable to Web deployment have been removed.

Obsolete Properties

Many properties, including those associated with character mode and menus, have been removed. When you open a form that contains these properties, the properties are ignored and do not appear in Forms Developer. Except as noted, code that attempts to use these properties at runtime will fail. Refer to Table 6–1, "Obsolete Properties" for more information.

Table 6–1 Obsolete Properties

Obsolete Property	Applies to	Migration Notes
Character Mode Logical Attribute	items, canvases, etc.	
Command Type	menu items	**Note:** This property is partially obsolete. The only valid values are Null, PL/SQL, and Menu. If your menu module uses Plus, Form, or Macro, which are no longer valid values, the values are replaced by the following PL/SQL code in the Command Text property: Plus: /* HOST('plus80 <old_code>'); */ null; Form: /* CALL_FORM(<old_code>); */ null; Macro: /* MACRO: <old_code> ; */ null; where <old_code> is the value of the Command Text property before migration. The replacement PL/SQL code is commented out so that you can replace the original code with new PL/SQL code.
Data Block Description	blocks	
Fixed Length	items	Use a format mask with the relevant number of placeholders to limit or control the length of data entered for an item.
Help Description	menu items	
Listed in Data Block Menu	blocks	
List Type	LOVs	Because all LOVs are now based on record groups, this property is obsolete.
Menu Source	forms	A value of Database is no longer valid. File is the only valid value for this property, which indicates that at runtime, Forms will use the normal search path to locate the MMX file.

Table 6–1 Obsolete Properties

Obsolete Property	Applies to	Migration Notes
Runtime Compatibility Mode	forms	Ignored at runtime. 5.0 behavior is always used. (See the Forms Developer online help for a description of runtime behavior.) To allow WHEN-VALIDATE-ITEM to run for NULL items, specify 4.5 for the DEFER_REQUIRED_ENFORCEMENT property. (If your Forms application used "4.5" as the Runtime Compatibility Mode property setting, the Oracle Forms Migration Assistant will automatically set the Defer Required Enforcement property to "4.5".)
Trigger Style	triggers	All triggers are now PL/SQL triggers.
White on Black	items, canvases, etc.	

7

Changes to Client-Server Deployment and Forms Runtime

In order to streamline the tools available and simplify the development process for building Forms applications for the Web, client-server runtime is obsolete in Forms Developer and Forms Services, as it is not applicable to Web deployment.

If you currently deploy applications in a client-server environment and do not want to switch to Web-based deployment, you must remain on Forms 6*i*. When you use Forms Developer, part of the migration process is to upgrade your Forms applications for Web-based deployment.

See Chapter 15, "Migrating Client-Server Applications to the Web" for information about how client-server deployment differs from Web-based deployment.

Effect on Forms Development

The obsolescence of client-server deployment will have little to no effect on the development and debugging of Forms applications. You can still run your code in Forms Developer without having to deploy on the Web first. Use the run-on-Web facility, which renders a true WYSIWYG representation of a Web-deployed form.

The PL/SQL debugger has been improved to allow debugging in a three-tier environment.

Obsolete Forms Runtime Command Line Options

The following command line options for Runform have been removed because they relate to obsolete features:

- OptimizeSQL
- OptimizeTP
- Keyin
- Keyout
- Output_file
- Interactive
- Block_menu
- Statistics

Obsolete Character Mode Runtime

Character mode runtime, which was only available on UNIX and VMS platforms, is no longer available. All character mode support has been removed from Forms Developer and Forms Services. See Chapter 6, "Properties" and Chapter 8, "Logical and GUI Attributes" for more information.

8

Logical and GUI Attributes

In order to streamline the tools available and simplify the development process for building Forms applications for the Web, logical and GUI attributes that are not applicable to Web deployment have been removed.

Use Visual Attributes to Replace Logical and GUI Attributes

For Web-deployed forms, you can use Visual Attributes in place of logical and GUI attributes to define the appearance of dynamic items.

Obsolete Logical and GUI Attributes

Replace any references to the following obsolete logical and GUI attributes in SET_ITEM_PROPERTY, SET_FIELD, or DISPLAY_ITEM with an equivalent Visual Attribute.

Table 8–1 Obsolete Logical and GUI Attributes

Obsolete Attribute	Where Used and Migration Notes
Alert	Alert text.
AlertBackground	Alert background.
AlertIcon	Icon in an alert window.
AlertMessage	Message text in an alert window.
Boilerplate	Constant text.
Bold	Bold for all items (including check boxes).

Table 8–1 Obsolete Logical and GUI Attributes

Obsolete Attribute	Where Used and Migration Notes
Bold-inverse	Inverse bold for all items.
Bold-text	Boilerplate.
Button-current	Current button.
Button-non-current	Non-current button.
Field-current	Color for current text item.
Field-non-current	Color for text item that is not currently selected.
Field-Queryable	Queryable field in Enter-Query mode.
Field-selected-current	Currently selected text item.
Field-selected-non-current	Text item that is not currently selected.
Full-screen-title	Screen title.
ItemQueryDisabled	When a Block goes into Enter-Query Mode, any non-queryable items will inherit this set of attributes.
ListItemNonSelect	Unselected item in a text list.
ListItemSelect	Selected item in a text list.
ListPrefix	List prefix.
Listtitle	List of Values (LOV) title.
Menu	Selected menu.
Menu-bottom-title	Current title at bottom of menu.
MenuItemDisabled	Disabled menu item.
MenuItemDisableMnemonic	Mnemonic of a disabled menu item.
MenuItemEnable	Enabled, non-current menu item.
MenuItemEnableMnemonic	Mnemonic of an enabled menu item.
MenuItemSelect	Current menu item.
MenuItemSelectMnemonic	Mnemonic of the current menu item.
Menu-subtitle	Current menu subtitle.

Table 8–1 Obsolete Logical and GUI Attributes

Obsolete Attribute	Where Used and Migration Notes
Menu-title	Current menu title.
Normal	Text item.
NormalAttribute	Normal background for windows.
PushButtonDefault	Default or current button.
PushButtonNonDefault	Button that is not default.
Scroll-bar-fill, Inverse, Inverse-underline, Bold-underline, Bold-inverse-underline	These logical attributes are not unique to Forms Developer. As a result, these logical attributes can be overridden by the visual attributes defined by the window manager.
ScrollThumb	Elevator box on scroll bar.
Status-Empty	Controls the look of the empty Status Line.
Status-Hint	Controls the font of any item hint appearing on the Status Line.
Status-Items	Controls the look of the Operator Information Area which contains the LOV lamp, record count, etc.
Status-Message	Controls the font of any message appearing on the Status Line.
Sub-menu	Selected submenu.
TextControlCurrent	Current field or text editor.
TextControlFailValidation	When an item fails a validation check, it will be set to this attribute set.
TextControlNonCurrent	Disabled or non-current field or text editor.
TextControlSelect	Selected text in an enabled field or text editor.
ToolkitCurrent	Generic attribute.
ToolkitCurrentMnemonic	Generic attribute.
ToolkitDisabled	Generic attribute.

Table 8–1 Obsolete Logical and GUI Attributes

Obsolete Attribute	Where Used and Migration Notes
ToolkitDisabledMnemonic	Generic attribute.
ToolkitEnabled	Generic attribute.
ToolkitEnabledMnemonic	Generic attribute.
Underline	Underline for all items.
WindowTitleCurrent	Title of active window.

9

Item Types

In order to streamline the tools available and simplify the development process for building Forms applications for the Web, item types that are not applicable to Web deployment have been removed.

Item Types Specific to Operating Systems

The following item types are operating-system specific and are obsolete in Forms Developer and Forms Services. These items are not removed by the migration process. However, any modules that contain them will not compile. Use JavaBeans and Pluggable Java Components for equivalent functionality.

Table 9–1 Obsolete Item Types

Item Type	Migration Notes
VBX	Was applicable to 16-bit Windows platforms only. No migration path or replacement functionality.
OLE Container	Was applicable to Windows platforms only. Programmatic OLE interaction is supported with external OLE servers on the middle tier.
OCX/ActiveX Controls	Was applicable to Windows platforms only. JavaBean support provides similar functionality.
Sound	No migration path. JavaBeans can be used to provide equivalent functionality.

10

List of Values (LOVs)

In order to streamline the tools available and simplify the development process for building Forms applications for the Web, List of Values (LOVs) that are not applicable to Web deployment have been removed.

Obsolete List of Values (LOVs)

LOVs based on record groups are still valid.

"Old-style" LOVs (V2.3-style LOVs) are obsolete in Oracle Forms.

When forms that contain old-style LOVs are migrated to Forms Developer, the old-style LOVs' Old LOV Text property, which refers to a table and column (such as EMP.ENAME), is converted to a "new-style" LOV by creating a record group based on a query (select <column> from <table>). The new-style LOV is based on the new record group.

11

User Exits

As a result of removing old-style (V2) triggers, V2 user exits have also been removed.

Obsolete V2 User Exits

The following user exits, which are hard-coded callbacks to V2 trigger functionality, have been removed. It is now assumed that any calls to these user exits is calling a user-defined user exit rather than an Built-in one. Therefore, any code that attempts to locate them is directed to the user exit with the same name in the IAPXTB structure. (IAPXTB is a file that acts as an index to locate each of the user exits that you create at runtime.)

You should recode any code that uses these callbacks to PL/SQL:

- COPY
- ERASE
- HOST
- EXEMACRO
- EZ_GOREC
- EZ_CHKREC

12

Menu Parameters

In order to streamline the tools available and simplify the development process for building Forms applications for the Web, menu parameters, which are not applicable to Web deployment, have been removed.

All menu parameters will be removed from your applications when you migrate to Oracle Forms.

Predefined Menu Parameters

Predefined menu parameters have names like UN and PW. Using predefined menu parameters, you were able to refer to bind variables, for example, :UN and :PW in PL/SQL code attached to menu items.

When migrating to Forms Developer, use the recommended Built-ins as replacements for the obsolete predefined menu parameters in the following table below:

Table 12–1 Obsolete Predefined Menu Parameters

Obsolete Parameter	Recommended Built-in
:UN	GET_APPLICATION_PROPERTY(USERNAME)
:PW	GET_APPLICATION_PROPERTY(PASSWORD)
:LN	GET_APPLICATION_PROPERTY(USER_NLS_LANG)
:AD	GET_FORM_PROPERTY(NAME_IN('SYSTEM.CURRENT_FORM'),FILE_NAME)
:SO	:SYSTEM.TRIGGER_MENUOPTION

Table 12–1 Obsolete Predefined Menu Parameters

Obsolete Parameter	Recommended Built-in
:TT	Only relevant in a character mode environment. This parameter has no replacement.

User-Defined Menu Parameters

User-defined menu parameters are obsolete for Oracle Forms. Any menu item that calls the MENU_PARAMETER or APPLICATION_PARAMETER Built-ins allowed you to define values for menu parameters.

At runtime, an un-customizable Query Parameters dialog box would let you inspect or change the values of menu parameters. Built-ins associated with Query Parameter dialogs, such as TERMINATE, are obsolete as well. See Chapter 4, "Built-ins, Packages, Constants, and Syntax" for more information.

Therefore, to replace obsolete user-defined parameters, manually redefine them as Global variables (:GLOBAL). The initial value property of parameters can be emulated by initializing your replacement Global variables in your Menu startup code.

For other features, such as the dialog box that pops up using the MENU_PARAMETER Built-in, there is no replacement functionality, although you can emulate this functionality by building a dialog using Forms.

13

Java-Related Issues

This chapter describes migration steps to take if your Forms applications use Java-related components.

Using oracle.ewt Classes in Pluggable Java Components

Pluggable Java Components (PJCs) and JavaBeans use classes that are part of the oracle.ewt framework. When upgrading to Oracle Forms, there are steps you must take to ensure equivalent functionality.

- Be sure to set the following:

  SET FORMS90_BUILDER_CLASSPATH=%ORACLE_HOME%\forms90\java\f90all.jar

- The Oracle Forms JAR files (f90all.jar and f90all_jinit.jar) only contain the ewt classes, which are required by the Forms Java Client. Therefore, PJCs that had been used with Forms 6*i* may fail at runtime in Oracle Forms because the classes that were available in Forms 6*i* can no longer be located. Missing oracle.ewt classes are available in ewt.jar, which is supplied with Oracle9*i* JDeveloper.
- Because of security constraints related to Java 1.3, which is used by Oracle Forms, you must re-sign all of the classes (including the Forms classes) with your own certificate. Java 1.3 requires that all classes from the same package be signed with the same certificate. Therefore, if you need to use extra classes from ewt.jar in combination with f90all.jar, you must re-sign all of the classes with your own certificate.
- The sample PJCs and JavaBeans provided by Oracle are re-coded to use Swing classes or oracle.ewt classes. The sample PJCs and JavaBeans that are provided with Oracle Forms will not encounter this problem.

JDK Versions and Font-Rendering Issues

When migrating Forms applications from JDK 1.1 to JDK 1.3 or higher, you may encounter font height changes. This is because the code that renders fonts underwent significant changes from JDK 1.1 to JDK 1.3. As a result of these changes, the font height for logical fonts of the same size increased in JDK 1.3. For example, a dialog font of size 12 points has a height of 15 points in JDK 1.1 and a height of 17 points in JDK 1.3.

In Forms applications, the font size changes may affect labels, which can overlap text fields. One possible workaround is to set the following applet parameter to "yes":

```
<PARAM NAME = "mapFonts" VALUE = "yes" >
```

After making this change, check the appearance of the font size to be sure it is acceptable. You may still need to modify the form if this workaround does not provide acceptable font sizes. See the *Oracle Application Server Forms Services Deployment Guide* for details.

Another workaround is that when the font is unspecified, the default font name and size in Registry.dat will be used. The default font in registry.dat is Dialog with a default size of 900. The size of this font can be modified to a smaller value in the registry.dat file. In cases where the font is not specified, you can work around the problem without modifying the form. However, use caution because it will modify the font size for the entire application.

14

Integrating Reports and Graphics

Oracle Graphics6*i* is no longer shipped with Oracle Forms. In addition, the Charting wizard has been removed from Forms. This chapter describes how you can call existing Graphics6*i* and Reports applications from a form.

Oracle Graphics6*i*

You cannot create new charts because Oracle Graphics6*i* is no longer shipped with Oracle Forms. However, you can display existing charts in Forms and Reports applications that have been migrated.

The following whitepapers at http://otn.oracle.com/products/forms provide additional information:

- *Deploying Oracle Graphics in Oracle9iAS V2 Forms Services: Best practices for integrating Oracle Graphics in Forms on the Web*
- *Deploying Interactive Charts on the Web: Migrating from the Graphics Cartridge*
- *Integrating Oracle9iAS Reports in Oracle9iAS Forms*

Displaying Existing Charts in Oracle Forms

If your form contains embedded Oracle Graphics6*i* displays, you can still display them in Oracle Forms by doing the following:

- Use the Oracle Graphics (OG) package wrappers
- Be sure that Oracle Graphics6*i* is installed on the same machine as Forms Services, but in a separate ORACLE_HOME, for example, OG6I_HOME.
- On Windows, set the registry variable ORACLE_GRAPHICS6I_HOME to the ORACLE_HOME directory where Graphics6*i* is installed. This variable must be

set in the registry key that contains other Oracle Forms registry variables. The version of Graphics6*i* must include Developer 6*i* Patch 8 or higher.

- On UNIX, set the environment variable ORACLE_GRAPHICS6I_HOME to the ORACLE_HOME directory where Graphics6*i* is installed. The version of Graphics6*i* must include Developer 6*i* Patch 8 or higher.
- If you use OG.PLL to run Graphics6*i*, then you must open and compile OG.PLL in Oracle9*i* Forms Developer before you can use it. OG.PLL also needs to be included in FORMS90_PATH.
- Set the path definition in the Oracle Forms (**not** Forms 6*i*) servlet environment to include Graphics in the default.env file as follows:

  PATH=%FORMS9I_HOME%\bin;%ORACLE_GRAPHICS6I_HOME%\bin;

  Note: You must also add the PATH and GRAPHICS60_PATH variables to the default.env file in Oracle Forms. You add these variables with Enterprise Mangager Application Server Control. For information on how to do this, see Chapter 4 of the *Oracle Application Server Forms Services Deployment Guide* part number B10470-01.

  Note: The following registry entries are read directly from the Graphics 6*i* ORACLE_HOME registry:

  - GRAPHICS60 = %ORACLE_HOME%\TOOLS\DBTAB60\GRAPH60
  - DE60 = %OG6I_HOME%\TOOLS\COMMON60
  - GRAPHOGD60 = %ORACLE_ HOME%\TOOLS\DBTAB60\GRAPH60\GWIZ_OGD
  - MM60 = %OG6I_HOME%\TOOLS\COMMON60
  - TK60 = %OG6I_HOME%\TOOLS\COMMON60
  - UI60 = %OG6I_HOME%\TOOLS\COMMON60
  - VGS60 = %OG6I_HOME%\TOOLS\COMMON60
  - OCL60 = %ORACLE_HOME%\TOOLS\DBTAB60\GRAPH60

Editing Existing Chart Items

You can edit the properties of existing Oracle Forms chart items, create new Oracle Forms chart items in Forms Developer, and populate an Oracle Forms chart item with an OGD file that was created in Oracle Graphics6*i* Developer. However, future releases of Forms may further restrict Graphics6*i* functionality.

Since Oracle Graphics was not originally conceived for Web deployment, issues may arise when forms with embedded charts are re-deployed from a client-server environment to the Web, particularly more complex charts with more complex character sets. In some cases, text dimensions may alter, causing text items to overlap or to extend over the edge of a display. These issues, particularly concerning scaling, are akin to the side effects that can occur when porting applications from one platform to another. Recommendations on how to deal with these issues can be found in *Deploying Oracle Graphics in Oracle9iAS V2 Forms Services* at http://otn.oracle.com/.

Migration to the Web of forms with embedded Oracle Graphics displays may well entail modifications to modules. Allow time to check the behavior of each embedded chart on the Web, and decide if any of the suggestions in the white paper need to be implemented. See how the chart looks with the largest set of data it is likely to be passed, not just a small set of test data (good practice whether running in a client-server environment or on the Web). Consider whether reports deployment may be preferable in some cases. Review the charts to see if there are any items that can be removed, particularly redundant axes labels or legends. If you are migrating forms with embedded OGDs from client-server deployment to Oracle9iAS on the Web, it would be best to break this down into two stages. First migrate to Forms Services 6i Web deployment and then upgrade to Oracle9iAS. In this way, you can deal with any Oracle Graphics Web-migration issues in the 6i environment in which you still have Graphics Builder and the Chart Wizard. Nevertheless, in the second stage, the upgrade to Oracle9iAS, check out the appearance of your charts once again.

About Integration with Oracle Reports

You can embed new and existing Reports applications in forms that have been migrated to Oracle Forms.

You can no longer use the Reports client runtime engine to output Reports in the Web. From Forms50 forward, the RUN_REPORT_OBJECT Built-in is available in Forms Developer to run integrated reporting in Forms. The RUN_PRODUCT Built-in is obsolete in Oracle Forms for running Oracle Reports applications from Oracle Forms.

Oracle9*i* Forms and Oracle Reports are now Web-based only, and do not have client/server runtime engines. Therefore, integrated reports in Oracle Forms applications must be recoded to use the RUN_REPORT_OBJECT Built-in and Oracle Application Server Reports Services.

For additional information about integrating Oracle Forms with Reports, see *Integrating Oracle9iAS Reports in Oracle9iAS Forms* at the Oracle Technology Network at http://otn.oracle.com.

Displaying Reports in Oracle Forms

If your form contains embedded Oracle Reports applications, you can migrate the form to Oracle Forms by changing the integrated call to Oracle Reports to use:

- RUN_REPORT_OBJECT Built-in (Do not use the RUN_PRODUCT Built-in to call Reports.)
- WEB.SHOW_DOCUMENTBuilt-in

Using RUN_PRODUCT in Oracle9*i* Forms to run integrated Oracle Reports is no longer supported in Oracle9*i* Forms. The Oracle9*i* Forms Migration Assistant is provided to help you migrate your applications to use Run_Report_Object. See Chapter 2, "Using the Oracle Forms Migration Assistant" for more information.

Example

The following example runs a report using the RUN_REPORT_OBJECT Built-in. The report_object node defined in Forms Developer is assumed to be "report_node1". A user-defined Reports parameter "p_deptno" is passed by Forms using the value in the "dept.deptno" field. The Reports parameter form is suppressed. For additional details about the logic used in this example, see Notes About the Example.

```
/* The following example runs a report using the RUN_REPORT_OBJECT Built-in. The
report_object node defined in Forms Developer is assumed to be "report_node1". A
user-defined Reports parameter "p_deptno" is passed by Forms using the value in
the "dept.deptno" field. The Reports parameter form is suppressed */

 DECLARE
 v_report_id Report_Object;
 vc_report_job_id  VARCHAR2(100); /* unique id for each Report  request */
 vc_rep_status      VARCHAR2(100); /* status of the Report job */

 BEGIN
 /* Get a handle to the Report Object itself. */
 v_report_id:= FIND_REPORT_OBJECT('report_node1');
 SET_REPORT_OBJECT_PROPERTY(v_report_id,REPORT_COMM_MODE, SYNCHRONOUS);
 SET_REPORT_OBJECT_PROPERTY(v_report_id,REPORT_DESTYPE,CACHE);

 /* Define the Report output format and the name of the Reports Server as well
```

```
as a user-defined parameter, passing the department number from Forms to the
Report. The Reports parameter form is suppressed by setting  paramform  to "no".
*/
 SET_REPORT_OBJECT_PROPERTY(v_report_id,REPORT_DESFORMAT,
'<HTML|HTMLCSS|PDF|RTF|XML|DELIMITED>');
 /* replace <ReportServerTnsName> with the name of the Reports9i Services as
defined in your tnsnames.ora file */
 SET_REPORT_OBJECT_PROPERTY(v_report_id,REPORT_SERVER, '<ReportServerTnsName>');
 SET_REPORT_OBJECT_PROPERTY(v_report_id,REPORT_OTHER, 'p_
deptno='||:dept.deptno||'paramform=no');
  /* finally, run the report and retrieve the Reports job_id as a handle to the
Reports process */
 vc_report_job_id:=RUN_REPORT_OBJECT(report_id);

 /*The report output is not delivered automatically to the client, which is okay
because the Web is a request model. Thus the next step is to check if the report
finished. */

 vc_rep_status := REPORT_OBJECT_STATUS(vc_report_job_id);
 IF vc_rep_status='FINISHED' THEN
 /* Call the Report output to be displayed in a separate browser window. The URL
for relative addressing is only valid when the Reports Server is on the same
host as the Forms Server. For accessing a Remote Reports Server on a different
machine, you must use the prefix http://hostname:port/ */
 web.show_document ('/<virtual path>/<reports cgi or servlet name>/getjobid='||
vc_report_job_id ||'?server='|| '<ReportServerTnsName>','_blank');
 ELSE
 message ('Report failed with error message '||rep_status);
 END IF;
END;
```

Notes About the Example

- Calling a report synchronously makes the user wait while the report gets processed on the server. For long-running Reports, it is recommended that you start the report asynchronously, by setting the REPORT_COMM_MODE property to asynchronous and the REPORT_EXECUTION_ MODE to batch.

  ```
  SET_REPORT_OBJECT_PROPERTY(report_id,REPORT_EXECUTION_MODE,BATCH);
  SET_REPORT_OBJECT_PROPERTY(report_id,REPORT_COMM_MODE,ASYNCHRONOUS);
  ```

- After calling the RUN_REPORT_OBJECT Built-in, you must create a timer to run frequent checks on the current REPORT_OBJECT_STATUS using a When-Timer-Expired trigger. For performance reasons, the timer should not

fire more than four times a minute. After the report is generated, the When-Timer-Expired trigger calls the WEB.SHOW_DOCUMENT Built-in to load the Reports output file, identified by a unique job_id, to the client's browser.

Note: Do not forget to delete the timer when it is no longer needed.

The following example shows the When-Timer-Expired trigger that checks for the Report_Object_Status.

```
(...)
 /* :global.vc_report_job_id needs to be global because the information
about the Report job_id is shared between the trigger code that starts the
Report and the When-Trigger-Expired trigger that checks the current Report
status. */
 vc_rep_status:= REPORT_OBJECT_STATUS(:global.vc_report_job_id);
 IF vc_rep_status='FINISHED' THEN
 web.show_document ('/<virtual path>/<reports cgi or servlet
name>/getjobid='|| vc_report_job_id ||'?server='||
'<ReportServerTnsName>','_blank');
 ELSIF vc_rep_status not in ('RUNNING','OPENING_REPORT','ENQUEUED') THEN
 message (vc_rep_status||'  Report output aborted');
  END IF;
(...)
```

Using Parameter Lists in RUN_REPORT_OBJECT

Parameter lists that were used with RUN_PRODUCT in client-server mode can also be used with RUN_REPORT_OBJECT calling Oracle Application Server Reports Services. System parameters must be set by the Set_Report_Object_Property. The syntax for using parameter lists in RUN_REPORT_OBJECT is as follows:

```
report_job_id:=run_report_object(report_id,paramlist_id);
```

where paramlist_id is the same id used with RUN_PRODUCT.

Parameter settings can be any of the following:

- REPORT_COMM_MODE: Batch, Runtime
- REPORT_EXECUTION_MODE: Synchronous, Asynchronous
- REPORT_DESTYPE: File, Printer, Mail, Cache
- REPORT_FILENAME: The report filename (not used with CACHE)
- REPORT_DESNAME: The report destination name (not used with Cache)

- REPORT_DESFORMAT: The report destination format
- REPORT_SERVER: The report server name

Other settings are as follows:

- Reports9i CGI name is "rwcgi90" (UNIX) and "rwcgi90.exe" (Windows NT)
- Reports Servlet default name is "rwservlet"
- Reports Servlet virtual path is /reports/

Migration Steps

You can use the Oracle9*i* Forms Migration Assistant described in Chapter 2, "Using the Oracle Forms Migration Assistant" to change integrated Reports calls in your Oracle Forms modules. The Oracle9*i* Forms Migration Assistant adds code to your application modules that redirects Run_Product calls to Reports and uses the Run_Report_Object Built-in and Reports Services. The resulting conversion is of the same quality as using Run_Product and the runtime engine in Forms 6*i*.

To manually migrate Reports in Oracle Forms, do the following:

1. Find all occurrences of Run_Product.
2. Identify and locate the parameter lists used with these calls.
3. Remove all of the Reports system parameter settings like desname and destype from the parameter lists.
4. Find the Reports node ID for the Reports node name defined in Forms Developer or the Forms 6*i* Builder.
5. Create Set_Report_Object_Property codes for DESNAME, REPORT_SERVER, DESFORMAT, DESTYPE, COMM_MODE, and EXECUTION:MODE in your PL/SQL.
6. Use Run_Report_Object(report_node_id, paramlist_id) to reuse your parameter lists that had been created for Run_Product.

   Note: Detailed information about how to change calls to Oracle Reports in Forms 6*i* to use Run_Report_Object are provided in a whitepaper on http://otn.oracle.com.

15

Migrating Client-Server Applications to the Web

If you are currently using the client-server version of Forms Server, migrating applications to Forms Services for the Web is straightforward. This chapter briefly describes the differences between client-server and Web implementations, and then gives guidelines to migrate your current applications from client-server-based to Web-based Forms Services. In addition, you will find white papers and other resources at http://otn.oracle.com.

Client-Server-Based Architecture

In the client-server-based implementation, shown in Figure 15–1, the Forms Server Runtime Engine and all application logic are installed on the user's desktop

machine. All user interface and trigger processing occurs on the client, except for database-server-side triggers and logic that may be included in some applications.

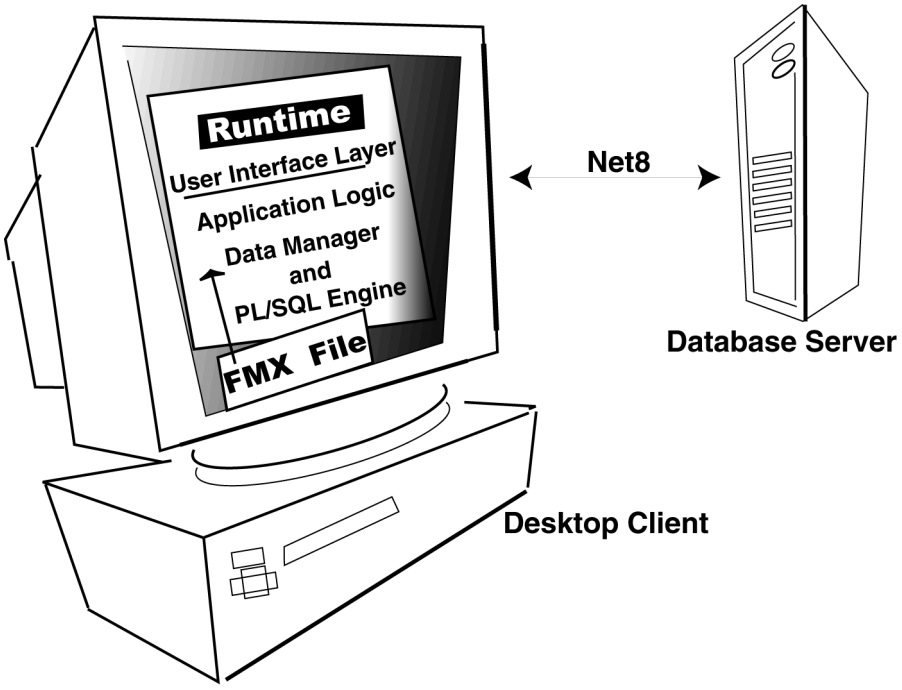

Figure 15–1 Legacy Forms Server client-server-based architecture

Web-Based Architecture

In a Web-based implementation, shown in Figure 15–2, the Forms Services Runtime Engine and all application logic are installed on application servers, and not on client machines. All trigger processing occurs on the database and application

servers, while user interface processing occurs on the Forms client, located on users' machines.

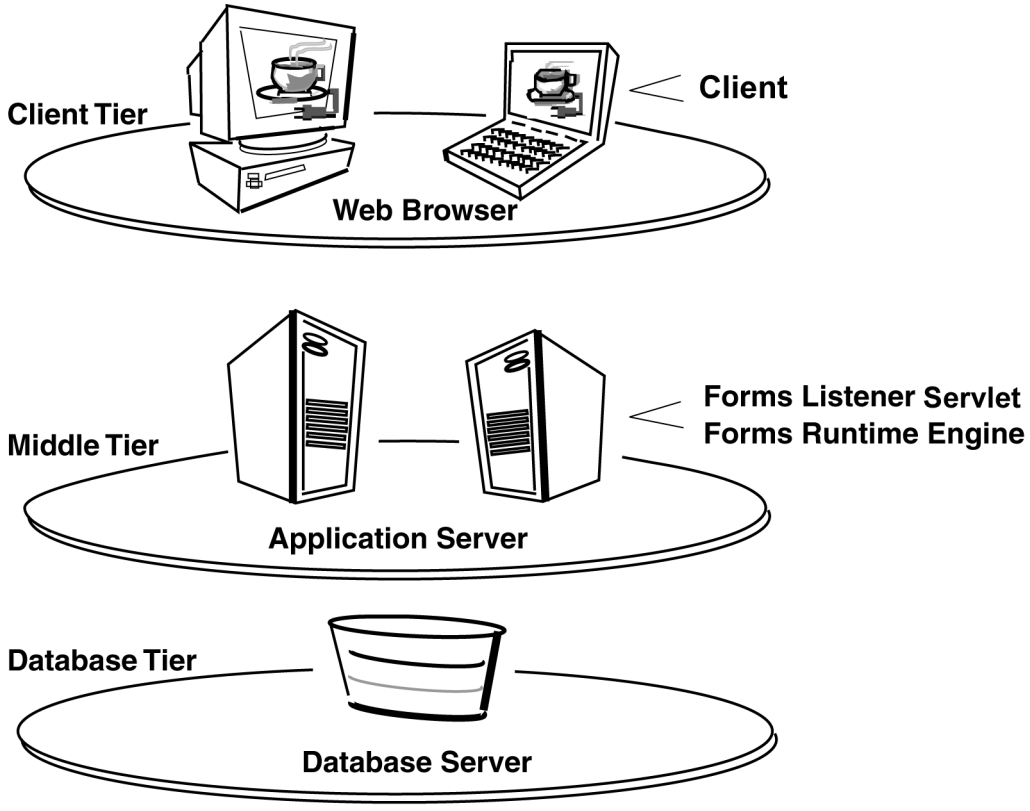

Figure 15–2 Forms Services Web-based architecture

Who Should Read this Chapter?

This chapter will be useful to you if the following statements apply to your deployment environment:

- You currently deploy Web-based Forms Developer applications.
- You use Oracle Application Server for Web server support.

Guidelines for Migration

When migrating your applications from client-server deployment to the Web, note that a Web-based application:

- Supports JPEG and GIF image types only, so convert existing images to these formats.
- Supports the use of compressed JAR (Java Archive) files for file transfer, so use JAR files whenever the transfer of large files is required between the Forms Services and the Java client.
- Does not support ActiveX, OCX, OLE, or VBX controls in the user interface. Instead, use JavaBeans to duplicate functionality in the user interface. Any other Microsoft Windows user interface dependencies should also be replaced with JavaBeans.
- Does not support MouseMove triggers, such as When-Mouse-Enter, When-Mouse-Leave, and When-Mouse-Move.
- Does not natively support write access to the client hard drive. This can be accomplished by writing a JavaBean for the pluggable Forms user interface.
- Supports Java fonts only, so check applications for the types of fonts used. If necessary, switch to Java fonts. Java uses a font alias list, located in the Registry.dat file. The font aliases described in Table 15–1 are supported**:**

Table 15–1 Font Support for Web-based Applications

Java font	Windows font	XWindows font	Macintosh font
Courier	Courier New	adobe-courier	Courier
Dialog	MS San Serif	b&h-lucida	Geneva
DialogInput	MS San Serif	b&h-lucidatypewriter	Geneva
Helvetica	Arial	adobe-helvetica	Helvetica
Symbol	Wingdings	itc-zapfdingbats	Symbol
Times Roman	Times New Roman	adobe-times	Times Roman

- Has some Built-ins and packages that execute only in the application server, but not in the client browser:

  - TEXT_IO

  - HOST

- ORA_FFI
- GET_FILE_NAME
- READ_IMAGE_FILE
- WRITE_IMAGE_FILE

If the functionality of these Built-in and packages is required for the client, use JavaBeans as a replacement.

16

Upgrading from Pre-Forms 6*i* Applications to Oracle Forms

About Upgrading Forms

Forms Developer is upwardly compatible with earlier versions of Oracle Forms, including Versions 3.0, 4.0, 4.5, and 5.0.

Note: If you are migrating from older releases of Forms to Oracle Forms, you must first migrate your applications to Forms 6*i*, and then migrate them to Oracle Forms.

Note: Before converting your forms or menus, it is recommended that you first make backup copies of all files. Once you upgrade a module, you will not be able to open it in an earlier version of Forms Developer.

Upgrading a Form

To upgrade a version 4.x or 5.x Forms application to Forms 6*i*:

1. Start Forms 6*i*.
2. Choose **File | Open** to display the file or database dialog.
3. Choose the module you want to upgrade.
4. Click **OK**.
5. Choose **File | Save**.
6. Choose **Program | Compile | All** to compile the newly-upgraded module.

   Note: You can also use the Forms compiler (f60genm and ifcmp60) to upgrade a Formscapplication to 6*i*.

Note: All form modules and libraries must be upgraded and recompiled.

To upgrade a pre-4.0 Forms application to Forms 6*i*:

To upgrade a pre-4.0 form to Forms 6*i*, first upgrade the form to version 4.5 (using Forms 4.5 to perform the upgrade), and then upgrade from version 4.5 to version 6*i* according to the instructions above.

To upgrade a pre-4.0 form to version 4.5, type this statement at the command line, substituting the correct form of the Forms 4.5 Generate command for your environment:

```
f45gen32 <module_name> <username>/<password> upgrade=yes version=<version_
number>
```

Table 16–1 Version Number for Forms 4.5 Generate Command

To upgrade from	Use this version number
Version 3.0	30
Version 2.3	23
Version 2.0	20

PL/SQL 9 Support

Stored program units can use all the new PL/SQL 9 features.

Compatibility with Earlier Versions of PL/SQL

If you have client-side program units written in PL/SQL V1 or V2, you must convert that code to the new level. The PL/SQL V1 conversion utility is provided to help automate the upgrading process.

Stored program units can use all the new PL/SQL 9 features that are valid for use in client side PL/SQL. Certain PL/SQL features such as the supplied DBMS_LOB routines cannot be directly called from Client Side PL/SQL. Forms programs must call a stored procedure which in turn calls the routine in question in these restricted cases.

Forms Developer Runtime Behavior

The default runtime behavior of forms created with Forms 5.0 through 6*i* differed from runtime behavior for Forms 4.5. The form-level Runtime Compatibility Mode

property could be set to "4.5" to provide Forms release 4.5 behavior. (This happened by default in forms that were upgraded from Forms release 4.5.)

Starting with Forms Developer, 5.0 behavior is used in all cases, and the form-level property, Runtime Compatibility Mode is ignored.

If you are upgrading a form that specifies 4.5 behavior to Oracle Forms, you must alter the logic, as necessary, to reflect the differences between 4.5 and 5.0 behavior. See the Forms 6*i* online help for information about the Runtime Compatibility Mode property and the differences between 4.5 and 5.0 behavior.

Index

Symbols

& (NAME_IN), 4-6
(FORMS_OLE.) ACTIVATE_SERVER, 4-3
(FORMS_OLE.)CLOSE_SERVER, 4-3
(FORMS_OLE.)EXEC_VERB, 4-3
(FORMS_OLE.)FIND_OLE_VERB, 4-3
(FORMS_OLE.)GET_INTERFACE_POINTER, 4-3
(FORMS_OLE.)GET_VERB_COUNT, 4-3
(FORMS_OLE.)GET_VERB_NAME, 4-3
(FORMS_OLE.)INITIALIZE_CONTAINER, 4-3
(FORMS_OLE.)SERVER_ACTIVE, 4-3

A

ActiveX controls, 9-1
AD, 12-1
advanced converter options, setting, 2-10
Alert, 8-1
AlertBackground, 8-1
AlertIcon, 8-1
AlertMessage, 8-1
ampersand, NAME_IN, 4-6
Application_Menu, 4-1
Application_Parameter, 4-1

B

Background_Menu, 4-1
BLOCK_MENU, 4-3
Block_menu, 7-2
Boilerplate, 8-1
Bold, 8-1
Bold-inverse, 8-2
Bold-inverse-underline, 8-3
Bold-text, 8-2
Bold-underline, 8-3
Built-ins, obsolete, 4-1
Built-ins, other, 4-3
Button-current, 8-2
Button-non-current, 8-2

C

CALL, 4-3
cartridge, 1-2
CGI, 1-2
CHANGE_ALERT_MESSAGE, 4-3
Character Mode Logical Attribute, 6-2
character mode runtime, obsolete, 7-2
charts, editing, 14-2
client-server architecture, 15-1
client-server migration, 15-1
client-server runtime, obsolete, 7-1
command line options, obsolete, 7-2
Command Type, 6-2
components, obsolete, 1-2
COMPRESSION_OFF, 4-5
COMPRESSION_ON, 4-5
constants, obsolete, 4-5
converter.properties file, 2-7
converter.properties, editing, 2-3
COPY, 11-1

D

Data Block Description, 6-2
DEBUG, 4-5

Debug_Mode, 4-1
Disable_Item, 4-1
DISPATCH_EVENT, 4-3

E

Enable_Item, 4-1
ERASE, 11-1
EXEMACRO, 11-1
Exit_Menu, 4-1
EZ_CHKREC, 11-1
EZ_GOREC, 11-1

F

f45gen32, 16-2
features, obsolete, 1-1
Field-current, 8-2
Field-non-current, 8-2
Field-Queryable, 8-2
Field-selected-current, 8-2
Field-selected-non-current, 8-2
Fixed Length, 6-2
FMTs, 3-1
Forms 3.0, 4.0, 4.5, 5.0, 16-1
Forms Listener, 1-2
Forms Server Cartridge, 1-2
Full-screen-title, 8-2

G

global variables, 12-2
Graphics, 1-2
Graphics registry entries, 14-2
GUI attributes, obsolete, 8-1

H

Help Description, 6-2
Hide_Menu, 4-1
HIGH_SOUND_QUALITY, 4-5
HIGHEST_SOUND_QUALITY, 4-5
HOST, 11-1
HTTPS support, 1-2

I

images, 15-4
Interactive, 7-2
Inverse, 8-3
Inverse-underline, 8-3
item types, obsolete, 9-1
Item_Enabled, 4-2
ItemQueryDisabled, 8-2

J

JARs, 15-4
Java fonts, 15-4
JavaBeans, 9-1

K

Keyin, 7-2
Keyout, 7-2

L

List of Values, obsolete, 10-1
List Type, 6-2
Listed in Data Block Menu, 6-2
ListItemNonSelect, 8-2
ListItemSelect, 8-2
ListPrefix, 8-2
Listtitle, 8-2
LN, 12-1
load balancing, 1-2
logical attributes, obsolete, 8-1
LOVs, obsolete, 10-1
LOVs, record groups, 10-1
LOVs, V2.3-style, 10-1
LOW_SOUND_QUALITY, 4-5
LOWEST_SOUND_QUALITY, 4-5

M

MACRO, 4-4
Main_Menu, 4-2
MEDIUM_SOUND_QUALITY, 4-5
Menu, 8-2
menu built-ins, 4-1

menu parameters, obsolete, 12-1
menu parameters, user-defined, 12-2
Menu Source, 6-2
Menu_Clear_Field, 4-2
Menu_Failure, 4-2
Menu_Help, 4-2
Menu_Message, 4-2
Menu_Next_Field, 4-2
Menu_Parameter, 4-2
Menu_Previous_Field, 4-2
Menu_Redisplay, 4-2
Menu_Show_Keys, 4-2
Menu_Success, 4-2
Menu-bottom-title, 8-2
MenuItemDisabled, 8-2
MenuItemDisableMnemonic, 8-2
MenuItemEnable, 8-2
MenuItemEnableMnemonic, 8-2
MenuItemSelect, 8-2
MenuItemSelectMnemonic, 8-2
Menu-subtitle, 8-2
Menu-title, 8-3
Migration Assistant, about batch mode, 2-7
Migration Assistant, about starting, 2-6
Migration Assistant, starting in batchmode, 2-8
Migration Assistant, starting in UNIX, 2-6
Migration Assistant, starting in Windows, 2-6
Migration Assistant, wizard version, 2-1
MMTs, 3-1
MONOPHONIC, 4-5
Mutiple Log Support, 2-2

N

New_Application, 4-2
New_User, 4-2
Next_Menu_Item, 4-2
Normal, 8-3
NormalAttribute, 8-3

O

obsolete components, 1-2
obsolete features, 1-1
OCA, 1-3
OCX controls, 9-1
OG6I_HOME, 14-1
OG.PLL, 14-2
OHOST, 4-4
OLE Container, 9-1
ON-DISPATCH-EVENT, 5-1
Open Client Adapters, 1-3
OptimizeSQL, 7-2
OptimizeTP, 7-2
ORACLE_GRAPHICS6I_HOME, 14-1
ORIGINAL_QUALITY, 4-5
ORIGINAL_SETTING, 4-5
OS_Command, 4-2
OS_Command1, 4-2
Output_file, 7-2

P

packages, obsolete, 4-4
parameter lists, RUN_REPORT_OBJECT, 14-6
parameters, menu, 12-1
PECS, 1-3, 4-5
Performance Event Collection Services, 1-3
PLAY_SOUND, 4-4
PL/SQL 8 support, 16-2
Pluggable Java Components, 9-1
POPUPMENU_COPY_ITEM, 4-6
POPUPMENU_CUT_ITEM, 4-6
POPUPMENU_DELOBJ_ITEM, 4-6
POPUPMENU_INSOBJ_ITEM, 4-6
POPUPMENU_LINKS_ITEM, 4-6
POPUPMENU_OBJECT_ITEM, 4-6
POPUPMENU_PASTE_ITEM, 4-6
POPUPMENU_PASTESPEC_ITEM, 4-6
Previous_Menu, 4-2
Previous_Menu_Item, 4-2
Procedure Builder, 1-3
Project Builder, 1-3
properties, obsolete, 6-1
PushButtonDefault, 8-3
PushButtonNonDefault, 8-3
PW, 12-1

Q

Query Builder, 1-3
Query Parameter dialogs, 12-2
Query_Parameter, 4-2

R

READ_SOUND_FILE, 4-4
registry entries, integrating Graphics, 14-2
REPORT_COMM_MODE, 14-6
REPORT_DESFORMAT, 14-7
REPORT_DESNAME, 14-6
REPORT_DESTYPE, 14-6
REPORT_EXECUTION_MODE, 14-6
REPORT_FILENAME, 14-6
REPORT_SERVER, 14-7
Reports integration, 14-3
ROLLBACK_FORM, 4-4
ROLLBACK_NR, 4-4
ROLLBACK_RL, 4-4
ROLLBACK_SV, 4-4
RUN_PRODUCT, 4-4
RUN_REPORT_OBJECT, 14-4
RUN_REPORT_OBJECT and parameter lists, 14-6
Runform, obsolete command line options, 7-2
Runtime Compatibility Mode, 6-3

S

Schema Builder, 1-3
Scroll-bar-fill, 8-3
ScrollThumb, 8-3
search_replace.properties, editing, 2-4
Set_Input_Focus, 4-2
Show_Background_Menu, 4-2
SHOW_FAST_FORWARD_BUTTON, 4-6
Show_Menu, 4-2
SHOW_PLAY_BUTTON, 4-6
SHOW_POPUPMENU, 4-6
SHOW_RECORD_BUTTON, 4-6
SHOW_REWIND_BUTTON, 4-6
SHOW_SLIDER, 4-6
SHOW_TIME_INDICATOR, 4-6
SHOW_VOLUME_CONTROL, 4-6
SO, 12-1
sound, 9-1
Statistics, 7-2
Status-Empty, 8-3
Status-Hint, 8-3
Status-Items, 8-3
Status-Message, 8-3
STEREOPHONIC, 4-6
Sub-menu, 8-3
syntax, obsolete, 4-6

T

Terminal, 1-3
Terminate, 4-3
TextControlCurrent, 8-3
TextControlFailValidation, 8-3
TextControlNonCurrent, 8-3
TextControlSelect, 8-3
ToolkitCurrent, 8-3
ToolkitCurrentMnemonic, 8-3
ToolkitDisabled, 8-3
ToolkitDisabledMnemonic, 8-4
ToolkitEnabled, 8-4
ToolkitEnabledMnemonic, 8-4
Translation Builder, 1-3
TranslationHub, 1-3
Trigger Style, 6-3
triggers, obsolete, 5-1
TT, 12-2
Tuxedo, 1-3

U

UN, 12-1
Underline, 8-4
user exits, V2, 11-1
user-defined menu parameters, 12-2

V

V2 triggers, 5-1
V2 user exits, 11-1
V2.3-style LOVs, 10-1
VBX, 9-1
VBX.FIRE_EVENT, 4-4

VBX.GET_PROPERTY, 4-4
VBX.GET_VALUE_PROPERTY, 4-4
VBX.INVOKE_METHOD, 4-4
VBX.SET_PROPERTY, 4-4
VBX.SET_VALUE_PROPERTY, 4-4

W

Web-based architecture, 15-2
Web-based deployment, 7-1
web.show_document, 14-4
WHEN-CLEAR-BLOCK, 5-2
WHEN-CREATE-RECORD, 5-2
WHEN-DATABASE-RECORD, 5-2
WHEN-NEW-FORM-INSTANCE, 5-2
WHEN-NEW-RECORD-INSTANCE, 5-2
WHEN-REMOVE-RECORD, 5-2
Where_Display, 4-3
White on Black, 6-3
WindowTitleCurrent, 8-4
WRITE_SOUND_FILE, 4-4