

# ADPATCH BASICS

Sean Tuck 3/2000

Following is a document that explains the basics of adpatch. This does not cover all aspects or functions of adpatch, but should cover the primary functions that most people need to understand, and can be covered in a 2 hour class. For complete information regarding adpatch, reference an *Installation Manual* for the appropriate release of Applications. (For R11i, reference the *Maintaining Oracle Applications* manual)

This document was originally created while R10 and R11 were available. However, the information in this document is still accurate for R11i. There have been some changes in 11i patching, and I have tried to highlight the important changes in this document.

## Table Of Contents

- I. Types of Patches
- II. Basic Explanation of R11 Architecture
- III. Anatomy of a Patch
  - A. R10 Server Patch
  - B. R10 NCA Patch
  - C. R11 Patch
  - D. Example of an Unbundled Patch
- IV. Installing a Patch
  - A. R10 Server Patch
  - B. R11 Patch
- V. Patch Failure or Successful Completion
- VI. After Applying a Patch
- VII. Additional Patch Information
- VIII. Explanation of Driver Files
  - A. R10 Server patch.drv
  - B. R10 NCA patch.drv
  - C. R10 Database driver, dXXXXXXX.drv
  - D. R11 Copy driver, cXXXXXXX.drv
  - E. R11 Database driver, dXXXXXXX.drv
  - F. R11 Generate Driver, gXXXXXXX.drv
  - G. Other Commands found in driver files
- IX. References
- X. Quiz
  - A. Quiz Questions
  - B. Quiz Answers

## I. Types of Patches

### **Individual Patch, One-off, Standalone**

These are terms to describe an individual patch that is created to fix one particular bug. Currently, most Application products deliver their patches primarily through 'Patch Sets' or 'Mini Packs'. Individual patches are possible, but the rules as to when one is created varies by product.

### **Patch Set, Mini Pack**

'Patch Set' was the original term used in R10. In R11 the same type of patch is now being called a 'Mini Pack'. Both terms mean a large, cumulative patch, for a particular product, that fixes most or all bugs that have been fixed for that release and product. These patches....

- Are named using a letter.
  - ⇒ For example, 16.1.PO.L or 11.0.AR.D.
- Mini Packs and Patch Sets are cumulative.
  - ⇒ For example, 16.1.AP.D, would include fixes in AP.A, AP.B, AP.C, plus bugs fixed between when AP.C was released and when they began building AP.D.
- These patches are created periodically.
  - ⇒ How often they are created varies by product.
- R10 Patch Sets require both a server side and a client side patch.
- R11 Mini Packs require just one patch.
  - ⇒ However, if you have a multi-node install, then the patch must be installed on each machine.
  - ⇒ If different platforms are used, the patch must be ported for each platform.

### **Release Update, Maintenance Pack**

Maintenance Pack is the term that Oracle is using for R11, while Release Update was used for R10.

A 'Maintenance Pack' is a collection of Mini Packs that are bundled together onto a set of CDs that can be ordered and easily installed by the customer.

- With the installation of this type of patch, the 3<sup>rd</sup> digit of Applications Release will change.
  - ⇒ Some examples Maintenance Packs are 11.0.1, 11.0.2, or 11.0.3.
  - ⇒ In R10, some examples of Release Updates would be 10.6.1, or 10.4.2.
- In R11, when applying a Maintenance Pack, you can choose to apply certain product's Mini Packs individually, or you can apply all the product's Mini Packs at once using one set of drivers.
  - ⇒ However, the Application's version will not be updated, for example to 11.0.3, if the Mini Packs are applied individually.
- As with Mini Packs, Maintenance Packs are cumulative.
  - ⇒ So if you apply 11.0.3, you do not need to apply any of the prior Maintenance Packs (11.0.1 and 11.0.2).
- The primary purpose of Maintenance Packs is to fix bugs that have been identified. However, on occasion new functionality may be introduced.

## **NLS Patches**

If your install has multiple languages, additional patches may need to be installed for each language. The American patch needs to be applied first, then an NLS version of the same patch needs to be applied for each language. When applying the NLS patch, be sure to set the NLS\_LANG variable to AMERICAN\_AMERICA.<character set>, substituting the appropriate character set for your language. For more information on other variables and steps, consult the NLS Installation manual.

Not ALL patches will require an NLS version of a patch. For example, if the patch is only creating a new package, an NLS patch would not be required. However a patch that is replacing a form or report, or all large patches, such as Mini Packs or Maintenance Packs will require an NLS patch. Consult Oracle Support to confirm if an NLS patch is needed when applying a patch.

Also, 11i is introducing a feature where adpatch will alert the customer if an NLS patch may be needed.

## **Minor Release**

A couple examples of a minor release are 11i or 10.7. A minor release is not a patch, it is installed using AutoInstall.

- The primary purpose of a Minor release is to introduce new functionality.
- Supported versions of Applications are based off of a Minor Release.

## II. Basic Explanation of R11 Architecture

In order to understand how R11 patches are applied, you need to understand the R11 architecture. For a complete picture, it is recommended that you read the manual Oracle Applications Release 11 for UNIX Concepts (Part number A63418), or the NT version (A63472). The R11i version is A82932-01.

Documented below is a very simple explanation of R11 Architecture. Note, this is not a complete description, and to get a full understanding you should read the Concepts manual.

R10.7 NCA has three tiers, which contain the following files (among other files):

### Server Tier

- Concurrent processes (Ex: reports, executables)
- Character Forms (.inp and .frm, which do not exist in R11)
- Files that maintain the database (Ex: files in patchsc/107/sql or patchsc/107/sql)
- If Self Service Web Apps are installed, html and java files

### NCA or Forms Server Tier

- GUI Forms (.fmb and .fmx)

### Desktop Client Tier

- Contains appletviewer or JInitiator to log on to Applications from the client

In R11 (or 11i), Oracle has allowed customer to break these components into the following tiers and servers.

### Database Tier

#### Concurrent Processing Server

- Concurrent processes (Ex: reports, executables)

#### Administration Server

- Files that maintain the database (Ex: patch/110/sql)

### Application Tier

#### Forms Server

- GUI Forms (.fmb and .fmx)

#### Web Server

- html and java files

### Desktop Client Tier

- Contains appletviewer or JInitiator to log on to Applications from the client

Do not interpret the term 'Server' to mean a machine, such as Solaris or NT. These servers can all be on the same machine (single node install), or divided into numerous machines (multi-node install). For example, the customer could have the Concurrent Processing Server and the Administration Server on a Solaris, and the Forms Server and the Web Server on an NT. This is the commonly used 'two-tier' type of

install. Many other combinations can be used, including multiple Forms Servers for Load Balancing or Fault Tolerance.

When a patch is applied in R11, the patch process asks 4 questions to determine which of these servers exist on the machine the patch is being applied to. These questions are listed below. I have included which server these questions correspond to in ( ) under the question.

Do you currently have files used for installing or upgrading the database installed in this \$APPL\_TOP [Yes] ?  
(Administration Server)

Do you currently have Java and HTML files for Self-Service Applications installed in this \$APPL\_TOP [Yes] ?  
(Web Server)

Do you currently have Oracle Applications forms files installed in this \$APPL\_TOP [Yes] ?  
(Forms Server)

Do you currently have concurrent program files installed in this \$APPL\_TOP [Yes] ?  
(Concurrent Processing Server)

After these questions are answered, adpatch will know which files it needs to copy and the jobs it will need to perform.

***R11i Note:***

***In R11i, the install process stores the answer to these 4 questions in the file \$APPL\_TOP/admin/adconfig.txt. Then when adpatch is run, it reads this file and automatically answers the 4 questions for you. This makes adpatch easier, and helps prevent user error.***

Example:

You have two machines.

Solaris Machine A, has the Concurrent Manager and Administration servers

NT Machine B, has the Forms and Web servers

The patch, ported for Solaris, would need to be applied to Machine A. When the 4 questions are asked, you would answer 'Yes' for questions 1 and 4, and 'No' for questions 2 and 3.

The same patch, ported for NT, would then be applied to Machine B, and the questions answered in the reverse. 'Yes' to 2 & 3. 'No' to 1 & 4.

### III. Anatomy of a Patch

#### A. Unbundling a Patch

The method of unbundling a patch will differ depending on the Applications release and the type of patch.

- R10 patches use the 'tar' function, except for VMS, NLS, and NCA patches which use 'zip'.
- All R11 patches use the zip function instead of 'tar'. Many platforms may not have zip by default, but it is available via many sources including the platform's interoperability CD, or the web site <http://www.cdrom.com/pub/infozip/>.

Examples of unbundling a patch:

##### R10

```
Uncompress 714657.1070t.Z  
tar -xvf 714657.1070t
```

##### R11

```
unzip p834212_1100.zip
```

#### B. R10 Server Patch

When a patch is unbundled it creates a directory that is the same number as the bug. For example, 714657.1070t will create the directory 714567. The 714567 directory will contain the following:

readme.txt	This is a text file that contains a brief explanation of what the patch fixes, and instructions on how to apply the patch. The readme may also contain manual steps to execute after applying the patch, or an explanation of a new feature introduced by the patch. <b>It is very important that the instructions in the readme are followed closely. Not following the readme may cause the patch to fail, or not correct the problem it was intended to fix.</b>
patch.drv	First driver file executed. A driver provides adpatch with the instructions on the jobs it needs to perform. This driver copies files from patch to directories, generates character forms, generates reports, and relinks executables.
dXXXXXXX.drv	'D' or 'Database' driver, which is the second driver file executed. A patch, Ex. d714567.drv, will contain a database driver only if the patch contains files that will update the database, such as files in the patchsc/107/sql directory. Some examples, are scripts that... <ul style="list-style-type: none"><li>• Create packages</li><li>• Create new error messages</li><li>• Add a new table or view to the database</li><li>• Add a new column to a table</li><li>• Add new seed data to a table</li></ul>
Product directories	Each patch will contain at least one product directory (ap, gl, inv, etc.) that will contain sub-directories and the files that are included in the patch. The product directories will mirror a \$XX_TOP structure. Meaning the files will be in the same directories and sub-directories that you will find in your product directories.

### C. R10 NCA Patch

A R10 NCA patch's structure is the same as a R10 Server patch, but it will not contain a database driver, since the database is updated on the server, not the NCA tier.

In addition, many NCA patches will also include a server side 'coreq' patch which must be applied with the NCA patch.

### D. R11 Patch

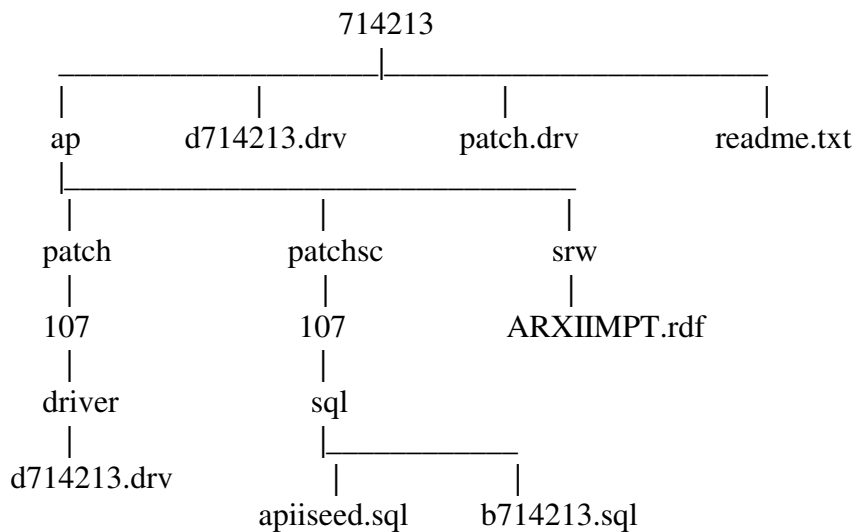
The R11 patch's structure is the same as R10, except for the drivers. R11 will contain the following drivers, if applicable.

cXXXXXX.drv	The copy driver. Similar to patch.drv except it does not generate any files.
dXXXXXX.drv	Database driver. Functions the same as the R10 database driver.
gXXXXXXdrv	Generate driver. Generates forms, reports, libraries, and message files.

Every patch will have a 'c' driver. The other drivers that are included with a patch depend on which files the patch contains. If the patch does not contain any files that update the database, then there is no need for a 'd' driver. Or if the patch does not contain any generated files, such as a form, library, or report, then there is no need for a 'g' driver.

### E. Example of an Unbundled Patch

Following is an example of a small unbundled patch. After you unbundle patch 714213.1070t, it creates the directory 714213.





## IV. Installing a Patch

### A. R11 Patch

Following are the basic steps to installing a patch on R11. Refer to the Installation manual for a complete list of steps to installing a patch.

1. Have all Oracle Application users log out and shut down the concurrent managers.
2. Run the environment file found in \$APPL\_TOP to set the environment.
3. Make sure \$ORACLE\_HOME and \$ORACLE\_SID or \$TWO\_TASK are set correctly.
4. unzip the patch.  
Example:  
unzip p743423\_1100.zip
5. Go to the directory created by 'unzip'.
6. **Read the readme and follow the instructions.**
7. The first step will be to run the 'c' driver, c743423.driv. To start the adpatch process, all you have to do is type:  
adpatch <return>
8. After the patch completes successfully, adpatch may need to be run again using a 'd' (database) and/or 'g' (generate) driver that is included with the patch. Consult the readme for specifics.

#### *R11i Note:*

*In R11i AutoPatch has a non-interactive mode. Using this mode users can run all drivers for a given patch in a single non-interactive AutoPatch session by using another new feature, the defaults file. Details on how to use this feature can be found in the R11i 'Maintaining Oracle Applications' document.*

Following are examples of the questions that are asked when running adpatch. In bold are additional comments that I have added to explain the questions. Any values in [ ] are the default, and is what will be used if you just hit return.

#### ADPATCH PROCESS

Your default directory is '/u02/applmgr/PROD'.

Is this the correct APPL\_TOP [Yes] ?

**This is just confirming your APPL\_TOP. Some customers may have multiple APPL\_TOPs, and this helps prevent someone from accidentally applying a patch to the wrong APPL\_TOP.**

AutoPatch records your AutoPatch session in a text file you specify. Enter your AutoPatch log file name or press [Return] to accept the default file name shown in brackets.

Filename [adpatch.log]

**adpatch.log is the default. However, it is recommended that a log name that is specific to the patch you are applying is used ( Examples, are 714567.log, c714567.log 714567\_PROD.log). This way you can easily find and review the log file for this patch. Otherwise, the log file will be appended to adpatch.log, with all other patch log files.**

You can be notified by email if a failure occurs.

Do you wish to activate this feature [Yes] ?

**If you answer Yes, adpatch will then ask the following question:**

You chose to be notified by email when a failure occurs.

Please enter the email id(s) (separated by a space) that notifications should be sent to [applmgr]:

**You can enter one or multiple email addresses for adpatch to send a message to if it fails. The email sent will contain the last lines of the log file, which should contain the error message.**

Please enter the batchsize [1000] :

**The patch may contain scripts that update data in batches. This prompt allows you to specify how many rows will be updated at a time. It is recommended that you accept the default unless you know your system well.**

NOTE: If you do not currently have certain types of files installed in this APPL\_TOP, you may not be able to perform certain tasks.

**The following examples provided by adpatch are there to help explain the new R11 architecture discussed previously.**

Example 1: If you don't have files used for installing or upgrading the database installed in this area, you cannot install or upgrade the database from this \$APPL\_TOP.

Example 2: If you don't have forms files installed in this area, you cannot generate them or run them from this \$APPL\_TOP.

Example 3: If you don't have concurrent program files installed in this area, you cannot relink concurrent programs or generate reports from this \$APPL\_TOP.

**The next 4 questions help determine which server(s) are on the system you are applying the patch to. If this is a 'Single node' install, meaning ALL servers are on the same machine, then you would answer YES to each of these questions.**

*R11i Note:*

*As note previously, in R11i, AutoPatch/adadmin no longer ask these tier questions. AutoInstall records them in the configuration file \$APPL\_TOP/admin/adconfig.txt. Then adpatch reads and defaults the answers from the configuration file. Adpatch will also default in the 'APPL\_TOP Name' which is also a new feature in R11i.*

Do you currently have files used for installing or upgrading the database installed in this \$APPL\_TOP [Yes] ?

Do you currently have Java and HTML files for Self-Service Applications installed in this APPL\_TOP [Yes] ?

Do you currently have Oracle Applications forms files installed in this \$APPL\_TOP [Yes] ?

Do you currently have concurrent program files installed in this \$APPL\_TOP [Yes] ?

You are about to apply a patch to the installation of Oracle Applications in your ORACLE database 'VIS' using ORACLE executables in '/u03/oracle/805'.

Is this the correct database [Yes] ?

**This is just confirming that you are applying the patch to the correct database. Many customers have multiple instances. This helps prevent them from accidentally applying a patch to the wrong instance.**

AutoPatch needs the password for your 'SYSTEM' ORACLE schema in order to determine your installation configuration.

Enter the password for your 'SYSTEM' ORACLE schema:  
Connecting to SYSTEM.....Connected successfully.

The ORACLE username specified below for Application Object Library uniquely identifies your existing product group: APPLSYS

Enter the ORACLE password of Application Object Library [APPS] :

The default directory is [/u02/applmgr/PROD/patch/714567] :

**The directory that defaults, is the directory where adpatch is started from. If you start adpatch from the directory where the patch was unbundled, then all you need to do is press <return>.**

*R11i Note:*

*At this point, in R11i, adpatch will display the readme.txt file. This encourages users to read the readme.txt file before applying the patch.*

Please enter the name of your AutoPatch driver file [patch.drv] :

**patch.drv always defaults. Type in the name of driver file that needs to be used for this session of adpatch. For example, d714567.drv.**

Do you want to continue with AutoPatch [Yes] ?

Enter the number of parallel workers [6] :

**This determines the number of 'adworkers' that will be created to apply the patch. You only get this prompt if the driver contains the line "compatible parallel yes" or "compatible parallel always", or if the driver file contains at least one 'exec' or 'sql' command. The default value is equal to the number of CPUs on the machine, plus 2. It is recommended that the default is used, because this makes the patch run faster. If you choose more than the default it may bog your database server down and cause the patch to run slower.**

***R11i Note:***

***In R11i default logic for the number of workers looks at number of CPUs on machine where RDBMS is located, not the local machine.***

That is the last question. Then adpatch begins to apply the patch.

**B. R10 Patch**

Following are the basic steps to installing a patch on R10. Again, you should refer to the Installation manual for a complete list of steps to installing a patch.

1. Have all Oracle Application users log out and shut down the concurrent managers
2. Run the environment file found in \$APPL\_TOP to set the environment.
3. Make sure \$ORACLE\_HOME, \$ORACLE\_SID, and \$TWO\_TASK are set correctly.
4. Uncompress and untar the patch. Example:  
uncompress 743423.1070t.Z  
tar -xvf 743423.1070t
5. Go to the directory created by 'tar'. There you will see files similar to what was discussed in section II.
6. Read the readme and follow the instructions.
7. Start the adpatch process. All you have to do is type:  
adpatch <return>  
This starts the adpatch process.
8. After the patch is complete, adpatch may need to be run again using a database or 'd' driver that is included with the patch.

Once again, pasted below are examples of the questions that are asked when running adpatch. In bold are additional comments that I have added to explain the questions.

Any values in [ ] are the default, and is what will be used if you just hit return.

**ADPATCH PROCESS**

Your default directory is '/u02/applmgr/FRESH'.

Is this the correct APPL\_TOP [Yes] ?

**This is just confirming your APPL\_TOP. Some customers may have multiple APPL\_TOPS, and this helps prevent someone from accidentally applying a patch to the wrong APPL\_TOP.**

Filename [adpatch.log] :

**adpatch.log is the default. However, it is recommended that a log name that is specific to the patch you are applying is used (Examples are 714567.log, c714567.log 714567\_PROD.log). This way you can easily find and review the log file for this patch. Otherwise, the log file will be appended to adpatch.log, with all the other patch log files.**

Please choose one of the following:

- 1) Standalone installation
- 2) Client installation
- 3) Server installation
- 4) Node installation

The default choice is 1 - standalone installation.

**Choose the default. Details are explained in the R10 Install Manual, page 4-9,10.**

You are about to apply a patch to the installation of Oracle Applications  
in your ORACLE database 'FRESH107'  
using ORACLE executables in '/u01/orasean/7336'.

Is this the correct database [Yes] ?

**This is just confirming that you are applying the patch to the correct database. Many customers have multiple instances. This helps prevent them from accidentally applying a patch to the wrong instance.**

Enter the ORACLE username of Application Object Library [APPLSYS] :

Enter the ORACLE password of Application Object Library [APPS] :

Enter the password for your 'SYSTEM' ORACLE schema:

Enter the directory where your Oracle Applications patch has been unloaded

**The default directory is [/u02/applmgr/FRESH/patch/491035] :**

**The directory that defaults, is the directory where adpatch is started from. If you start adpatch from the directory where the patch was unbundled, then all you need to do is press <return>.**

Please enter the name of your AutoPatch driver file [patch.drv] :

**patch.drv always defaults. When you run adpatch to apply the database driver, you will need to type in the 'd' driver, such as d743342.drv.**

Do you want to continue with AutoPatch [Yes] ?

Enter the number of parallel workers [3] :

**This determines the number of 'adworkers' that will be created to apply the patch. You only get this prompt if the driver contains the line "compatible parallel yes" or "compatible parallel always" The default value is equal to the number of CPUs on the machine, plus 2. It is recommended that the default is used, because this makes the patch run faster. If you choose more than the default it may bog your database server down and cause the patch to run slower.**

That is the last question. Then adpatch begins to apply the patch.

## V. Patch Failure or Successful Completion

### How to tell if a patch completed successfully

If adpatch completes successfully, it will first spool out the contents of the readme.txt file, then display a message like the following.

#### **R10**

AutoPatch is complete

You should check the file

/u02/applmgr/PROD/install/log/D762433.log

for errors.

#### **R11**

AutoPatch is complete.

AutoPatch may have written informational messages to the file

/u02/applmgr/PROD/admin/PROD/log/C716457.lgi

You should check the file

/u02/applmgr/PROD/admin/PROD/log/C716457.log

for errors.

### What to do if a patch fails.

If the patch fails, adpatch may do one of the following. Included under each bullet are the basic steps to restarting adpatch.

- **Errors and returns to the o/s prompt**
  1. Review log file(s) to determine the cause of the error.
  2. Fix the cause of the error.
  3. Restart adpatch.
  4. Answer 'Yes' when adpatch asks if you want to continue the previous session.
  5. Adpatch will skip already completed jobs, and pick up from where it left off.

- **Informs the user that an error occurred and asks if you want to continue**

With this type of error there are two options.

Option 1 - Discontinue Patch Session

1. Review log file(s) to determine the cause of the error.
2. Fix the cause of the error.
3. Restart adpatch.
4. Answer 'Yes' when adpatch asks if you want to continue the previous session.
5. Adpatch will skip already completed jobs, and pick up from where it left off.

Option 2 - Continue Patch Session

1. Make a note of the failure.
2. Choose to ignore the error and continue with adpatch.
3. After adpatch has completed go back and try to identify the problem and complete the job manually.
4. An example of when this error may occur is 22 out of 23 reports generate successfully, but one fails. Continue with the patch, then try to generate the failed report manually after adpatch has completed.

- **Stops running after workers are spawned and informs the user that workers have failed.**

It asks for you to fix the problem and restart the workers. Adpatch does not exit to the o/s, it just suspends working until the problem has been fixed and the workers restarted.

Workers can be monitored and maintained using 'adctrl'. If one worker fails the others may continue until it gets to a point that it cannot continue. At that point adpatch will inform you that the errors in the workers must be corrected.

1. Review adworker log file(s) to determine the cause of the error.
2. Fix the cause of the error.
3. In another screen start 'adctrl'  
adctrl <return>
4. Answer the few questions required to start adctrl.
5. Pick option 2 'Tell worker to restart a failed job'
6. Type the worker number(s) you need to restart and press return. This will restart the failed worker for adpatch.

## VI. After Applying a Patch

Oracle recommends that the following is completed after each installation of a patch.

### 1. Review log files for errors.

Details about the different log files that are created by adpatch are detailed in #9 in the next section.

### 2. Review customizations

If you have registered customized files in the file \$APPL\_TOP/admin/applcust.txt, adpatch will list the files documented in applcust.txt that will be replaced during the install of the patch. This file does not prevent the object or the patch from being applied, it just lists the objects that will be replaced.

If you are concerned about customizations, there are a couple options to consider.

- a. unbundle the patch and review the contents of the patch.
- b. run the patch in test mode. See page 18 for details.

### 3. Consider removing backup or obsolete files

AutoPatch performs version checking before copying a file into the necessary directory. If a file in a patch is determined to have a higher version than the existing copy, or it has the same version but different file size, then the file is copied to the appropriate directory, and the existing version is backed up by appending an "O" to its filename. After a patch is applied you may see something like the following:

```
ARXVATR.N.rdf  
ARXVATR.N.rdfO
```

If you are tight on space, you may want to periodically delete these backup files. However, you should first confirm that the new files works as you expected.

### 4. Maintain MultiLingual or Multiple Reporting Currencies Schemas(s)

If either the MRC or MultiLingual functionality is being used, the system needs to be updated after applying a patch. This is done by running the 'Maintain MultiLingual' or 'Maintain Multiple Reporting Currencies' options from adadmin. For more information consult the appropriate Installation Manual.

### 5. Pin SGA packages

The Installation Manual also suggests that if database objects are updated by the patch (a 'D' driver is run), the scripts ADXGNPIN.sql and ADXGNPNS.sql are run again to pin new packages and sequences in your Oracle System Global Area. Details on this function are also documented in the Installation Manual.



## VII. Additional Patch Information

### 1. Is it OK to apply a patch when users are on the system?

As documented in the Installation Manual, Oracle advises that no users are logged on to the system while a patch is being applied.

### 2. How do you know what patches or files have been applied to a system?

The first time you run adpatch, a file named applptch.txt is created. This file is created in \$APPL\_TOP in R10, and in \$APPL\_TOP/admin/<ORACLE\_SID> in R11. This file is appended each time a patch is applied. The file contains the following information:

- a) Bugs not applied and why
- b) Bugs applied. For each bug applied it lists
  1. Actions executed
  2. Actions not executed
  3. From this detail you can see what files were applied to the system

#### *R11i Note:*

*In R11i the file \$APPL\_TOP/admin/applpsum.txt will also be created that will summarize the information in \$APPL\_TOP/admin/<SID>/applptch.txt*

### 3. How do you stop running adpatch while you are still answering the initial questions?

If you have not answered the final question, then you can stop adpatch at any time by typing 'abort'. This will stop the adpatch session. The next time you run adpatch, it will ask you if you want to continue your last session.

### 4. What if adpatch is already running? How can it be stopped?

If you are applying a D driver, and it has spawned workers, you can stop it by using the program 'adctrl'. Adctrl allows you to stop adpatch while it is running. You can also use it to monitor adpatch and restart adworkers after they have failed. For more details consult the Installation Manual. Otherwise, there is not a way to gracefully stop the adpatch process, so the best practice is to let it run to completion.

## 5. Always check for invalids, and recompile if necessary after applying a patch

If a 'd' or database driver is used while applying a patch, the database should be checked for invalids afterwards. Many times patches will create invalid objects, and this could prevent the patch from working or cause other problems.

### Checking for Invalids:

Select owner, object\_name, object\_type from all\_objects where status = 'INVALID';

### Compiling Invalid Objects:

- Use adadmin to run 'Compile Apps Schema'. Good to use if there is a very large number of invalids, because it compiles all products.
- \$AD\_TOP/sql/adcompsec.pls. This is a script that development will sometimes include as one of the jobs run by the 'd' driver. This script allows you to compile invalid objects in a given schema.
  - ⇒ To run the script use the syntax  
SQL> @adcompsec.pls <schema\_username> <schema\_password> <objects starting with>  
Examples:  
SQL> @adcompsec.pls apps apps PA (This will compile all the objects in Apps starting with PA%)  
SQL> @adcompsec.pls ap ap % (This will compile all objects in the AP schema)
- Recompile and recompile.sql. These are custom scripts created by a support analyst at Oracle. The script selects all invalid objects and dynamically creates 'alter' scripts to compile each of these invalid objects, then runs the scripts. For examples of these scripts look in the reference section, on page 30.
- Use alter command in sql:
  - ex: alter package ARP\_STANDARD compile body;
  - alter view TABLE\_V compile;

## 6. What if you answer the '4 questions' incorrectly when installing a R11 patch?

Suppose you only had your Administration and Concurrent Processing server on a box that you were applying a patch to, but you answered YES to all 4 questions. What would happen?

If you answer "Yes" to all of the tier questions, AutoPatch will take you at your word, and will attempt to apply the patch as if all tiers were installed in your current APPL\_TOP. If your current APPL\_TOP is a Forms Server, there is a fair chance that something will fail, though many patches might apply with no errors or warnings. The end result is the APPL\_TOP will have more files than is necessary.

*Again, this possibility is eliminated in R11i, because the patch process reads a file and answers the 4 questions for you.*

## 7. Where are the log files for the patch process stored?

The log files that are generated during the patch process are stored in the following locations:

R10 - \$APPL\_TOP/install/log  
R11 - \$APPL\_TOP/admin/<ORACLE\_SID>/log  
(ex. \$APPL\_TOP/admin/PROD/log)

## 8. What other adpatch options are there?

There are a few different modes in which adpatch can be run.

### Test Mode

Test mode will display all the files adpatch will replace and the actions it will take, but does not actually apply the patch. The syntax for using this mode is:

```
$ adpatch apply=no
```

This is useful if the user wants to know exactly what a patch will do before applying the patch. For example, if the user is concerned about customizations being replaced.

### Pre-AutoInstall Mode

There are times when Oracle may require the install of a patch before running AutoInstall to install or upgrade Oracle Applications. The patch usually updates AutoInstall itself or one of the utilities it calls during the installation. The syntax for this mode is:

```
$ adpatch preinstall=y
```

### Options parameter

The options parameter can be used to enable or disable certain functions during application of the patch. For example, you may want to prevent forms and libraries from being generated during the application of a patch. The syntax for this would be:

```
$ adpatch options=nogenrep,nogenrpll
```

Full details on all of the available options are in the *Installation Manual*, or for R11i the *Maintaining Oracle Applications* manual.

## 9. What other log files are generated when applying a patch?

When a patch is applied, it may generate other log files besides the log name that is entered at the start of adpatch. Following are some log files that may also be updated by the patch.

### R10

- adfrmgen.log (Form generation)
- adlibin.log (libin commands)
- adlibout.log (libout commands)
- admvcode.log (copying files to destination)
- adrelink.log (relinking executables)
- adrepgen.log (Report generation)
- adwork01.log (adwork02.log, adwork03.log, etc. Depending on # workers used)
- .req files (Some processes may generate .req files. There are numerous different commands that generate a .req file. Below is one example. This line was found in a adwork01.log file.

```
/u02/applsean/FRESH/fnd/6.1.1/bin/FNDMDGEN APPS/APPS 0 Y US PO SCRIPT_TO_DB po
patchsc/107/import/po10sc.msg
```

```
Log filename : /u02/applsean/FRESH/install/log/11189.req
```

```
Report filename : /u02/applsean/FRESH/install/out/ANONYMOU.1189
```

### R10 NCA

- adf45gen.log (Forms generation on NCA)
- adlibin.log
- adlibout.log
- admvcode.log
- adrelink.log

### R11

- adlibin.log
- adlibout.log
- admvcode.log (Note: in R11i, this log file does not exist. The same info is available in the default log directory)
- adrelink.log
- adwork01.log
- .req files

Note: You will notice that there is not a separate log file for forms generation in R11, as there is in R10. That is because Forms generation is now done via the adworkers. So all the details of a Form generation or its error will be in an adworkXX.log file.

### Providing log files to Support

If you have a problem applying a patch and need to call support, be prepared to provide the appropriate log files. This will help expedite the resolution of the issue. Since many of the log files are appended by each patch applied, these log files may be very large. If so, you may want to extract the section that shows the error. You should include about a page before and after the error to assure that all important information is included.

Another option is to backup/rename the log file that is needed and reapply the patch to get the error. For example, a patch is failing during relinking.

- a. First backup the current log file  
`mv adrelink.log adrelink.bk`
- b. Rerun the patch so that it will error again.
- c. A new adrelink.log file will be created that only contains info from the current patch.
- d. Rename this new log file, then restore the backup.

### **10. What is the j<patchnumber>.zip file that exists in some 11i patches?**

You may find this type of zip file when applying Java patches in 11i. These zip files are known as a Zipped Resource Unit. You do NOT have to unzip this file or do anything with it, it will be used by the 'c' driver when adpatch is run. Somewhere in the 'c' driver will be a 'jcopy' command. The jcopy command removes the contents of the .zip file and merges them into your apps.zip file, which is located in the \$AU\_TOP/java directory. This jcopy command replaces the old class files with new ones found in the .zip file from the patch. Your apps.zip file in your \$JAVA\_TOP will also be updated by the jcopy command.

## VIII. Explanation of Driver Files

Driver files, or .drv files, are files that adpatch uses to determine what jobs to perform. The following section provides a basic explanation on how to read a driver file and understand the jobs it will perform.

### A. R10 Server, patch.drv

The patch.drv driver is the first driver which must be run in R10. This driver will do the following:

- a) copy the files from the patch to the appropriate directory
- b) relink executables
- c) generate different files, such as Forms and Reports.

The general form of a driver command line is:

<command> <product> <subdirectory> <file> <other arguments...>

Some of the common <command>s that are used are:

Copy	copy file into target sub-directory with version checking
forcecopy	copy file into target sub-directory without version checking
libout	extract object module from library
libin	put object module into library
genform	generate form
genrep	generate report
genrpll	generate Oracle Reports PL/SQL file and copy to \$AU_TOP/plsql
link	relink executable
exec	run executable program
sql	Run SQL script.

Following are some examples of common commands found in a patch.drv and a brief explanation.

**copy ar admin/sql arpker.sql 70.3**

Driver will check the version of arpker.sql on your system ('version checking'). If it is less than 70.3, then it will copy this file to the \$AR\_TOP/admin/sql directory.

**libout ar lib arstax.o**  
**copy ar lib arstax.o 70.143**  
**libin ar lib arstax.o**

Driver will first extract the .o file from your libar.a file in \$AR\_TOP/lib (libout). It then performs a version check. If the current version is less than 70.143 it copies the .o file from the patch to \$AR\_TOP/lib (copy). It then inserts the .o file back into libar.a (libin). Later, when the patch relinks the executables, the libar.a file is used, and the new file will be linked into the executable.

**link            fnd    bin                            aiad**

Relink the executable \$FND\_TOP/bin/aiad.

**link            oe    bin                            OEBSHC**

Relink the executable \$OE\_TOP/bin/OEBSHC

Following is a simple explanation of 'relink':

Executables are numerous C modules bundled together to create the executable. Linking, or `adrelink`, is the process that bundles all the necessary C modules together to create the executable. In order for the new C code, or object files (arstax.o for example) to take affect, they not only have to be inserted in the product's C library (libar.a or libap.a), but the executables that use this C code must be relinked. This command is the same as if you ran the following commands from the o/s.

R10 - `adrelink force=y ranlib=y "fnd aiad"`

R11 - `adrelink force=y "oe oebshc"`

**genform        ar    forms                            ARXSUDCB.inp**

Generate the form \$AR\_TOP/forms/ARXSUDCB. The character mode form file that Oracle sends the customer is the .inp file. An .inp is a text file, it is NOT the file that is executed by the applications. The `genform` command converts the .inp file into a .frm, which is the executable file. The .inp files, since they are text files, are not port specific. The `genform` command converts this text file into a platform specific executable.

**genrep        ar    srw                            ARXVATRN.rdf**

Generate the report \$AR\_TOP/srw/ARXVATRN.rdf.

## B. R10 NCA, patch.drv

The patch.drv driver on the NCA tier does the following functions.

- a) Copies files to the appropriate directory.
- b) Generates the following
  - Forms PL/SQL Library files (.lc)
  - Forms 4.5 libraries, menus, and form executable files (.plx, .mmx, .fmx)
- c) Relinks executables.

### IMPORTANT NOTE:

One difference between NCA and Server tiers is that the Form and Library files on NCA (.pll and .fmb) are NOT copied to the product directories (ex., \$PO\_TOP, \$AP\_TOP). All .fmb and .pll files are stored in \$AU\_TOP under the forms and resource directories.

- a) The forms (.fmb) are stored under \$AU\_TOP/forms/US. They are then generated and the resulting executable (.fmx) is stored under the product directory (ex. \$PO\_TOP/forms, \$AP\_TOP/forms, etc.)
- b) The libraries (.pll) are stored under \$AU\_TOP/resource. They are then generated and the resulting executable (.plx) is also stored under \$AU\_TOP/resource, not under a specific product.

This driver also relinks executables, but this is limited on the NCA tier. The only executables on the NCA tier exist in \$AD\_TOP/bin and \$FND\_TOP/bin, and are used to maintain the tier, such as, adadmin, adpatch, etc.. Patches may also contain '.o' files that are linked into \$FND\_TOP/bin/f45webm. These '.o' files are user exits, or c code that is used within the NCA forms. User exits may be used for tasks requiring more complex code, or for commonly executed tasks via forms, such as converting currency or rounding.

The command structure is the same:

**<command> <product> <subdirectory> <file> <other arguments...>**

Following are examples of commands found in the NCA version of the patch.drv driver.

**copy bom forms/US CSTFQWTD.fmb 60.78 au no mirror**  
**genf45 bom forms/US CSTFQWTD.fmb**

Driver will check the versions of CSTFQWTD.fmb on your system. If it is less than 60.78, then it will copy this file to the \$AU\_TOP/forms/US directory. The genf45 command then generates this form, and the executable file that it generates (.fmx) is stored in \$BOM\_TOP/forms/US.

**libout po lib poxtr.o**  
**copy po lib poxtr.o 80.7**  
**libin po lib poxtr.o**  
**link fnd bin f45webm**

Driver will extract the .o file from your libpo.a file in \$PO\_TOP/lib (libout). It then compares the extracted file's version to the version in the patch. If the current version is less than 80.7 it copies the .o file from the patch to \$AR\_TOP/lib (copy). It then inserts the .o file back into libpo.a (libin) libpo.a is then later used when executing the link command, and linking f45webm. Remember f45webm is a collection of C code (user exits), that is used by different NCA forms.

**copy po resource POXBWVRP.pll 916.23 au no mirror**  
**genfppl po resource POXBWVRP.pll**

Driver will check the versions of POXBWVRP.pll on your system. If it is less than 916.23, then it will copy this file to the \$AU\_TOP/resource directory. The genfppl command then generates this library, and the executable that is generated (.plx) is also stored in \$AU\_TOP/resource.



### C. R10 Database driver, dXXXXXX.drv

This is the driver that runs .sql, .pls, .odf and other files that update the database. As mentioned previously, some common ways the database is updated by the this driver are:

- Create packages
- Create new error messages
- Add a new table or view to the database
- Add a new column to a table
- Add new seed data to a table

The database driver uses the same command structure:

**<command> <product> <subdirectory> <file> <other arguments...>**

However, as you will see, it uses the <other arguments> section much more. There are numerous different arguments that can be used, but following are some more common examples:

- sql** Run the script directly from the worker
- sqlplus** Spawn a new sqlplus session and run the script
- package** Same as sqlplus but performs package version checking

Another argument you may see at the end of the command string is a '**phase**' command. Before performing any actions, adpatch divides all actions contained in the patch driver file into phases based on information specified in the patch driver. Adpatch performs all actions grouped in one phase in parallel before proceeding to the next.

Many 'd' drivers in patches contain phase definitions that determine what order the files in the driver will run. If the driver does not contain any phase definitions, then the files are executed in the order they appear in the driver.

There are currently around 19 different phases that can be defined, but some examples are:

<u>Phase name</u>	<u>Action taken</u>
seq	Create sequences
tab	Create tables and indexes
pls	Create package specifications (specs)
vw	Create Views
plb	Create package bodies

For a detailed list of all phases, the order they are run, and a description of the actions taken in each phase, consult the AutoPatch Utility section in an Installation Manual.

Following are some examples of common commands found in a dXXXXXX.drv driver, and a brief explanation.

**sql ar patchsc/107/sql b512706a.sql !AR\_PERIOD\_TYPES &un\_ar &pw\_ar sql &phase=tab**

Run the sql script \$AR\_TOP/patchsc/107/sql/b512706a.sql using the ar username and password. The !AR\_PERIOD\_TYPES is another parameter expected by the .sql script. As discussed previously the &phase argument specifies what type of action this script is doing, in this case creating a table, and when the script will be executed by the driver.

**sql ar patchsc/107/sql ARTEHPCS.pls none none none package &phase=pls**

Run the script \$AR\_TOP/patchsc/107/sql/ARTEHPCS.pls which creates a package (phase=pls).

**sql ar patchsc/107/sql artcall5.sql none none none sqlplus &phase=tbm+5 &un\_ar &pw\_ar**

Run the script \$AR\_TOP/patchsc/107/sql/artcall5.sql. The phase 'tbm' tells you that it is altering a table.

The +5 is a way of determining the order a script will run within a phase. For example, within a phase, commands would be run in the following order:

&phase=tbm

&phase=tbm+5

&phase=tbm+99

#### **D. R11 Copy driver, cXXXXXX.drv**

The R11 copy driver's functions are the same as the R10 NCA patch.drv, except that there is not a generation step. The generation of files is done by the 'g' driver in R11.

Following are the primary functions of the R11copy driver:

- a) Copies all files to the appropriate directory. Remember, as in R10 NCA...
  - The forms (.fmb) are stored under \$AU\_TOP/forms/US
  - The libraries (.pll) are stored under \$AU\_TOP/resource.
- b) Relinks executables.

The command format is the same as in R10. Following is a couple examples.

**copy            ap       forms/US                    APXIISIM.fmb        110.8**

Driver will check the versions of APXIISIM.fmb on your system. If it is less than 110.8, then it will copy this file to the \$AU\_TOP/forms/US directory.

**copy            ap       patch/110/sql                apaiithb.pls        110.2**

Driver will check the versions of apaiithb.pls on your system. If it is less than 110.2, then it will copy this file to the \$AP\_TOP/patch/110/sql directory.

**copy            fnd       resource                    JE.pll                110.8**

Driver will check the versions of JE.pll on your system. If it is less than 110.8, then it will copy this file to the \$AU\_TOP/resource directory.

#### **jcody**

You might find a jcody command in a **R11i** patch if it is updating Java files. This command is explained in the **Additional Patch Information** section on page 21.

#### **E. R11 Database driver, dXXXXXX.drv**

See R10 Database driver. The function of this driver is the same in R10 and R11.

## **F. R11 Generate Driver gXXXXXX.drv**

In R10 the generation of files was performed by the patch.drv driver. At times reports would fail because they required new database objects which were not installed yet, because patch.drv is run before the database driver which creates the new objects. In R11 this problem is resolved by breaking out the generation tasks into a driver that is run last.

Following are some examples of commands in a 'g' driver. Once again, remember that

- The .fmb files are stored in \$AU\_TOP, and when generated the executable (.fmx) is stored under the product.
- The .pll is stored under \$AU\_TOP, and when generated the executable (.plx) is also stored under \$AU\_TOP.

**genform      ap      forms/US                      APXIISIM.fmb**

Generate the form \$AU\_TOP/forms/US/APXIISIM.fmb and store the executable in \$AP\_TOP/forms/US/APXIISIM.fmx

**genrep        ap      reports                              APXIIADV.rdf**

Generate the report \$AP\_TOP/reports/APXIIADV.rdf, and store the resulting file APXIIADV.rdf in the same directory.

**genfpll        fnd      resource                              JE.pll**

Generate the PL/SQL library \$AU\_TOP/resource/JE.pll, and store the executable JE.plx in \$AU\_TOP/resource.

## **G. Other Commands**

Following are some other commands that may be found in a driver file. Once again, this does not cover all commands that may be found in a driver file, but these are some common ones that may be of interest.

### **compatible release 10.7**

This command checks the release the customer is on to verify they are applying the correct patch. Patches are specific to release versions. That is why when you will see separate ARU checkins by release, and have to specify a release when requesting an ARU.

### **compatible parallel yes**

### **compatible parallel no**

This command determines if parallel mode will be used when applying the patch. Parallel mode works essentially the same was AutoInstall, where it starts up workers and the workers execute certain commands.

### ***R11i Note:***

***Following are new commands that have been added to the R11i release.***

### ***compatible platform***

*Prevents customers from applying patches for an incorrect platform.*

### ***compatible language***

*Prevents customers from applying patches for an incorrect language.*

### ***compatible translation\_required***

*When applying a US patch, informs NLS customers when language patches may be required*

## IX. References

The following are documents that you can refer to for more information.

Part #	Title
A47542	Release 10.7 for UNIX Installation
A52781	Oracle Applications Server Installation Manual for Windows NT
A57983	Release 11 for UNIX Installation
A57978	Release 11 for Windows NT Oracle Applications Installation
A63418	Release 11 for UNIX Concepts
A63472	Release 11 for Windows NT Architecture
A65054	Oracle Applications NLS Installation Guide

### Recompile and Recompile.sql scripts.

Following are examples of the recompile scripts and instructions on implementing

#### recompile

```
sqlplus system/manager @/home/applmgr/recompile.sql  
rm -Rf /tmp/recompile-sql.sql
```

#### recompile.sql

```
set echo off  
set verify off  
set feedback off  
set pagesize 0  
spool /tmp/recompile-sql.sql  
select 'set feedback on' from dual;  
select 'set echo on' from dual;  
select 'set pagesize 10000' from dual;  
select 'alter ||object_type|| ||owner||.||object_name|| compile;'   
  from dba_object where status='INVALID'   
  and object_type != 'PACKAGE BODY'   
union   
select 'alter package ||owner||.||object_name|| compile body;'   
  from dba_objects where status='INVALID'   
  and object_type = 'PACKAGE BODY';  
select 'select owner, object_type, count(*) "Still Invalid"   
  ||   
  'from dba_objects where status="INVALID" group by owner, object_type;' from dual;  
select 'exit;' from dual;  
spool off  
@@/tmp/recompile-sql.sql
```

#### Instructions

1. create recompile and recompile.sql scripts from the above examples.
2. Modify recompile file, putting the correct address on your system where you are storing recompile.sql
3. chmod 777 recompile (or make recompile an executable file)
4. run recompile  
\$ recompile [return]

## X. Quiz

### A. Quiz Questions

1. When applying a R11 patch, adpatch asks questions to determine which servers are on the current system. What are the 4 servers that adpatch asks questions about?
2. What is the term for a R11 install where all of the different type of servers are on the same machine?
3. If you are applying a patch to a machine that has the Concurrent Processing and Administration servers, how would you answer the following questions

Do you currently have files used for installing or upgrading the database nstalled in this \$APPL\_TOP [Yes] ?

Do you currently have Java and HTML files for Self-Service Applications installed in this APPL\_TOP [Yes] ?

Do you currently have Oracle Applications forms files installed in this \$APPL\_TOP [Yes] ?

Do you currently have concurrent program files installed in this \$APPL\_TOP [Yes] ?

4. On R11 what are the 3 different type of drivers that a patch may contain?
5. Which driver on R11 will ALL patches contain?
6. On R10, which driver will ALL patches contain?
7. What is the difference between the c driver in R11 and the patch.drv driver in R10?
8. In what sub-directory do most of the scripts that update the Applications database reside?
9. If the driver performs the commands libout and libin, which other command will also be run?
10. What is the name for c code that is used in forms?
11. In R10 NCA, what is the name of the executable that the user exits are linked into?
12. In R10 NCA, in what directory are ALL Forms executables stored?
13. In which directory are the libraries stored?
14. What are the extensions (.xxx) for a form text file and an form executable?
15. What is the form executable extension on NCA?
16. What does the argument '&phase=tab' determine?
17. What does 'version checking' mean?
18. In R11, why wouldn't a patch contain the d driver?
19. If a driver contains the command 'compatible parallel yes', then what will the driver do?
20. What does adpatch do with the current version of a file, if it is replaced by a patch?
21. In R11 which directory stores the log files for adpatch?
22. In R10, where are the log files?
23. What servers are on the R11 Database tier?
24. What servers are on the R11 Application tier?

## **B. Quiz Answers**

1. Concurrent Processing Server, Administration Server, Forms Server, Web Server
2. Single Node
3. Yes, No, No, Yes
4. c or copy driver, d or database driver, and g or generate driver.
5. The c driver.
6. The patch.driv driver
7. The R11 c driver does no do any generation (Generate forms or reports)
8. The \$XX\_TOP/patchsc/107/sql or \$XX\_TOP/patchsc/110/sql directory.
9. link.
10. User exits
11. f45webm
12. \$AU\_TOP/forms/US
13. \$AU\_TOP/resource
14. .inp is the text file, and .frm is the executable
15. .fmx
16. It determines which order the script in a database driver will be run
17. When adpatch compares the version of a file included in a patch against what is currently on the system. If it is not a higher version, then it will not be applied.
18. If the patch does not contain any files that will update the database, then the database driver does not need to be created or run.
19. It will spawn workers, (ex: adwork01) to run the jobs in the driver.
20. It renames the file and appends it with an 'O'. For example, ARXVATR.N.rdfO
21. \$APPL\_TOP/admin/<ORACLE\_SID>/log
22. \$APPL\_TOP/install/log
23. The Administration server and Concurrent Processing server
24. The Form server and Web server.