GeoFrame System Administration
Exercise Guide

GeoFrame 4
April 3, 2002
Copyright Notice

© April 3, 2002 Schlumberger. All rights reserved.

No part of this manual may be reproduced, stored in a retrieval system, or translated in any form or by any means, electronic or mechanical, including photocopying and recording, without the prior written permission of GeoQuest, 5599 San Felipe, Suite 1700, Houston, TX 77056-2722.

Disclaimer

Use of this product is governed by the License Agreement. Schlumberger makes no warranties, express, implied, or statutory, with respect to the product described herein and disclaims without limitation any warranties of merchantability or fitness for a particular purpose. Schlumberger reserves the right to revise the information in this manual at any time without notice.

Trademark Information

GeoFrame™, StratLog™, WellPix™, WellEdit™, WellSketch™, and CPS-3™ are trademarks of Schlumberger.

All other products and product names are trademarks or registered trademarks of their respective companies or organizations.
Contents

About this Course ........................................... ix

Chapter 1 Online Help
GeoFrame Bookshelf ........................................ 1-2
Appendices .................................................... 1-5
Keywords ....................................................... 1-6

Chapter 2 Preparing the GeoFrame Environment
Exercise: Exercise 1. Hardware/Software Requirements . . . . . . 2-2
Exercise: Exercise 2. Preparing For GeoFrame Installation . . . . . . 2-4

Chapter 3 GeoQuest Installtool & Geonet Installation
Exercise: Exercise 3. Using the Installtool & Installing Geonet ........ 3-2
Exercise: Exercise 4. Configuring the Geonet Users ............... 3-5

Chapter 4 Creating the Oracle Database
Exercise: Exercise 5. Installing Oracle and Creating the Database ........................................ 4-3
Exercise: Exercise 6. Checking the Database ....................... 4-8
Exercise: Exercise 7. Stopping and starting the Database . . . . . . 4-12

Chapter 5 GeoFrame Installation
Exercise: Exercise 8. Preparing to Install GeoFrame ........... 5-2
Exercise: Exercise 9. Installing GeoFrame ....................... 5-8
Exercise: Exercise 10. Checking the GeoFrame Installation . . . 5-12

Chapter 6 GeoFrame Configuration
Exercise: Exercise 11. License Management ................... 6-2
Exercise: Exercise 12. GeoFrame Administration ................. 6-4
Exercise: Exercise 13. SQL*Net Configuration .................. 6-8
Exercise: Exercise 14. Creating Tablespace Pairs ............... 6-10
Exercise: Exercise 15. Creating Projects ....................... 6-12
Chapter 7  Oracle Administration; Performance

Exercise: Exercise 16. Space Management; Adding Datafiles 7-2

Exercise: Exercise 17. Space Management; Removing Projects 7-4

Exercise: Exercise 18. Moving Tablespaces 7-7

Exercise: Exercise 19. Performance; Tuning the SGA 7-9

Chapter 8  Oracle Administration; Backup & Recovery

Exercise: Exercise 20. Cold Backups 8-2

Exercise: Exercise 21. Securing your Controlfiles 8-7

Exercise: Exercise 22. GeoFrame Backups 8-11

Exercise: Exercise 23. Database Security 8-15

Chapter 9  Other GeoFrame Issues

Exercise: Exercise 24. CTREE Server 9-2

Exercise: Exercise 25. Manually Configuring Geonet 9-7

Exercise: Exercise 26. Plotting Configuration 9-10

Chapter 10  Appendix-1

Appendix A; Geonet Installation  Appendix-2

Getting Started  Appendix-3

Installation Phases  Appendix-4

Setup Procedure for New User Accounts  Appendix-7

Setup Procedure for Existing User Accounts  Appendix-9

Appendix B; Oracle Installation  Appendix-11

Preparing to install Run-Time Oracle  Appendix-13

Hardware and Operating System Requirements  Appendix-14

Other pre-installation tasks performed by root  Appendix-24

Installing the Run-Time Oracle  Appendix-29

Post Installation Tasks  Appendix-40

Appendix C; GeoFrame Installation  Appendix-47

Pre-Installation Tasks  Appendix-48

Pre-installation tasks performed by root  Appendix-52

Planning for installation  Appendix-53

Installing GeoFrame  Appendix-56
Index of Keywords

Symbols
  .cshrc 3-1, 3-5
  .gq_kshrc 3-1
  .InstallLog 3-1, 3-2
  _remote_save.csh 8-1, 8-13
  "init.ora" 7-10

A
  ADI_DEBUG 6-1, 6-12

B
  Baseline Account 5-1
  Baseline account 5-5
  BULK_SERVER_ADDRESS 9-1, 9-2

C
  catalog 6-1, 6-4
  checklist
    planning 5-4, Appendix-65
  cleandb 7-1, 7-6
  Cold Backup 8-2
  cold backup 8-1
  connect internal 4-2, 4-12

D
  daemon 6-1, 6-2
  Data Tablespace 7-1, 7-2
  data tablespace 7-7
  Database 4-1, 4-3
  Datafile 4-1
  db_block_buffers 7-1, 7-10
  db_block_size 7-1
  DB_DEF_DISK1_DIR 5-1, 5-2
  disk space
    requirements 5-4, Appendix-65
  DSL 6-1, 6-4
Contents

F
Fast Backup 8-1, 8-12
FLEXLM_BIN_DIR 5-1, 5-2
free_space 7-1, 7-2

G
geoframe 3.0 installation tool Appendix-54
  examine prerequisites phase Appendix-70
  executing procedures Appendix-71
  extracting files Appendix-70
  installation parameters phase Appendix-62
  installation planning checklist 5-4, Appendix-65
  installation steps phase Appendix-59
  license file phase Appendix-58
  mode of installation Appendix-56, Appendix-58
  prerequisite tasks Appendix-56
  ready to execute procedures phase Appendix-70
  ready to extract files phase Appendix-69
  root password phase Appendix-61
  running the install tool Appendix-56
  select locations to install components phase Appendix-59
GeoFrame Archive 8-1
GeoFrame_Runtime.csh 6-1
Geoframe_Runtime.csh 6-3
gf_accounts 5-1, 5-13
GF_DBA 6-1
gf_dba 6-4
gf_setup 8-1, 8-16
gf_users 5-1, 5-13
GN_DIR 3-1, 3-5, 9-1
GN_dir 9-7
gn_examples 3-1, 3-5
gncopy.sh 3-1, 3-5
group 2-1, 2-4, 2-5
gstab 9-1, 9-7

H
hardcopy 9-1, 9-10
hcsetup 9-1, 9-10
hostid 6-1

I
Incremental Backup 8-1
Index Tablespace 7-1
index tablespace 7-7
init.ora 7-1
installation Appendix-47
  planning 5-4, Appendix-65
  requirements 5-4, Appendix-65
installation planning checklist 5-4, Appendix-65
installtool 3-1, 3-2
Instance 4-1, 4-4

K
kernel 2-1, 2-2
keyword/value pair 9-1, 9-11

L
LABEL 9-7
  9-1
LIC_BIN 6-1, 6-3
LIC_DIR 6-1, 6-3
LM_LICENSE_FILE 5-1, 5-2
LOCAL 9-3
  9-1
log_buffer 7-1, 7-10

N
-nobulk 8-1
nobulk 8-13

O
ORACLE_HOME 4-1, 4-3
Oracle_Runtime.csh 4-1, 4-8
ORACLE_SID 4-1, 4-3
owner 2-1, 2-4, 2-5

P
packages 2-1, 2-2
patches 2-1, 2-2
PersonalHardcopy.hcd 9-1, 9-10
planning
  checklist 5-4, Appendix-65
pre-installation tasks
  required operating system patches Appendix-48
  set up kernel parameters Appendix-52
  set up UNIX group accounts Appendix-52
  set up UNIX user accounts Appendix-52
Printer 9-14
printer 9-1
proj_admin 9-1, 9-4
proj_changepasswd 8-1
proj_delete 7-1, 7-5
proj_dsl_listing.csh 7-1, 7-5
PROJECT
  9-1
project 9-3
project table 9-1, 9-3

R
remote_gen_save.csh 8-1, 8-13
required operating system patches Appendix-48
requirements
disk space 5-4, Appendix-65

S
schema 5-1, 5-8
semaphores 2-1
semmns 2-1, 2-3
semmsl 2-1, 2-3
Server 9-2
server 9-1
SET 9-7
  9-1
SGA 4-1, 4-7, 4-9
Shared 6-1, 6-13
Shared (Data Type) 6-1
shared memory 2-1, 2-2
shared_pool_size 7-10
shmmax 2-1, 2-3
shmseg 2-1, 2-3
shutdown abort 8-1, 8-4
SITE
  9-1
site 9-6
SiteHardcopy.hcd 9-1, 9-10
space_check 7-1, 7-4
Spooler 9-11
spooler 9-1
SQL*Net 4-1, 4-4
Standalone 6-1, 6-12
startup mount 7-1, 7-8
Sub 6-1, 6-13  
SYS_PWD 5-1, 5-2  
SYSTEM_ACCNTPASSWD 5-1, 5-2

T
  Tablespace 4-1, 4-6  
  Tablespaces 4-4  
  TIMED_STATISTICS 7-1, 7-9  
  TNS_ADMIN 4-1, 4-3  
  tnslsnr 4-2, 4-8  
  tnsnames.ora 6-1, 6-8  
  tnsping 4-2, 4-10  
  TWO_TASK 4-1, 4-3  
  TYPE 9-7  
    9-1

U
  umask 2-1, 2-4  
  utlbstat 7-1, 7-9  
  utlestat 7-1, 7-9

V
  v$controlfile 8-1, 8-2  
  v$datafile 8-1, 8-2  
  v$logfile 8-1, 8-2
Overview

This course will introduce all of the concepts and skills that must be mastered in order to successfully prepare an environment for the installation of GeoFrame 4, and of course, to install GeoFrame and to configure it to work as required. This will include application servers such as Oracle and FLEXlm, as well as other third party products.

Aside from the obvious objective of installing GeoFrame, the course will also serve to introduce basic Oracle architecture and the most common Oracle Database Administration (DBA) tasks that are required to keep the database running, and to maintain adequate recovery resources.

This course assumes a minimal amount of Oracle experience, and although students should be fairly comfortable with navigating around a Unix environment and editing files, the pace of the class should allow for any level of experience in these two areas. Due to the nature of the course and the significance of the level (root) at which some of the exercises are performed, make sure any questions you may have are brought to the attention of the instructor. Understanding the importance of the root user and the items performed as root user during the course are part of the course objectives as well.

With that said, the objective of the course is not to produce Unix System Administrators or Oracle DBAs. It is, however, to provide a better understanding of these concepts which will allow you to return to your jobsite and create a working GeoFrame installation, and maintain it to the level that is required by your client. If your duties will include those of a System Administrator or Oracle DBA, additional references can be recommended by the instructor.
How to Use the Exercise Guide

This document is provided as a set of practical exercises that are intended to guide you through each task that usually needs to be performed as part of creating a working GeoFrame environment.

The entire course uses various parameters that are intended for the environment in this classroom, and may not be the best setting for the machines or setup you have at your jobsite. Usually the recommended setting will be provided, or a reference to a location where the recommended settings can be found.

Throughout the exercises, any lines depicting an example of the readout from the screen, for instance, a command or expected output will be in a bold font as follows:

I am output from the computer screen

% I am a command to enter

Additionally, whenever you are meant to provide a command, the following prompts are used for normal system prompts or whenever logged on as an operator:

%
>
prompt>

And for commands meant to be issued as the root user:

#

Finally, when using SQL*Plus or the Server Manager Line Mode Oracle utilities, the following prompts are used:

SQL>
SVRMGR>

Aside from those, any remaining words in **bold** are:

- Keywords
- Values
- Items > to be clicked > with the mouse

Of which, the intention of the bold font will be obvious.
Exercises

The exercises may list specific steps for the user to perform, and specific data to enter, or they may assume a certain amount of experience or knowledge at that point in the course. For instance, the following sequence may appear earlier in the course to stop the database:

1. Open a terminal window and logon as oraown.
2. Source the Oracle_Runtime.csh file.

   % source Oracle_Runtime.csh

3. Start the Server Manager by typing svrmgrl at the prompt.
4. Connect to the database as Oracle user sys.

   SVRMGR> connect internal

5. Shutdown the database by issuing the shutdown command at the Server Manager prompt:

   SVRMGRL> shutdown

Whereas, later in the course the instructions to stop the database might look more like this:

1. Shutdown the database.

If the intention of the step in an exercise is not apparent, it is helpful to review the steps in the exercise before (or after) that step. Certainly if there is any confusion as to what should be performed during the step, ask the instructor.

Discussion

In addition to the numbered steps in each exercise there may be topics that are meant to open for discussion, denoted in the exercises by:

DISCUSSION:

This is intended to be an opportunity to expand on the current topic in the exercise by discussing different scenarios that can exist or reasons for choosing the parameters that were chosen. It is also a time for people to share experiences they may have on the subject, or perhaps, to make an even better recommendation.
Course Layout

The GeoFrame System Administration Course is intended to be a 5 day course.

The course begins with not much more than a computer system with the root user account, and progresses from there. The general sequence of events is illustrated by the following diagram, with the daily agenda as follows:

Day 1
- Preparing the systems kernel parameters and checking for other Operating system requirements.
- Creating the necessary Unix accounts and allocating resources
- Introduction to the Installtool and Installing Geonet
- Installation and Creation of the Oracle database

Day 2
- Preparation for GeoFrame installation
- Installing GeoFrame
- Post-installation tasks
- GeoFrame Administration tasks
  — DSL assignment
  — Tablespace management
  — Match Rule editor
  — etc.
Day 3

- Configuring the FLEXlm License Manager
- CTREE server
- Final GeoFrame/Oracle configuration
  - SQL*Net
  - Additional Tablespaces
- Oracle DBA; Space Management

Day 4

- Space Management Continued
- Oracle DBA; Performance issues
  - Tablespace location; moving tablespaces
  - Fragmentation
  - Tuning the SGA
- Oracle DBA; Backup & Recovery
  - Cold Backup
  - Securing control Files
  - Other backups; GeoFrame/Oracle

Day 5

- Database Security
- Geonet Configuration
- Plotting issues

Occasionally, this course may be offered as a three-day course due to a client request or other logistical reasons. The overall flow of the course will remain however, some of the steps may be condensed or omitted in order to fit the course into 3 days, for instance, preparing the Unix system with regards to kernel parameters and accounts.
Chapter 1
Online Help

Overview

This chapter will introduce the resources that are available to you to obtain online help. Each chapter will specifically outline the paths to the online help documents mentioned in this chapter, as well as provide a list of keywords that you can search in the help documents.
GeoFrame Bookshelf

All of the reference material for this course can be found in the GeoFrame Bookshelf set of documents, which come with the GeoFrame software. The GeoFrame Bookshelf will also serve as your main source of reference in the field during practice.

The GeoFrame Bookshelf contains all of the information you should need to adequately install, configure, and maintain your GeoFrame environment, including Oracle administration, FLEXlm configuration, CTREE Server, and other applications. It also contains help on many of the GeoFrame administration utilities that will be covered in this course.

The GeoFrame Bookshelf is a set of FrameViewer documents that are shipped with GeoFrame. They are easily accessed from the Geonet window by clicking on the version of GeoFrame in the Geonet window and dragging the cursor down to GeoFrame Bookshelf:

That will launch FrameViewer, which you have already installed on your machine, or the version that was installed at the time of Geonet installation from the Utilities CD.

1. FrameViewer is a trademark and product of Adobe Systems Incorporated.
Upon launching, you will see the GeoFrame 4 Help window open, which will allow you to access the Help document for all of the different applications in the GeoFrame suite.

Most of the help documents that will be referenced during the System Administration course are found by selecting the **GeoFrame System** button at the lower right hand corner of the GeoFrame 4 Help window:
Additional help documents that are used during this course, and that are helpful in practice are the Project Manager help document, and some of the Utility documents. These may be viewed by selecting **Project** and **Utility** respectively from the window above.
Appendices

Given the nature of the course and the fact that all that is resident on the computer at the beginning of the course is a root user account and disk space, the GeoFrame Bookshelf does not yet exist. Therefore portions of the Installation guides taken from the GeoFrame Bookshelf have been appended to the end of this document.

Appendix A is a portion of the Geonet installation guide. This Appendix contains all of the reference information that is required to install Geonet and configure the user accounts. Additional help with Geonet, including the Geonet user’s guide can be obtained by the online documents provided with Geonet by selecting Util > Doc Browser from the Geonet window.

Appendix B is a portion of the Oracle Run Time Installation guide which is taken from the GeoFrame Bookshelf. It is only a portion of that document, but that will provide the assistance you will need to successfully install Oracle and create a database. Additional help with Oracle and Database administration will be available from The GeoFrame Bookshelf after GeoFrame Installation.

Likewise, Appendix C is a portion of the GeoFrame Installation Guide that will allow you to successfully install GeoFrame.

Upon the successful installation of GeoFrame, the GeoFrame Bookshelf will be available to you from the Geonet window.
Keywords

In addition to main objectives, each chapter in this manual contains a list of selected keywords, pulled from the exercises and help documents.

The keywords may be important parameters and are essential for the exercises in the chapter, or they may simply be words that are crucial to understanding a concept. It will be beneficial to understand the meaning or the significance of each of the keywords.

The keywords that are taken from the online help documents can be searched for within that document using FrameViewer. After opening the document click Edit > Find... to open the FrameViewer – Find dialog box:

Simply type the keyword in the box and click Find in the lower left corner of the window. The next occurrence of the word can be found by repeatedly clicking Find.
Overview

In this chapter you will learn how to prepare your system for a complete GeoFrame environment. This will include all of the requirements for an Oracle database as well as GeoFrame 4.

The chapter is divided into two parts: the systems’ performance and kernel parameters, and the actual configuration of the groups and accounts that will own and operate the software.

Keywords

The following keywords are present in this chapter. After completing the exercises, you should have a better understanding of what each of the keywords mean and their significance:

- patches
- packages
- shared memory
- semaphores
- kernel
- shmmax
- shmseg
- semmsl
- semmns
- umask
- group
- owner
Exercise 1. Hardware/Software Requirements

Prior to installing software on any machine, you need to find out if the software will fit and if the machine will be able to run the software at an acceptable speed.

1. Logon to your machine as the root user.

2. Do you have enough disk space on your machine for the applications you intend to install? How much space is required for the software on a machine that will be a dedicated Oracle Server? How much for a machine that will host Oracle and GeoFrame?

   # df -k /<your installation location>

   # /usr/sbin/swap -s

3. Does your machine meet the memory requirements for the applications you intend to install on it?

   # prtconf
   The amount of memory on your machine will be shown near the top of everything that just went by.

4. The following patches are required by GeoFrame: 108652-xx, 108940-xx, 109543-xx, 109544-xx. Are they present?

   # showrev -p | grep 108652

5. How about operating system packages? Are these present?

   # pkginfo -i SUNWarc SUNWbtool SUNWhea SUNWlibm SUNWlibs
   SUNWmfrun SUNWsprot SUNWtoo
   The Oracle Server will have some requirements with regards to memory resources; specifically shared memory resources. You need to check and possibly configure your UNIX Kernel parameters to accommodate these demands.
6 What's the status of the Interprocess Communication Facilities currently? How much memory is being used? How many semaphores are in use?

   # ipcs -b

7 Configure the Kernel parameters according to the teacher's instructions (which are derived from the settings in a default init.ora file.)

   # vi /etc/system

8 If the instructor does not already have a file prepared that you can append to the /etc/system file, edit the /etc/system file accordingly:

   set shmsys:shminfo_shmmax=128000000
   set shmsys:shminfo_shmmin=1
   set shmsys:shminfo_shmmni=100
   set shmsys:shminfo_shmseg=20
   set semsys:seminfo_semmns=200
   set semsys:seminfo_semmni=100
   set semsys:seminfo_semmss1=200

9 Since you have altered your machine's Kernel parameters, reboot your machine.

   # init 6(Sun)
Exercise 2. Preparing For GeoFrame Installation

This exercise will cover the final steps of setting up the UNIX environment in preparation for the installation of GeoFrame and all of it’s server applications (specifically Oracle.) This involves creating the appropriate UNIX groups and users.

1 What is the significance of the “umask” setting?

   # man umask

   # umask

2 On your Oracle Server, which UNIX user will own and administer the Oracle installation? Who will own the GeoFrame installation on the machine where it resides? What will be the group name?

   • • • • •

   NOTE: For this course please use “oraown” for the oracle owner, and “gfown” for the GeoFrame owner. A separate owner can own Geonet; however, you will really need to watch permissions later.

   The group “dba” is created for the Oracle owner. A group commonly called “gqs” is used for GeoQuest products (owners and operators). This allows all GeoQuest products to easily register themselves with Geonet and allow all members of the group to utilize the products accordingly.

3 Who will be the GeoFrame operators? Do they already have accounts? Write down the list of users here.

    ___________________________ ___________________________ ___________________________
    ___________________________ ___________________________ ___________________________
    ___________________________ ___________________________ ___________________________
    ___________________________ ___________________________ ___________________________
    ___________________________ ___________________________ ___________________________
Create the “dba” group on the Oracle server machine if it doesn’t already exist. Create the “gqs” group on the GeoFrame host. (the same machine for this course.) You can use UNIX commands for this, or the Administration tool, for instance; /bin/admintool for Sun.

4 Logon to your machine as the root user.

5 If you intend to use the admintool, start it by typing /bin/admintool at the prompt.

6 Click on browse > groups to view the groups that are created on your machine.

7 Click Edit > Add to create a group. Create the dba and gqs groups. You can let the machine specify the group ID for you. It should automatically choose the next available one on your machine. You do not need to specify members here.

![Admintool: Add Group](image)

Create the Oracle owner, GeoFrame Owner, and operators if they aren’t existing users. Make sure they belong to their appropriate groups. If you are going to use an existing UNIX account, you must add the user to the gqs group. When specifying home directories, start at the directory specified by the instructor.

8 Click browse > users to view the unix accounts that exist on your machine.

9 Click Edit > Add to create new users. Create the Oracle owner "oraown" and the GeoFrame/Geonet owner “gfown.”
• The machine will choose the next available user ID for you.

• In the primary group section for each user, you can type the appropriate group ID, or spell out the group name: “dba” for oraown, and “gqs” for gfown.

• Select C shell for all users.

• Set a **normal password** for each user. For the scope of this course, please use the username. Normally, in practice, you would select a more secure password.
• Enter the full path for the **home directories** for each user. For the software owners, this can be used as the installation location as well. For all owners, this is where many logs will be written and login files are kept. Start the path at the top level specified by the instructor. (The top level should be the same location you checked for space in the previous exercise.

**NOTE:** It is important that each user account has and owns their home directory, particularly the software owners that will be installing software in these directories.

10 Exit the admintool.
Online Help

The keywords can be found in the Installation guides for Geonet, Oracle Run time, and GeoFrame located in the GeoFrame Bookshelf. Since the GeoFrame Bookshelf is not installed yet these are printed out and are Appendix A, B, and C of this manual respectively. These documents can also be referenced for assistance with the concepts presented in this chapter.

Summary

At this point you should have completely configured your system in preparation for the Oracle database and GeoFrame 4, as well as developed a more thorough understanding of the impact the GeoFrame environment will have on your system’s resources. Also, you know which Unix accounts will be used to install Oracle and the GeoQuest products. This will allow you to maintain the security of the environment in the future.

In the next chapter you will learn to use the GeoQuest Installtool and install Geonet.
Chapter 3
GeoQuest Installtool & Geonet Installation

Overview

This chapter will introduce the GeoQuest Installtool. As a practical demonstration of the Installtool, the first exercise involves installing Geonet. Additionally, this chapter will provide an understanding of the steps that are necessary to allow the system administrator to easily configure new Unix users to operate Geonet.

The first topic of this chapter, Using the Installtool and Installing Geonet, covers providing a new Unix account with .login files from templates that are provided with the Utility CD. The second topic/exercise, Configuring the Geonet Users, will illustrate the required steps to configure an existing account as well.

Keywords

The following keywords are significant in this chapter. After completing the exercises you should have a better awareness of what the keywords mean and their significance:

installtool
GN_DIR
gn_examples
gncopy.sh
.cshrc
.gq_kshrc
.InstallLog
Exercise 3. Using the Installtool & Installing Geonet

The Unix system is now ready for complete GeoFrame installation. In addition to the media containing the software (usually a CD-ROM) you should have received a license file. This may have been on a floppy, CD-ROM, or even just as an e-mail.

This exercise will serve as an introduction to using the installtool, allowing you to learn about the phases of the installation process. At the end of this exercise you should know where to look in the log file (.InstallLog) that is produced for errors or installation parameters that were used.

1. Where is your license file? Load it onto your License Server machine. Who should own the license? What should the permissions on the license file be set to?

2. Locate your license file. You may move it to any location you like. Rename it to license.dat. The license file will only need to be read, so ownership is not such an issue.

   
   NOTE: If you have experience with license management, you may prefer some default location. Otherwise, for the scope of this course, keep the location simple, as we may be using the full path to the license repeatedly.

3. Logon as your Geonet owner and check your umask. Make sure it is set to 022.

   % umask

   
   NOTE: A umask setting of 22 will not give write privileges to the members of the same group. Any product wishing to register itself with Geonet will need to have write permission to certain Geonet directories and files. If you have a different owner for Geonet, GeoFrame won’t register correctly with Geonet automatically. It may have to be run manually.
4 Insert your Utilities CDROM and run the installtool. For the course, the media may be copied locally onto each machine. Run the installtool from that location, instead of the CD-ROM location.

% cd /media/utility_20

% ./installtool

5 Note the message window at the bottom of the screen. Watch for any errors.

6 Select only the Solaris components to install at this time. Use the arrows to view the sub-components. The IRIX components are not needed for this course, and Oracle will be installed by the Oracle software owner.
7 Confirm the installation location.

8 Select to register the Utilities documents with the DocBrowser Menu.

9 Install FrameViewer by choosing not to use an existing one. You may provide the correct license location at this time, or later. It will not matter here.

10 Proceed through the rest of the installation steps.

11 What is the name of the log file that was created? Take a look at the log file and see if there were any errors during installation. Study the logfile, noting the Installation parameters and execution phases. (The logfile will be hidden in the home directory of the Unix user that ran the installtool.)

The installation phases that the installtool proceeded through will be the same for every product you install. Some of the other products will have many more parameters that must be provided, and many more installation procedures that must be performed.

Remember that until the installation procedures have been executed, you can always move backwards if you want to change the parameters.
Exercise 4. Configuring the Geonet Users

It is not necessary to configure Geonet in order for Geonet to work. More correctly, it is necessary to configure the login files for any user who wishes to use Geonet. At a minimum, this will be all of the GeoFrame operators.

1. Login as `gfown` and open a new terminal window.

2. Locate the "`gn_examples`" directory. It should be located in `$HOME/gn502_00`. View the contents of the directory.

3. View the README file.

4. Locate the login files in the same directory. They are hidden just as the login files in your HOME directory are for each Unix user.

Configuring users to be able to use Geonet primarily involves setting an environment variable that points to the Geonet Baseline. For new user accounts, the `gncopy.sh` script can be used as instructed.

In an exclusively SUN environment, you can prepare the login files prior to running the script, such that anytime a new GeoFrame operator is to be created, only the `gncopy.sh` script needs to be run.

5. Edit the appropriate files to point to the proper GN_DIR (GN_DIR is the full path to the proper operating system tree in gn5c02_00.) You need to do an `ls –a` to see the files. What is your GN_DIR? Do you remember what shell you set to have started for each user?

6. Since you selected the “C” shell for the GeoFrame operators default shell, we need to edit the ".cshrc" login file.

7. Using your favorite editor, set the proper path to the Geonet directory for the correct operating system. You may comment out the other two, if you wish, since they are not necessary.

8. Record your GN_DIR value here: __________________________________________

9. Un-comment the indicated line to use the "GN_USER_DIR" functionality if you wish.

10. Save and Exit.
The login files are now edited to cover the majority of GeoFrame operators you may encounter; those being operators using Sun machines and running the C shell upon logging on. If this is not the case, run the script and then edit the appropriate login files accordingly. In order to finish the process, each GeoFrame Operator must logon and run the **gncopy.sh** script. Then logoff and logon to test.

11 Open another terminal window and switch users to your GeoFrame operator or test account:

```
% su - <your operator account> (Be sure to use the dash.)
```

12 As the GeoFrame operator, navigate to the **gncopy.sh** script and run it.

```
% cd ......./gn5c02_00/gn_examples
% ls
% id (to verify your identity)
% gncopy.sh
```

13 Enter the appropriate parameters as asked for them. Usually the defaults are correct, and a simple carriage return (Enter or Return on your keyboard) will do.

14 Test the GeoFrame Operator's account by completely logging out of the machine and logging in as the new GeoFrame operator.

15 Logout of the machine again.
Online Help

The installation guide for Geonet can be found in the GeoFrame Bookshelf in the GeoFrame System portion. There is also a reference document that provides detailed explanations of the different phases of the installation process and the use of the GeoQuest Installtool.

Additionally, there is a Geonet users guide that can be obtained from the Util in the Geonet window. Also from here you can get to the Geonet Installation guide.

Summary

You have just had an introduction to the GeoQuest installtool, and have learned how to configure an operators’ account to use an environment that comes from a template provided with Geonet.

For existing Unix accounts that need to use Geonet, but cannot have the entire environment changed to that of the template, it is only necessary to set the path to GN_DIR. This is illustrated in Appendix A of this manual.

With an understanding of how the installtool works, the next exercise will focus on installing Oracle and creating a working Oracle database.
Chapter 4
Creating the Oracle Database

Overview

In this chapter you will install Oracle from the GeoQuest Utilities CD and create a working database that will be used by GeoFrame.

After creating the database, you will learn a series of simple checks that can be performed on any database to verify that a viable database exists, all of the necessary elements required by GeoFrame are there, and that the database is running. Furthermore you will gain a better understanding of how GeoFrame will connect to the database, and how to test that the necessary processes and files are present and configured to do so.

After completing these exercises you should have a better overall understanding of basic Oracle architecture.

Keywords

The following keywords are significant in this chapter. After completing the exercises you should have a better awareness of what the keywords mean and their significance:

- ORACLE_HOME
- TNS_ADMIN
- ORACLE_SID
- TWO_TASK
- SQL*Net
- Instance
- Database
- Datafile
- Tablespace
- SGA
- Oracle_Runtime.csh
tnsnsnr
tnsping
connect internal
Exercise 5. Installing Oracle and Creating the Database

At this point you should have a good understanding of what your systems resources are, and how much of a burden Oracle is going to place on them. You are now ready to install Oracle and create a database.

1. **Login to your machine as the Oracle software owner (oraown). If you have switched user in an existing window, verify your ID. Confirm that all of the appropriate permissions will be set.**

   ```
   % id
   % umask 022
   ```

2. **Determine the approximate size of your database. Is there enough disk space where you intend to install? Does the Oracle owner have write permissions there? Total DB size is (Oracle tree) + (Tablespaces)**

   ```
   % df -k /<your oracle installation location>
   % touch /<your oracle installation location>/testfile
   % rm /<your oracle installation location>/testfile
   ```

3. **Make sure the installation environment is clean. Are any environment variables set for ORACLE_HOME, TNS_ADMIN, TWO_TASK, or ORACLE_SID? If so, un-set them.**

   ```
   % env | grep ORACLE_HOME
   % env | grep TNS_ADMIN
   % env | grep ORACLE_SID
   % env | grep TWO_TASK
   ```

   These should return nothing. If they do, use the “unsetenv” command.

Using the Installtool, begin installing Oracle. Note the name of the log file being created. If you like, you can view the log file as it is created in a separate terminal window. This saves you from having to carefully watch the message window in the Installtool and miss important information.
4 Run the same Installtool (from the GeoQuest Utilities CD).
5 Open another terminal; verify you are in your HOME directory.
6 Locate the most recent log file and follow it using the `tail` command,
   
   `% ls -lart
   % tail -f .InstallLog_<most-recent-one>
   
7 Begin progressing through the installation phases for Oracle:
   
   · · · · · ·
   
   **TIP:** Use the arrow to select only the Oracle Server Runtime Tree. The
   Forms/Reports Runtime environment is not required for GeoFrame.
   
8 Verify the Installation Location and the space.
   
   · · · · · ·
   
   **NOTE:** The Space Required that is indicated in the parameter area is only for the
   Oracle software. It does not consider the database we are about to create.
   
9 Indicate the following steps for the installtool to complete:
   
   · Re-link Oracle Executables
   · Create the SQL*Net Configuration Files
   · Setup the Oracle system boot time startup.
   · Create an Oracle instance for a GeoQuest Application (GeoFrame in our
     case)
   · Create Data and Index Tablespaces
10 In the Root Password phase, click "Enter Password" to supply the correct root password. This is for the creation of the startup/shutdown scripts. Verify in the messages that the password is saved.

11 Enter the Installation Parameters Phase.

**NOTE:** In the installation parameters phase of the Oracle installation, be sure to use the scroll bar and scroll down to view all of the parameters!
12 What are you going to call your Oracle instance? How about the Alias (TWO_TASK) that GeoFrame will use to identify your instance?

BEST PRACTICE: Oracle 8.1.6 allows for up to 8 characters for the Instance Identifier. Oracle recommends 4, due to filename length limitations; however, that is not an issue here. A “feature” in the installtool requires the Oracle Instance to be 7 characters or less.

13 The Default locations for TNS_ADMIN and DB_DEF_DISK1_DIR are quite adequate. If the locations don’t exist (as indicated by the red “prohibited” symbol) click the folder icon to create it.

NOTE: Selecting the Datafile Locations here is both a space and a tuning issue. Tuning is covered later in the week. For the scope of this course, create or use the default settings.

14 Where is your Default Data Directory (..../dbs)? The Data directory is very important for obvious reasons. Why do you suppose it is separated from the Oracle Parameters above?

NOTE: Be aware that the passwords for the Oracle users “sys” and “system” are “change_on_install” and “manager” respectively, by default.

15 Set your tablespace sizes. Choose “Small” initially. How does your Database size compare to your initial approximation?

NOTE: Tablespaces can be increased in size later. They can not be decreased easily however. Start small unless you have forecasted the necessary size needed for your shop. Realistically, you would probably choose Large here and end up increasing the sizes manually as well.

16 Manually change the tablespace sizes to the following amount (for the scope of this course only):

- Data Tablespace (TSDATA) = 500
- Index Tablespace (TSINDEX) = 300
The other tablespace sizes can remain as they are.

NOTE: Choosing “Small” here has other implications besides just choosing tablespace sizes for you. Choosing Small also selects default SGA settings that reflect the size of the database. This will be covered in more detail later in the course.

17 Use the default Data and Index Tablespace names

18 Scroll up to double-check your parameters. If you are happy with them, proceed to the next phase.

19 In the messages portion of the Installtool window, or in the terminal that you’ve been tailing the log in, check all of the environment settings that were just set. How does everything look? If everything looks good, proceed to extracting the files.

20 Proceed through to the execution phase. Watch the messages or your .InstallLog. What’s going on? What happened during the root procedures phase?

DISCUSSION: Prior to executing the procedures, look back at the procedures that were selected. What do you anticipate for this phase of the installation? (Primarily with regards to the creation of the Oracle instance and Tablespaces.)

21 Proceed through the execution phase. Keep a careful eye on the message window or your terminal. Any surprises? Continue through to the end of the installation.

22 Did you get any errors? How would you fix the error?

23 In the window you were watching the install log in, type ^C to stop the tail command

You have just created a database and configured an instance that will be your means of accessing that database. In the next exercise, you will learn to determine if the database is opened and if the instance is running. Also, you will learn to check that the database was created correctly and that the requirements for GeoFrame are there.

These are steps that should be performed on any database prior to installing GeoFrame, whether the database was just created, or is an existing database that you intend to use for GeoFrame.
Exercise 6. Checking the Database

Prior to installing the GeoFrame catalog in our database, it is a really good idea to verify the installation and the integrity of our database and the running instance. This is accomplished through a series of checks for various processes and communication links. First, let’s see what we have accomplished so far with regards to all of the Oracle installing we have done.

1. Logon as the Oracle owner and list the contents of the home directory. What new items are there? Look at the `Oracle_Runtime.csh` script. Record the values:

   ```
   ORACLE_HOME = 
   ORACLE_SID = 
   TWO_TASK = 
   TNS_ADMIN = 
   ```

2. Change directories to the various locations in the Oracle owners' home directory. Have a general look around. Any surprises? Is everything there that you would expect at this point?

   We can now use Unix commands to verify that our Oracle database is running, and through a combination of Unix commands and SQL commands, we can verify the integrity of the database that we have created. This will allow us to move on to creating the GeoFrame Oracle Baseline Account with confidence.

   Every properly running Oracle instance will have two types of processes running in the Unix environment. The first type “control” the database (there are six of these) and the second type listens for the SQL*Net connections to the database. The first will all have the instance name in them, and the listener process should be called “tnslsnr.” Check for these.

3. First, check for the Unix processes that are controlling the database:

   ```
   % ps -ef | grep ora_ 
   ```
You should see something like this:

4 Now check for the Unix process that is listening for requests made to the database

   % ps -ef | grep lsnr
   which should look like this:

5 Check to see that the SGA has been allocated. (Remember ipcs? use the –b option).

   % ipcs -b
   If the instance is started, you should see this:

6 How much memory has been allocated to the Oracle instance?
7 How many semaphores? What do these mean?

TIP: The following commands or utilities require the proper Oracle environment variables to be set. You can easily set the environment now, and whenever you need to administer the Oracle database by sourcing the Oracle_Runtime.csh script in the home directory of the Oracle DBA.

8 If you are happy that all of the process are running, you can verify the network connection using the “tnsping” command:

% source Oracle_Runtime.csh

% tnsping  <your TWO_TASK name>

9 You can further check that the database and listener are configured correctly by forcing a connection to the “system” Oracle account, via the listener.

% sqlplus system/manager@<your TWO_TASK name>
(note that there are no spaces around the “@” sign.)

By running SQLPlus and forcing a connection to the database through the listener, we can be fairly confident about the Network connection, listener, and the instance at this point. Now check the database for completeness with regards to tablespaces. Were all of the tablespaces created OK? You could check the default data file location in Unix, but a definitive way is to check using SQL and querying the Database:

10 SQL> desc DBA_TABLESPACES

11 SQL> select TABLESPACE_NAME, STATUS from DBA_TABLESPACES;

12 SQL> desc DBA_DATAFILES

13 SQL> select FILE_NAME from DBA_DATAFILES;

Furthermore, you can see what accounts exist so far in our new database. It's not so crucial as we are currently logged into one of the accounts (system) so we know it exists.

14 SQL> desc DBA_USERS

15 SQL> select USERNAME, PASSWORD from DBA_USERS;

At this point if all of these tests pass, we can be confident that our database is ready for GeoFrame.
The next exercise will demonstrate how to stop and start the database and the listener process in the event that neither is running. The exercise will also help explain and illustrate the scripts that have been created on your system that will stop and start the database at boot time.
Exercise 7. Stopping and starting the Database

It is important for a database to be shut down cleanly. The scripts to do this automatically were made upon installing Oracle and creating the instance using the Installtool. However, it may be necessary to shutdown and startup the database without having to reboot the system. This can be done manually using the Server manager utility.

### NOTE:
Connecting to your database as Oracle user “sys” and shutting down the database is done directly on the Oracle server; and not through the listener. If TWO_TASK is set in your environment, Oracle will by default use the listener, and you won’t be able to connect as sys. You can not connect as sys via the listener.

1. Source the Oracle_Runtime.csh file and set the correct instance name in the environment. This ensures that there is no TWO_TASK environment variable set.

   ```
   % source Oracle_Runtime.csh
   % setenv ORACLE_SID <yourSID>
   (if there is more than one database on your server)
   % svrmgrl
   ```

2. Logon to the Oracle database now as sys. Issuing the `connect internal` command at the server manager prompt accomplishes this.

   ```
   SVRMGR> connect internal
   (you should see “connected”)
   SVRMGR> shutdown
   ```
3 Check that the database has been shutdown, by checking for the Oracle processes and that the memory allocated for the SGA has been de-allocated.

```
SVRMGR> quit
% ps -ef | grep ora_
% ipcs -b
```

4 Now stop the Listener. Quit the server manager first and use the “lsnrctl” commands at the prompt

```
% lsnrctl stat
% lsnrctl stop
% lsnrctl stat
```

5 To start the Oracle database backup again, do the reverse. First start the listener, and then connect to the instance as sys, and startup the database. Check for processes and SGA again.

```
% lsnrctl start
% svrmgrl
SVRMGR> connect internal
SVRMGR> startup
SVRMGR> quit
% ps -ef | grep ora_
% ipcs -b
```

**DISCUSSION:** Go to the `/etc/init.d` directory and list the files there (`ls -lrt`). Look at the Oracle files there. What do they do? (As if their name does not imply it.) What level of shutdown is done?

Move to `/etc/rc0.d` and `/etc/rc2.d`. Do the same thing. Why are these files located where they are? What would have to be done if you wanted to create another Oracle instance on your Oracle Server? What is the importance of the “su oraown –c...” portion of the line in the scripts?
6 Log out of the machine.

Online Help

All of the online help that is associated with the topics covered in the exercise in this chapter is located in the Run Time Oracle installation guide located in the GeoFrame Bookshelf in the GeoFrame System portion.

As with the Geonet documents in the last chapter, portions of the online document regarding installation have been printed and included as Appendix B of this manual.

Summary

At this point you have verified that there exists a viable database, that the database is running, and that you can communicate and make requests of the database through the Listener process. These items are precisely what will be required in the next chapter in which you install GeoFrame onto the system, and create the necessary Oracle accounts for GeoFrame where the GeoFrame catalog will be placed.
Overview

In this chapter you will review some important parameters required for GeoFrame installation, and verify that the database is running. Also, you will verify communication to your Oracle database from your GeoFrame host (which may not always be the same machine.) After installing GeoFrame the focus turns toward what has happened in the Oracle database to make it a “GeoFrame database.

After completing these exercises you will have a good understanding of how to install GeoFrame. Much of the emphasis is placed on watching the Installation Log as you are installing GeoFrame to get a feel for what is going on in Oracle during the creation of the baseline account.

Keywords

The following keywords are significant in this chapter. After completing the exercises you should have a better awareness of what the keywords mean and their significance:

```
DB_DEF_DISK1_DIR
SYS_PWD
SYSTEM_ACCNTPASSWD
LM_LICENSE_FILE
FLEXLM_BIN_DIR
schema
Baseline Account
gf_accounts
gf_users
```
Prior to installing, it is a good idea to have on hand some of the information about your Oracle Instance, as well as some other installation parameters you will need later in the installation procedure. By now, you should have a general idea of where GeoFrame will reside and what should be necessary in order to connect with your instance.

---

**NOTE:** Are you installing GeoFrame on the same machine as Oracle? If not, do you need to install the Oracle binaries on the GeoFrame host?

Regardless of where your Oracle binaries reside, you will need access to some of the SQL*Net files in order to define the TWO_TASK you will use for GeoFrame.

1. Write down a few of your more important Oracle parameters:

   - **ORACLE_HOME**
   - **TNS_ADMIN**
   - **TWO_TASK**
   - **DB_DEF_DISK1_DIR (..../dbs)**
   - **SYS_PWD** (password for Oracle account “sys” on the Oracle Server machine)
   - **SYSTEM_ACCNTPASSWD** (same as above, but for “system”).

A few other useful parameters you might want to have handy are:

   - **GN_DIR**
   - **LM_LICENSE_FILE** (the location of your license)
   - **FLEXLM_BIN_DIR** (the flexlm bin directory.)
2 Verify that your database is up and running. Ping your database using the **TWO_TASK**
that you intend to use when installing GeoFrame.

**Logon as gfown**

% source Oracle_Runtime.csh

% tnsping <your TWO_TASK name>

% cd

The Installtool must be able to connect to the database in order to create the
GeoFrame baseline account and install the schema. Once we are convinced that the
connection to the database is good and the database is running, we can proceed to
installing GeoFrame.

The following checklist contains all of the parameters you would encounter during
the installation of GeoFrame. All of these parameters may not appear due to the
installation steps you will have chosen in the next exercise. You may fill in some of
these values if you would like, or wait and enter them into the Installtool
“Installation Parameters” phase in the next exercise.
### Installation Planning Checklist

#### Select Options (Solaris)

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Default Value</th>
<th>Your Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Oracle Parameters</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORACLE server location</td>
<td>ORACLE_INST_DIR</td>
<td>$HOME/geoframe_402_sunn</td>
<td></td>
</tr>
<tr>
<td>ORACLE_HOME directory</td>
<td>ORACLE_HOME</td>
<td>$HOME/geoframe_402_sunn</td>
<td>may be set the same as $GF_PATH</td>
</tr>
<tr>
<td>Oracle Network Admin directory</td>
<td>TNS_ADMIN</td>
<td>$HOME/network/admin</td>
<td></td>
</tr>
<tr>
<td>Default Oracle datafile directory</td>
<td>DB_DEF_DISK1_DIR</td>
<td>$HOME/dbs</td>
<td></td>
</tr>
<tr>
<td>Default Oracle second datafile directory</td>
<td>DB_DEF_DISK2_DIR</td>
<td>$DB_DEF_DISK1_DIR</td>
<td></td>
</tr>
<tr>
<td>Oracle logfile directory</td>
<td>DB_LOG_DIR</td>
<td>$DB_DEF_DISK1_DIR</td>
<td></td>
</tr>
<tr>
<td>Oracle system tablespace directory</td>
<td>DB_SYSTEM_TS_DIR</td>
<td>$DB_DEF_DISK1_DIR</td>
<td></td>
</tr>
<tr>
<td>Oracle system tablespace size</td>
<td>DB_SYSTEM_TS_SIZE</td>
<td>Small 150</td>
<td></td>
</tr>
<tr>
<td>Oracle rollback directory</td>
<td>DB_ROLLBACK_DIR</td>
<td>$DB_DEF_DISK1_DIR</td>
<td></td>
</tr>
<tr>
<td>Oracle rollback size (Mb)</td>
<td>DB_ROLLBACK_SIZE</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>Oracle tablespace directory</td>
<td>DB_TSDATA_DIR</td>
<td>$DB_DEF_DISK2_DIR</td>
<td></td>
</tr>
<tr>
<td>Oracle tablespace size (Mb)</td>
<td>DB_TSDATA_SIZE</td>
<td>200</td>
<td></td>
</tr>
<tr>
<td>Oracle tablespace index directory</td>
<td>DB_TSINDEX_DIR</td>
<td>$DB_DEF_DISK1_DIR</td>
<td></td>
</tr>
<tr>
<td>Oracle tablespace index size (Mb)</td>
<td>DB_TSINDEX_SIZE</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>Oracle tablespace temp directory</td>
<td>DB_TEMP_DIR</td>
<td>$DB_DEF_DISK1_DIR</td>
<td></td>
</tr>
<tr>
<td>Oracle tablespace temp size (Mb)</td>
<td>DB_TEMP_SIZE</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>Oracle service name</td>
<td>TWO_TASK</td>
<td>ora8gq ... <em>CHANGE</em></td>
<td></td>
</tr>
</tbody>
</table>

Chapter 5-4: GeoFrame 4 System Administration
<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Default Value</th>
<th>Your Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle instance identifier</td>
<td>ORACLE_SID</td>
<td>geoquest1 ... <em>CHANGE</em></td>
<td></td>
</tr>
<tr>
<td>Oracle sys account password</td>
<td>SYS_PWD</td>
<td>change_on_install</td>
<td><em>DO NOT CHANGE</em></td>
</tr>
<tr>
<td>Oracle system account password</td>
<td>SYSTEM_ACCNTPASSWD</td>
<td>manager</td>
<td><em>DO NOT CHANGE</em></td>
</tr>
<tr>
<td>Index tablespace name</td>
<td>TSINDEX</td>
<td>TSINDEX</td>
<td></td>
</tr>
<tr>
<td>Data tablespace name</td>
<td>TSDATA</td>
<td>TSDATA</td>
<td></td>
</tr>
<tr>
<td>Baseline account password</td>
<td>BASELINE_PWD</td>
<td>GF4_0_2 ...</td>
<td></td>
</tr>
<tr>
<td>Baseline account name</td>
<td>BASELINE_ACC</td>
<td>GF4_0_2 ...</td>
<td></td>
</tr>
<tr>
<td>GeoFrame system password</td>
<td>GF_PROJSYS_PASSWD</td>
<td>GF_SYS</td>
<td><em>DO NOT CHANGE</em></td>
</tr>
<tr>
<td>Copy DSLs from TWO_TASK</td>
<td>FROM_TWO_TASK</td>
<td>geoquest1 ... <em>CHANGE</em></td>
<td></td>
</tr>
<tr>
<td>Copy DSLs from Account Name</td>
<td>FROM_ACC</td>
<td>GF3_8 ... <em>CHANGE</em></td>
<td></td>
</tr>
<tr>
<td>Copy DSLs from Account Password</td>
<td>FROM_PWD</td>
<td>GF3_8 ... <em>CHANGE</em></td>
<td></td>
</tr>
</tbody>
</table>

### Application Parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charisma Uniras home directory</td>
<td>UNIHOME</td>
<td>$HOME/charisma/uniras</td>
</tr>
<tr>
<td>Charisma site directory</td>
<td>GM_SITE_DIR</td>
<td>$HOME/charisma/client</td>
</tr>
<tr>
<td>Charisma project directory</td>
<td>DB_GMHOME_DIR</td>
<td>$HOME/charisma/projects</td>
</tr>
<tr>
<td>GeoFrame home directory</td>
<td>GF_HOME</td>
<td>$HOME</td>
</tr>
<tr>
<td>Bulk Server Address</td>
<td>BULK_SERVER_ADDR</td>
<td>LOCAL:</td>
</tr>
<tr>
<td>SITE mode Ctree Server owner</td>
<td>SITE_CTREE_USER</td>
<td></td>
</tr>
<tr>
<td>SITE mode Ctree Server base directory</td>
<td>SITE_CTREE_HOME</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Name</td>
<td>Default Value</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-----------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>SITE mode Ctree Server name</td>
<td>SITE_CTREE_NAME</td>
<td>FAIRCOMS</td>
</tr>
<tr>
<td>Allow private project disk</td>
<td>PRIVATE_DISK_OPTION</td>
<td>off</td>
</tr>
<tr>
<td>Geonet Home Directory</td>
<td>GN_DIR</td>
<td>/usr/local/gn5c02_00/solaris</td>
</tr>
<tr>
<td>Charisma scratch directory</td>
<td>GM_SCRATCH_DIR</td>
<td>$HOME/charisma/scratch</td>
</tr>
<tr>
<td>Charisma/InDepth product directory</td>
<td>GM_PRODUCT</td>
<td>InDepth</td>
</tr>
<tr>
<td>GeoFrame plotting Scratch Directory</td>
<td>GF_HARDCOPY_DIR</td>
<td>$GF_HOME/hardcopy_files</td>
</tr>
<tr>
<td>Directory containing compilers</td>
<td>COMPILERS</td>
<td>/opt/SUNWspro/SC5.2</td>
</tr>
<tr>
<td>General Parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLIS Data File Path</td>
<td>DLIS_DATAPATH</td>
<td>$GF_HOME/dlis_files</td>
</tr>
<tr>
<td>FlexLM License File</td>
<td>LM_LICENSE_FILE</td>
<td>/usr/local/flexlm/licenses/license.dat</td>
</tr>
<tr>
<td>FlexLM bin directory</td>
<td>FLEXLM_BIN_DIR</td>
<td>/usr/local/flexlm/bin</td>
</tr>
<tr>
<td>FlexLM license file to be updated</td>
<td>FLEXLM_LICENSE_FILE</td>
<td>/usr/local/flexlm/licenses/license.dat</td>
</tr>
<tr>
<td>FlexLM new license Keys file</td>
<td>FLEXLM_KEY</td>
<td>$HOME/license.dat</td>
</tr>
<tr>
<td>Geoframe Environment Umask</td>
<td>GF_UMASK</td>
<td>002</td>
</tr>
<tr>
<td>Open windows home directory</td>
<td>OPENWINHOME</td>
<td>/usr/openwin</td>
</tr>
<tr>
<td>Motif home directory</td>
<td>MOTIF.HOME</td>
<td>/usr/dt/</td>
</tr>
<tr>
<td>Is GF4.0 for SGI installed</td>
<td>GF_SGI_INSTALLED</td>
<td>NO</td>
</tr>
<tr>
<td>GF4.0 SGI Baseline Location</td>
<td>GF_SGI_PATH</td>
<td>$GF_PATH/geoframe_4_0_2_sgil</td>
</tr>
<tr>
<td>Create Local Baseline Dir?</td>
<td>CREATE_LOCAL_DIR</td>
<td>No</td>
</tr>
</tbody>
</table>
You have verified that the Oracle database is ready for the GeoFrame installation. This could be any database, not necessarily one created for GeoFrame. It must pass these tests though, or else the GeoFrame installation will fail.

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Default Value</th>
<th>Your Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Baseline Directory</td>
<td>L_BSL_TARGET_DIR</td>
<td>$GF_PATH/local_geoframe_402_sun</td>
<td></td>
</tr>
<tr>
<td>Copy from Local Source Area?</td>
<td>COPY_SRC</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Local Source Area</td>
<td>L_BSL_SOURCE_DIR</td>
<td>$GF_PATH ... <em>CHANGE!</em></td>
<td></td>
</tr>
</tbody>
</table>
Exercise 9. Installing GeoFrame

You should be fairly confident with your Oracle database now, and have a good idea of how you are going to connect to it from GeoFrame. In addition to installing the GeoFrame applications in this next exercise, we will also be creating certain Oracle accounts and importing the GeoFrame schema (the tables, et.al.) into an account that will be created in our database. This is the important part, and is what will finally make our database a GeoFrame database.

1. Login as the GeoFrame software owner. If you have switched users in an existing window, verify your ID. Confirm that all of the appropriate permissions will be set.

   % id
   % umask 022

2. Approximate your installation size. Is there enough disk space where you intend to install? Does the GeoFrame owner have write permissions there?

   % df -k </your geoframe installation location>
   % touch </your geoframe installation location>/testfile
   % rm </your geoframe installation location>/testfile

3. Make sure the installation environment is clean. Are any environment variables set for ORACLE_HOME, TNS_ADMIN, TWO_TASK, or ORACLE_SID? If so, un-set them, or close the terminal and open a new one. (You probably need to do this from the last exercise, during which you sourced the Oracle_Runtime.csh file. If so, do it and start at step 1 again.)

   % env | grep ORACLE_HOME
   % env | grep TNS_ADMIN
   % env | grep ORACLE_SID
   % env | grep TWO_TASK
   These should return nothing. If they do, use the "unsetenv" command.
4 Run the Installtool on the GeoFrame 4 CD. If the media has been loaded onto your machine, run the Installtool located in /media/gf4 (or the location specified by your instructor.) Note the name of the log file being created.

5 Open another terminal; verify you are in the HOME directory.

   % cd

   % ls -lart

   % tail -f .InstallLog_<most-recent-one>

   It is easier to watch for important messages in the terminal window, rather than trying to catch information as it scrolls past in the message area of the Installtool. Additionally, during parts of the install, there is additional information in the log that is not seen in the Installtool message window.

6 Begin progressing through the installation phases for GeoFrame:

   • • • • • •

   TIP: It’s always a good idea to select custom install. Always keep control of what’s going on, and know what’s going on.

7 Proceed to the Feature Key File phase. If you prefer, provide the necessary license information. It is not necessary at this point.

   • • • • • •

   NOTE: The Installtool only wants to scan your license in order to automatically select installation steps for you, based on the products for which you have purchased a license. There are no mandatory parameters until you reach the parameters phase.

8 Select to install the entire GeoFrame Suite. You may use the arrows to view the sub-menus to see the products. But unless you have space issues, it’s recommended to install the entire suite of applications. For the scope of this course however, you don’t need to include the GeoFrame Developers kit or the GeoQuest Internal Software.

9 Verify the Installation Location and the space. Note that the Space Required that is indicated in the parameter area is only for the GeoFrame software.

10 Select the following installation steps:

   • Install GeoFrame Database Schema
   • Install Uniras
   • Install GM_SITE_DIR
   • Setup Flexlm startup files
11 Click "Enter Password" to supply the correct root password. This is for the creation of the startup/shutdown scripts. Verify in the messages that the password is saved.

12 Proceed to the Installation Parameters phase.

NOTE: In the installation parameters phase of the GeoFrame installation, be sure to use the scroll bar and scroll down to view ALL of the parameters!

The first section of parameters specifies the information needed by the Installtool to connect to the database and create the necessary Oracle accounts and install the schema. These are very important. Leave the defaults unless you are instructed to do otherwise.

13 Enter a valid ORACLE_HOME. This is primarily for the Oracle binaries.
14 Specify the path to the database files on the Oracle server, regardless of whether or not they are mounted on the machine onto which you are installing GeoFrame.

15 Specify the TWO_TASK.
   The application parameters give the Installtool the necessary information for other applications such as the Ctree server & Charisma. Unless instructed otherwise, use or create the default values specified by the Installtool.

16 Specify or create the GM_SITE_DIR. The default that it tries to use is fine.

17 The GeoFrame owner can be accountable for the Ctree server for the scope of this course. Choose/create a directory for it, and specify gftown as the owner.

18 Change InDepth Product to Charisma.
   General Parameters inform the Installtool of any other information necessary to complete the install. Unless instructed otherwise, use or create the default parameters specified by the Installtool.

19 You must specify the correct Geonet Home Directory, GN_DIR.  
   Don’t click to create the default location!  
   Remember, this is a full path to the Operating System directory, for instance: 
   /apps/gftown/gn5c02_00/solaris.

20 Specify LM_LICENSE_FILE if you like.

21 Provide the correct path to the flexlm/bin. It should be located at $GN_DIR/flexlm/bin.

22 Review all of your parameter settings prior to proceeding to the extraction phase. Pay close attention to the Oracle Parameters.

23 Proceed to the Ready for Extraction phase. Prior to extracting, check your log to see that the desired environment variables were indeed set, once again checking Oracle parameters, and the Geonet Directory, GN_DIR.

24 Select Next to begin extracting.

25 When you begin to execute procedures, watch the log. Did you have any errors? How are you going to go about fixing them? Will you need to run the Installtool to fix them?

• • • • • •

NOTE: If you received an error, a good plan is to look at the script that failed. If you can figure out what went wrong, you can re-run the script after you correct the problem and set the necessary environment variables.
26 Quit the Installtool & log out of the machine.

At this point you have installed GeoFrame. The final exercise in this chapter will simply illustrate what was just done inside the Oracle database. It will also give you a better understanding of the accounts in Oracle that are required for GeoFrame.

Exercise 10. Checking the GeoFrame Installation

The installation of GeoFrame completes the installation process and the creation of our database. There are a few administrative tasks left regarding configuring server applications; however, we should now have a complete database, loaded with the GeoFrame data model, and all of the necessary Oracle accounts created. Now, prior to configuring and preparing GeoFrame for project creation, is a good time to verify that the database is entirely complete; accounts, tables, etc.

1 Logon as your GeoFrame operator or a test account that has been configured to use Geonet.

2 Was the GeoFrame-Geonet registration a success?
3 Open a GeoFrame xterm window. Using some GeoFrame utilities and sqlplus queries, check the database for the baseline account, as well as other GeoFrame accounts.

% gf_accounts
% gf_users
% sqlplus system/manager

SQL> desc DBA_TABLESPACES
SQL> select TABLESPACE_NAME, STATUS from DBA_TABLESPACES;
SQL> desc DBA_DATA_FILES
SQL> select FILE_NAME from DBA_DATA_FILES;
SQL> desc DBA_USERS
SQL> select USERNAME, PASSWORD from DBA_USERS;
SQL> connect gf4_0_2 (then enter the password)
SQL> select * from cat;
If you are comfortable with SQL, have a look around.

4 What did these tell you? More importantly, is to think about how they did it.

DISCUSSION: Baseline, catalog, and schema are sometimes used very loosely and interchangeably. They are different however. What is each one? How does all of this tie into other versions of GeoFrame?

5 Exit SQL and close the GeoFrame xterm.

Now you should have a complete GeoFrame environment. Aside from some remaining configuration and perhaps some site specific details, you are about ready to begin using GeoFrame.
Online Help

As with Geonet and Oracle, a portion of the GeoFrame installation guide has been printed and included as Appendix C in this exercise guide. With the successful integration of GeoFrame with Geonet during the installation procedures, all of the help documents are available in the GeoFrame bookshelf.

The GeoFrame Installation guide contains information about installing GeoFrame as well as information regarding post-installation tasks that will be useful for the exercises to follow.

In addition to the installation and post installation information, and the GeoFrame Installation Guide, there are several different installation scenarios for some of the most common system environments. These scenarios might more closely resemble the steps required to create the GeoFrame environment at your workplace.

Summary

The exercises in this chapter covered the installation of GeoFrame. You should now have an understanding of the steps involved in creating a GeoFrame environment, as well as knowledge of the relationship between GeoFrame and the Oracle database.

Completion of the exercises to this point marks the end of the necessary installations required to create the GeoFrame environment. There does however remain some necessary configuration of GeoFrame as well as the FLEXlm license manager in order to begin using GeoFrame.
Chapter 6
GeoFrame Configuration

Overview

This chapter covers the steps that are necessary to configure GeoFrame in order to allow users to begin to create projects and begin loading data. Additionally, there are a couple of exercises that involve setting up site specific requirements such as using GeoFrame with multiple databases, and creating tablespace pairs that can be for the use of a specific department or purpose. Although these are Oracle administration issues, they exist in this chapter because they are part of configuring the GeoFrame environment.

Prior to all of that, there is an exercise covering the configuration of the License server, which will be required in order to start GeoFrame.

Keywords

The following keywords are significant in this chapter. After completing the exercises you should have a better awareness of what the keywords mean and their significance:

- **hostid**
- **daemon**
- **LIC_DIR**
- **LIC_BIN**
- **GeoFrame_Runtime.csh**
- **GF_DBA**
- **DSL**
- **tnsnames.ora**
- **Shared (Data Type)**
- **ADI_DEBUG**
- **Shared**
- **Sub**
- **Standalone**
- **catalog**
Exercise 11. License Management

This course may utilize a demo license. The demo license configuration now requires configuration of the FLEXlm software. Previously a Demo license contained “DEMO” in all of the feature keys, and a running license daemon was not even needed. With the new demo licenses (or a real license) however, certain parameters need setting properly in order for the license server to work properly.

One of the objectives of this exercise is to realize that to this point in creating the GeoFrame environment, there was no need for a license, nor knowledge of where it would reside. Every parameter regarding the licensing can be changed at any time.

1. The FLEXlm software was installed with Geonet. Logon to your machine as the Geonet owner (gfown).

2. Locate your license file. Move it and rename it if necessary.

   **NOTE:** If the instructor retrieved the licenses via e-mail on a Windows PC, you most likely will have to perform the `dos2unix` command on the license file, and redirect the output to another file. This will remove the unnecessary characters from your license file.

3. What is the host id of your machine? Does it match the host id in your license?

4. The name of the daemon you will be running reflects the version of FLEXlm you are running. Go to the FLEXLM_BIN directory and locate the license daemon you will be running. (You are using FLEXlm version 7.2e)

5. Edit the DAEMON line in your license file to correctly define the daemon in each FEATURE line. You only need to edit the DAEMON line.

   **NOTE:** The daemon in each feature line is sort of an alias. The alias specified in each FEATURE line needs to be defined in the DAEMON line by specifying the full path to an actual executable daemon.
This should be all of the editing you will need to do to your license file. Now it is necessary to make sure the license server is running.

6 Look at the startup script that was created during GeoFrame installation (flexlmstart.sh). It is located in /etc/init.d. Are the “LIC_DIR” and “LIC_BIN” parameters set correctly?

7 Look at the rest of the script. Which user is being switched to, in order to issue the proper commands? Is it right?

8 What’s the proper usage for starting and stopping the license manager?

9 What should your license file be called?

10 Where is the log being written to? Will you be able to write there? Edit the location of the log if necessary.
   Edit the startup/stop script as needed so the FLEXlm daemon will have all of the correct information it needs.

11 Try stopping and starting your License Manager using the scripts. When you stopped it, what information did it provide?

12 What information should we specify to GeoFrame for LM_LICENSE_FILE in order for GeoFrame to find a license?

13 Edit the LM_LICENSE_FILE variable in the Geoframe_Runtime.csh script to look to the running daemon for a license now, instead of a license file. This should be located in $GF_HOME/geoframe_381_sun.

You have now forced GeoFrame to acquire its licenses from a license server as opposed to a file that resides locally.

The location that GeoFrame looks to for a license can be modified at any time by simply editing the LM_LICENSE_FILE environment variable in the GeoFrame_Runtime.csh
Exercise 12. GeoFrame Administration

Prior to creating projects, there are a few more tasks that must be performed as the GeoFrame Database Administrator. This includes defining a global pool of storage locations, and granting users access to them as well as managing the tablespaces that are available for use by the catalog. Also in GeoFrame 4, the GeoFrame Database Administrator can manage the match rules for the catalog as well as peripheral devices such as tape drives.

1. Logon as your GeoFrame operator.

   Assign the DSLs for use by this catalog. This is a requirement prior to beginning to use the GeoFrame installation.

2. What are the actual disk locations that are intended for GeoFrame Data Storage? Open an Xterm and switch users to gfown (`su -l gfown`) and create some for your use. Start in `/apps/gfown`, and make some DSLs that reflect your machine name. Finally, make sure everybody in the group will have read & write privileges to the DSL location.

3. Run GeoFrame from Geonet to open the Project Manager window. What database are you using? What is available to you? Can you connect to anything?

   Access the DBA mode.

4. Select the Project Management tab and click on “DBA”.

5. Enter the DBA password (default was `gf_dba`).

6. Now connect to the catalog or projects you are interested in. Move back to the login tab. What is available now? Connect to it.

7. After you have connected to the catalog, move back to the Project Management tab. You should see now that all of the Database Administration choices are available.

8. Select “Disks...” Select the storage type, to which you wish to assign disks, and navigate to the desired location in the upper portion of the screen. Apply your selections for each storage type.
Now you must grant users the privilege of creating projects using this catalog. This is also a required step. This involves first adding the operator to the list of users, and finally assigning them DSLs from the global pool of DSLs for the catalog.

9 Select “Users...” to bring up the Users (DBA) Dialog box. What is the top portion of the window showing you?

10 Add the Unix username of all of the people you want to be able to use GeoFrame in the lower right hand corner of the window. This will allow you to assign storage locations to these people, which they in turn will be able to select for their projects.

11 Finish granting the users’ privileges by selecting the data types each user may store, and the locations in which they may store it.

You can view the Tablespace management options available to you at this point.

12 Select “Tablespaces...” to access the Project Group Manager window. What tablespaces are available for you? When were these registered? Did they have to be then?

13 Check the status of the tablespaces. Are they online? Are there any projects created using these tablespaces?

In the next portion of the exercise you will register a tape drive with the catalog. Upon registering the device here, it becomes available for use by the GeoFrame system for backups, etc. This step is not required to begin using the GeoFrame installation, and can be performed anytime a new device needs to be registered with the catalog.

14 Select “Devices...” to access the I/O Device Configuration window.

15 There should be no configured devices at this point. In the Device Editing portion of the window you can select the type of device you would like to set up. Select Tape to configure the tape drive connected to the instructors machine.

16 In the Name area, you may choose any name you wish. This is the way that you and all of the users of the catalog will identify this device.

17 Host Name should be the hostname of the machine to which the device is connected. The instructor will specify the hostname for you.

18 Actual Device Name is going to be the full path to the device on the host machine. Unless otherwise specified by the instructor, use /dev/rmt/0n. For an 8mm drive, the Media Capacity is 10Gb (depending on the tape.)
19 Choose OK to finish configuring the device. You can verify the configuration by selecting “Recover...” in the catalog portion of the project management window. In the Source portion of the window, look for the device that you registered with the catalog. You should see the name you gave the device along with the 2 default choices, File and Tape.

20 Select the device that you configured. Did you receive any error messages? Did you spell the hostname correctly? Do you have remote shell capabilities on the host you specified?

The final button in the GeoFrame Database Administration portion is the Match Rule Editor. As with the Devices, this is not a required step to begin using GeoFrame, however, only if the default Match Rule Template will suffice. It is here that you can create Match Rule templates for use by the operators. Upon creation of a project, the user can choose a Match Rule Template from those available in the catalog.

21 Select “Match Rule Editor...” to access the Match Rule System Editor.

22 Use the “folder” icon, or select file > open from the menu, and open the Default Match Rule Template.

23 For each Data Item, you can view the Match Key by specifying the desired Data Item in the drop down list. View some of the keys in the default template to get an idea of how the system determines data is identical by default.

24 Add Name as a match key for the data type well.

25 Add as many Match Keys for as many Data Types as you would like.

26 In the menu bar, select File > Save as... to save your template. Specify any name you would like for your template, and then click OK.

27 Select File > Exit to exit the Match Rule Editor.

You have created a template for the system. This template will now be available as a selection upon project creation. Once the template is chosen upon creating a project, it shouldn’t be changed for the life of that project. With identical data being identified by the Match rules, it is necessary to specify how to handle the identical data, or Merge it. This is accomplished by creating a set of Merge Rules, much in the same way the Match rule template was created. Merge templates can be created by the GeoFrame DBA at the catalog level for use by the entire system, or at the project level, in which case that Merge template can only be utilized within that project.
28 Exit GeoFrame.
Let’s now take a closer look at the TNS_ADMIN directory and what it contains. A good understanding of these files will be very beneficial when you are installing/configuring GeoFrame in order to connect to your Oracle database, or to anyone else’s.

The main objective of this exercise is to learn to configure the SQL*Net in order to see multiple databases with GeoFrame. Additionally, you will finish this exercise with a better understanding of how GeoFrame exactly gets the TWO_TASK.

This exercise can be done while you are logged in as your GeoFrame operator. This will allow you to stop and start GeoFrame for the purpose of illustrating the changes you will be making. The edits you will be making to the Oracle files are best done in a separate window in the second display having switched users to oraown.

1. Navigate to your TNS_ADMIN directory. List the contents of the directory. You should see the following files (minimum.)

   listener.ora
   tnsnames.ora
   tnsnav.ora
   sqlnet.ora

   **DISCUSSION:** View each of the files in the TNS_ADMIN directory. Have a discussion about what each file contains. Do the settings look familiar?

2. Which settings do you think are absolutely critical for clients to be able to connect to the database in the **listener.ora** file? How about the **tnsnames.ora** file?
3 Edit your `tnsnames.ora` file to be able to connect to everybody else’s database. Copy the portion of the file that defines a TWO_TASK. Use the information on the board describing your classmate’s databases.

---

**NOTE:** The `tnsnames.ora` file you want to edit is the one located in the `TNS_ADMIN` directory that GeoFrame uses to connect to Oracle. This is specified in the `GeoFrame_Runtime.csh` file located in `$GF_HOME`.

4 Stop and Start GeoFrame again.
At this point, when you connect to another person’s database you will have to logon as the GeoFrame Database administrator and add your Unix username to the list of users in their catalog. Likewise, you will have to register your DSLs.

5 Add your DSLs to their catalog. You won’t be able to use the DSLs they registered already, unless they have shared them out to you, and you have mounted them on your machine.

6 Add yourself (your Unix GeoFrame Operator) to the list of privileged GeoFrame users in everybody else’s catalog.

**DISCUSSION:** Look at the `$GN_DIR/config/geoframe402.db` file. You can see what your default TWO_TASK is set to. Discuss the ways to change the default.
Exercise 14. Creating Tablespace Pairs

Although there are performance issues related to creating new tablespace pairs, it is also often done to fulfill site specific requirements. For example, certain tablespace pairs to be used by certain business units. In this sense, it makes creating tablespace pairs part of configuring GeoFrame and the Oracle Database for use.

Tablespaces must be created in Oracle prior to registering them for use in GeoFrame. This can be done using SQLplus or svrmgr.

1. Open a GeoFrame xterm to run sqlplus, or switch users to oraown in a regular xterm and source Oracle_Runtime.csh.

   **NOTE:** Remember that when you create a tablespace in an Oracle database, it has got nothing to do with GeoFrame – until you register the tablespace pair for use in GeoFrame. Therefore, the Oracle account you will connect to is “sys” or “system”, not one of the GeoFrame accounts. When you create tablespace pairs in an Oracle Database, they are there for everybody to see, and for everybody to use.

2. % sqlplus system/manager

   Creating Tablespaces requires issuing an SQL command to create the tablespace followed by the name of the tablespace, then the physical name and location of the Unix datafile that will make up the tablespace. Additionally, you must provide information about the size of the datafile, and the parameters that define how the tables are allowed to grow within the tablespace.

3. At the SQL prompt, issue the following command to create the tablespace:

   ```sql
   SQL> create tablespace CSDATA
   datafile '/apps/oraown/dbs/CSDATAgf40.dbf' (or your own choice here)
   size 200M reuse
   default storage (initial 10K next 10K minextents 1 maxextents 100 pctincrease 10) online;
   ```
4 Now create an INDEX tablespace to complete the pair. Typically, the INDEX tablespace can be approximately 75% of the size of the corresponding DATA tablespace.

```sql
SQL> create tablespace CSINDEX
datafile '/apps/oraown/dbs/CSINDEXgf40.dbf' (or your own choice here)
size 150M reuse
default storage (initial 10K next 10K minextents 1 maxextents 100 pctincrease 10) online;
```

5 You can create additional pairs in your database, or in everybody else's database if you like. Remember to keep the sizes of the Unix data files small (10M) initially. We can always add to a tablespace later.

*TIP:* If you created a tablespace in error, you can remove empty tablespaces by dropping them in Oracle:

```sql
SQL> DROP TABLESPACE <tablespace_name>;
```

6 Exit SQL*Plus by typing `exit` at the prompt.

7 Close the GeoFrame xterm.

8 Now it is necessary to register the tablespaces for use in GeoFrame.

9 Start GeoFrame and enter the DBA mode.

10 Connect to the desired baseline account in Oracle. In our case, GF4_0_2.

11 Return to the Project Management window and select the **Tablespaces**... button.

12 Select **Create**... to register the tablespace pairs.

13 Choose a name and a description for the pair. Select your new “DATA” Tablespace for the Project Tablespace. Do the same for your new “INDEX” tablespace.

Now you have configured GeoFrame for use, satisfying all of the GeoFrame requirements as well as any site specific requirements that might exist. The next exercise will allow you to see and use all of the items you have configured while also demonstrating a new method of viewing the project creation process.
Exercise 15. Creating Projects

When you create a new project in GeoFrame, many significant things happen within the Oracle Database. The following exercise is intended to illustrate this and to introduce a new method of viewing the console that may prove itself useful for diagnosing problems in the future.

In a normal console, there is a lot of useful information scrolling by when you start GeoFrame and create a project. You can receive even more messages from Oracle by setting an environment variable called ADI_DEBUG in a GeoFrame xterm and running proman from there. The higher the value you set for ADI_DEBUG, the more information you receive.

1. Logon as your GeoFrame operator. Open a GeoFrame xterm window.
2. % setenv ADI_DEBUG 500
3. Start GeoFrame by typing proman at the prompt.
4. What’s happening in Oracle when you first start GeoFrame?
5. Create a new Standalone project. What Match Rule templates are available to you? Is the one you created there? Select it. Watch the xterm window.
6. Do not choose to create the Charisma extension.
7. When it comes time to choose project parameters, choose the following simply for the purpose of moving on:
   - Use the default Project parameters. Enter a description if you would like.
   - For Display coordinate system, select Set Projection.... (what happened in the xterm?)
   - Click Create... (Did you see that?)
   - In the Projection area below, select US State Planes Coordinate Systems, and select one of the Alabama systems. It’s quite arbitrary for the scope of this course.
   - Back in the Set Projection System window, choose your newly created coordinate system and click OK.
• For Storage coordinate system, click Set Projection... and choose your newly created coordinate system again. Click OK.

• Click OK to carry on with the project creation. Watch the xterm window.

8 Create another project. This time create a Shared project. You can use the disks that are assigned to you by the DBA. Be sure to select Shared Project. You may choose any Match Rule System that you would like.

9 When the Storage Settings dialog box appears, view the DSLs for each Storage Type. What Storage types are used by a Shared Project?

10 You can create the same coordinate system as the last project.

11 Click on Login in the Project Manager window. Recognize the Type of the Shared project.

12 Return to Project Management. Create at least two Sub-Projects. Select Sub Project from the choices of project types. What happened to the ability to select a Match Rule System? Why? Discuss this.

13 Click OK to proceed to Selecting a Shared Project. Choose the Shared Project that is available to you.

14 Proceed through the steps to create the Sub Project. Notice that the coordinate system of the shared project has already been selected for you as a default.

15 After creating a couple Sub Projects, click on Login in the Project Manager window. Recognize the type of the sub projects.

16 Close the Project Manager when you are finished creating your projects.

DISCUSSION: What happened in Oracle when you created a project? Were Oracle accounts created? Use some GeoFrame utilities to verify this. What is the relationship between the Shared and Sub Projects?

17 Now experiment with creating projects in other people’s databases, and using different tablespace pairs. Create as many as you can in the time allotted by the instructor. Continue to choose not to create the charisma portion of the projects to save time.
Online Help

Most of the online help for this chapter is located in the post-installation tasks in the GeoFrame Installation guide. This will include the GeoFrame admin portions of the exercises and the FLEXlm configuration.

Information about the Oracle portions of this chapter, SQL*Net and Creating tablespaces, can be found in the Oracle installation guide as well as the Database administration guide also located in the GeoFrame System portion of the GeoFrame Bookshelf.

Since FLEXlm is a third party product, you may find additional information at their web site: http://www.Globetrotter.com

Summary

You now know the steps required of the GeoFrame DBA that must be performed in order to make the GeoFrame environment operational. You should have a better awareness of the significance of the GF_DBA Oracle account as well as the role of the other “GeoFrame” Oracle accounts.

From here the exercises in this course turn towards administering the environment as it is configured. As the amount of administration required for the GeoFrame software itself is minimal, you will notice that the remainder of the course deals with the Oracle database and are at the “Database level.”
Overview

If for a moment you disregard the actual GeoFrame software and focus on the Oracle portion of the GeoFrame environment, you can stand back and imagine the course being divided into three main categories: creating the database, maintaining the database (i.e., keeping it running as smoothly as possible), and finally recovering the database from a failure, which involves maintaining appropriate backups.

At this point the database is running and operators are using it. The focus will change now to monitoring the database and ensuring that it continues to run smoothly.

The chapter begins with space monitoring. This covers the utilities available to you that allow you to watch the database grow. The exercises focus on common methods of obtaining more room in the database. Next, the chapter focuses on enhancing the performance of the database. You will learn some of the GeoFrame recommended settings as well as some Oracle settings that can be implemented.

Keywords

Here is a list of the important keywords to learn from this chapter:

<table>
<thead>
<tr>
<th>Data Tablespace</th>
<th>TIMED_STATISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index Tablespace</td>
<td>utlstat</td>
</tr>
<tr>
<td>free_space</td>
<td>utlstat</td>
</tr>
<tr>
<td>space_check</td>
<td>init.ora</td>
</tr>
<tr>
<td>proj_delete</td>
<td>db_block_buffers</td>
</tr>
<tr>
<td>proj_dsl_listing.csh</td>
<td>db_block_size</td>
</tr>
<tr>
<td>cleandb</td>
<td>log_buffer</td>
</tr>
<tr>
<td>startup</td>
<td></td>
</tr>
<tr>
<td>mount</td>
<td></td>
</tr>
</tbody>
</table>
Exercise 16. Space Management; Adding Datafiles

The most common way of acquiring tablespace is to add to existing tablespaces, as opposed to creating new tablespaces. This primarily involves “registering” additional physical Unix datafiles with Oracle as part of an existing tablespace in your database.

How have you determined which tablespaces are running out of space?

1. Open a GeoFrame xterm and run the free_space utility. Don’t forget to include the Oracle account information (system/manager)

2. Are any table spaces running low?

3. How much continuous space is recommended for the creation of a new project? (Hint: GeoFrame Bookshelf; Oracle RunTime Installation Guide)

   . . . . .

   NOTE: The amount of space remaining as well as which projects are using the most tablespace can be monitored as gf_dba from the Tablespace... button in the Database Administration portion of the Project Manager.

For the purpose of simulating a real environment, imagine that the SYSTEM, TSDATA, and TSINDEX tablespaces are nearly full. Add additional space to each one.

4. Open a GeoFrame xterm or switch users to oraown in a regular xterm and source the Oracle_Runtime.csh file. Connect to the system account in Oracle using sqlplus

5. Add to the SYSTEM tablespace first

   SQL> alter tablespace SYSTEM add datafile

   '/apps/oraown/dbs/SYSTEM2gf40.dbf' size 25M;
   (or your choice)

   You should see a message stating the statement was processed or that the tablespace was altered. Repeat the process for TSDATA and TSINDEX.
6 Add to TSDATA:

```
SQL> alter tablespace TSDATA add datafile
'/apps/oraown/dbs/TSDATA2oragf38.dbf' size 50M;
(or your choice)
```

7 Add to TSINDEX

```
SQL> alter tablespace TSINDEX add datafile
'/apps/oraown/dbs/TSINDEX2oragf38.dbf' size 50M;
(or your choice)
```

8 Exit SQL*Plus

```
SQL> quit
```

9 Run free_space again to see your new space. You can also query for a new list of datafiles that will show the new files that you just added.

**DISCUSSION**: How many datafiles can be added to a single tablespace? How can you change that amount?

Adding datafiles to existing tablespaces is the most common, not to mention probably the easiest method of adding free space to your Oracle tablespace. The next exercise demonstrates a somewhat obvious way to obtain space, while demonstrating tablespace fragmentation as well.
Exercise 17. Space Management; Removing Projects

Another method of obtaining additional space within your tablespaces is to remove old or undesired projects. Once you have identified the projects that can be removed, there are different ways you can remove them. This exercise demonstrates the methods of doing that, while illustrating that the space obtained may not be continuous.

Understanding completely what has to be removed and what then has to be updated to properly clean a GeoFrame database of a project is important. You should have a better understanding of this after completing these exercises.

The ideal way to delete a project is from the Project Manager. This will clear all references to the project, including Charisma, IESX, and CPS data associated with it.

1. Determine the oldest project. This is the one we will delete first in order to demonstrate some tablespace fragmentation. Open a GeoFrame xterm.

   ```shell
   % sqlplus gf_dba/gf_dba
   SQL> select ACCOUNT_NAME, DATE_CREATED from FINDER_ACCOUNTS order by DATE_CREATED;
   ``

2. Choose a project that was created earlier than most. Make sure it's a project! Do not delete gf_sys, gf_dba, or your baseline account!

3. SQL> quit

4. In a GeoFrame xterm, run free_space and space_check to see the amount and numbers of continuous free space you have.

5. Run gf_users to verify that nobody is connected to the project that you are about to delete.

6. Set the ADI_DEBUG to 500 again and start GeoFrame from the GeoFrame xterm.

7. Switch to the DBA mode. This will allow you to delete projects without knowing the password.

8. Connect to the project that you are going to delete.

9. Select the Project Management tab and choose to Delete the project. After the project has been deleted, close the Project Manager in order to free up your GeoFrame xterm.
10 Run `gf_accounts` again to verify that the project was deleted.

**NOTE:** It will take Oracle a while to completely remove the project account. The project may still show up after running `gf_accounts` until all of the necessary updates have been made to the `gf_dba` tables.

11 Run `free_space` again. What happened to the Average Mbytes of free space? How many continuous blocks of free space do you now have over 1 Mbyte in size?

A scripted method of deleting projects is provided as well. The `proj_delete` executable can be used to remove GeoFrame projects only. Also, the DSLs will not be deleted. You will need to remove them in the Unix world, after running the `proj_delete` script (unless you specify an option to force the deletion of the DSLs, `-force`).

12 In a GeoFrame xterm, determine all of the DSLs for the project you want to delete by using the `proj_dsl_listing.csh` script:

```
% cd
% touch <your project name>_dsl_list.dat
% proj_dsl_listing.csh <your project name> <you project password>
```

13 In a GeoFrame xterm, use the `proj_delete` utility to delete a project:

```
% proj_delete <your project name> <your project password>
```

14 Run `gf_accounts` to verify that the project has been deleted.

15 Are the DSLs still there?

```
% ls -lrt
(Do this in the same directory you were in when you ran proj_dsl_listing.csh)
% more <your project name>_dsl_list.dat
% cd /<the path to your project DSLs>
(From the list immediately above)
```

16 Remove them using Unix commands.

17 Run `free_space` again from the GeoFrame xterm. Did you create another block of free space in your tablespaces?
If the previous two methods did not clear the Oracle database of your project, you can run the cleandb utility. This should be used as a last resort, after you have tried the other two. **Cleandb** will clear the Oracle portion of your project only. You will need to remove the DSLs manually, same as before.

18 In a GeoFrame xterm, determine all of the DSLs for the project that you want to delete by using the *proj_dsl_listing.csh* script

```bash
% cd
% touch <your project name>_dsl_list.dat
% proj_dsl_listing.csh <your project name> <your project password>
```

19 In a GeoFrame xterm, set the appropriate environment variables that **Cleandb** requires:

```bash
% setenv FGS_USERID gf_dba/gf_dba
% setenv SYSTEM_USERID system/manager
```

20 Run **gf_users** to verify that nobody is connected to that project.

21 Run the **Cleandb** executable

```bash
%cleandb <your project name>
```

22 Find and delete the DSLs for that project using the output from the *proj_dsl_listing.csh* script and the Unix “*rm*” command.

**DISCUSSION:** To entirely delete a project from GeoFrame, what has to be done? Why can’t you simply remove the project using the Oracle command **drop user**?

23 In order to have a project or two for the upcoming exercises, go ahead and create some projects. See exercise 12 for information on creating projects.
Exercise 18. Moving Tablespaces

If you installed Oracle and used the default locations for all of your database files, then they were all placed in the same location on one machine. This is a performance issue with regards to I/O. You may be required to move tablespaces for other reasons; for instance, a department wants to use a machine where a tablespace resided for another purpose, and you must get your datafiles off. However, you should be prepared to request to move and/or move your tablespaces to take advantage of new machines, disks, or other machines in the network to enhance the performance of your database.

Some of the Tablespaces are written to simultaneously during the normal operation of Oracle. At least one recommendation by Oracle is to have your index tablespace located apart from your data tablespace. Simulate moving your TSINDEX tablespace to a different disk. For the purpose of illustrating this, just move them one directory up, into the Oracle owners home directory.

The process of moving datafiles involves shutting the database down, physically moving the files on your machine, and then informing Oracle of the new location of those datafiles.

• • • • • •

NOTE: Prior to shutting the database down, locate all of the datafiles for the tablespace you intend to move. Forgetting to move one datafile and having your tablespace “spread” across different locations could introduce additional I/O issues.

1 In a GeoFrame xterm, and using the space_check utility, locate all of the datafiles that make up the TSINDEX tablespace

    \% space_check system/manager

2 Open a window and switch users to the oracle owner

3 Shutdown the database

4 Exit the Server Manager and move all of the necessary datafiles to their new location.
At this point if you tried to start the database, Oracle would not be able to find all of the necessary datafiles and the database would not open. You must update Oracle with the new datafile locations. This is done by starting the instance, but not opening the database.

5 Start **Server Manager**.

\% srmgrl

6 Start the instance and connect with the database, but don’t open the datafiles:

**SVRMGR**> connect internal

**SVRMGR**> startup mount

7 Update the datafile locations:

**SVRMGR**> alter database rename file

`<your_old_complete_path_here/yourTSDATA.dbf>`

to `'<your_new_complete_path_here/yourTSDATA.dbf>';`

**NOTE:** Be sure to move all of the datafiles for the TSINDEX tablespace that were revealed in step 1.

8 Shutdown and startup the Database, or

**SVRMGR**> alter database open;

If the database starts up with no errors, then you’ve successfully moved your Tablespace. You can verify it by running the space_check utility again.

You have successfully moved the location of a Tablespace. You will have a better understanding of the different stages of starting up the database from some additional exercises as well. If you have any questions about the stage of starting up a database at this point, be sure to ask the instructor.
Tuning the Shared Global Area of your Oracle Instance can be one of the most effective ways of increasing the performance of your database. It is most effective when you can classify your database into a main category: Decision Support, or Transaction Processing. Unfortunately, a GeoFrame database does not readily fall into one of those categories. Different applications in GeoFrame require different sets of configuration parameters to optimize performance, however, certain common settings are beneficial for most, if not all of the applications.

Although the most in-depth details regarding the performance statistics of your Oracle database are beyond the scope of a five-day course, it is good to know how to obtain the statistics. The following portion of the exercise shows one method, using a couple scripts that are supplied with Oracle.

1. Logon as your GeoFrame operator, and open an additional xterm in the second display. In that terminal window, switch users to the Oracle Administrator account (oraown) and source the `Oracle_Runtime.csh` file.

2. Start the Server Manager in the window that you are logged on as oraown and connect to the database.

   ```
   % svrmgrl
   SVRMGR> connect internal
   ```

3. Activate the TIMED_STATISTICS in the database:

   ```
   SVRMGR> alter system set TIMED_STATISTICS = TRUE;
   ```

4. Run the supplied Oracle utility that will begin the timing and recording of the statistics.

   ```
   SVRMGR> @$ORACLE_HOME/rdbms/admin/utlbstat
   ```

5. In the other display, start GeoFrame and create a project. This will create much activity in the database in order to obtain some statistics.

6. After the project has been created, go back to the oraown xterm and end the tracking of the database:

   ```
   SVRMGR> @$ORACLE_HOME/rdbms/admin/utlestat
   ```
This will create a file called report.txt in the current directory. View the file and look at the various statistics.

DISCUSSION: Look over the statistics. Does anything look peculiar? Given the nature of our datafile layout, are there some poor performance figures we expected?

Given the knowledge of how to obtain these statistics, there are many good Oracle references that can help you make sense of them and guide you in making changes to the database parameters, should you desire.

Once you have determined the changes you would like to make to your database, either through monitoring the performance, or to simply adhere to the recommended settings by GeoQuest, you must shutdown the database, stopping the instance, and edit the “init.ora” file for your instance. Then simply start it back up. The Oracle parameter file init<SID>.ora defines many important Oracle configuration parameters and system wide setup parameters. This exercise focuses on this file and it’s contents. Your “init.ora” file (for your SID) can be found in $ORACLE_HOME/dbs. The name of the parameter file for your instance will contain the SID you gave to your database.

In the xterm that you have switched users to oraown, exit the Server Manager.

Navigate to $ORACLE_HOME/dbs. List the contents of the directory and locate the init<SID>.ora file for your instance.

Using your favorite editor, begin editing the init<SID>.ora file to reflect the following recommendations for a GeoFrame Oracle instance using Oracle 8.1.6:

---

NOTE: The following settings most likely will NOT be the best settings for the Oracle instance at your shop. They are intended for use in this course to illustrate how these parameters are set.

- `db_block_buffers = 4000`
- `shared_pool_size = 50000000`
- `shared_pool_reserved_size = 5000000`
- `log_buffer = 262144`
- `sort_area_retained_size = 512000`
- `sort_area_size = 2048000`

All other parameters default to an acceptable setting.
---

Stop and Start the database.
12 Exit the Server Manager. Check the resources that Oracle is using with the \texttt{ipcs -b} command. Is Oracle using one continuous memory segment?

\textbf{DISCUSSION}: Why were you able to edit the init.ora file while the database was running? When is the init.ora file used? What other Oracle files are similar to this?

\textbf{Online Help}

The online help for the Oracle issues discussed in this chapter is located in two of the GeoFrame Bookshelf documents. Both are located in the GeoFrame System portion of the Bookshelf in the lower right hand corner of the menu. The Oracle Installation Guide contains some of the information, with the other being the Database Administration document.

In addition to the GeoFrame Bookshelf documents, you can obtain more information about any error messages you receive from Oracle by using the \texttt{oerr} utility. For example, if you receive an error message that contains the following:

\textbf{Error: ORA-02233}

you can obtain more information by typing:

\texttt{% oerr ora 2233}

at the Unix prompt, after you have set the proper path in the environment (sourced the Oracle\_Runtime.csh file.)

Performance tuning an Oracle database is at least a five-day course by itself. For more information regarding performance tuning, you can turn to many books on the subject from the Oracle Press, or one of many other publishers (such as O’Reilly.)
Summary

In this chapter you have learned how to monitor your database for many of the things that can slow your database down, or even prevent the normal use of, it as is the case when you run out of space.

You have also learned that the way GeoFrame uses Oracle prevents it from falling into one of the two main categories of databases; therefore, the best all around tuning parameters may be those recommended by GeoQuest. In addition to the SGA parameters however, you have also learned how to move Tablespaces in the event that it is required, or to take advantage of newly available equipment in order to implement some of Oracle’s recommendation regarding Tablespace locations.

The next chapter will move the focus towards maintaining adequate recovery procedures, as well as some of the steps you should take to prevent database corruption. This will include the various types and levels of backups and the proper security measures that should be employed.
Chapter 8
Oracle Administration; Backup & Recovery

Overview

Within the Oracle portion of the course, recalling the three categories mentioned in Chapter 7; with the first two being creation and maintenance of the database, this chapter marks the third and final category: Backup and Recovery of the database.

This chapter will cover all of the methods that exist for maintaining adequate backups of all of the data, both at the database level and the project level. Additionally, the chapter covers some of the basic security measures that can be practiced, minimizing the risk of database corruption.

At the end of this chapter you will have a good idea of which backup strategy will work best for your workplace.

Keywords

The following keywords are important in this chapter. Gain an understanding of these words:

<table>
<thead>
<tr>
<th>cold backup</th>
<th>Incremental Backup</th>
</tr>
</thead>
<tbody>
<tr>
<td>v$logfile</td>
<td>remote_gen_save.csh</td>
</tr>
<tr>
<td>v$controlfile</td>
<td>&lt;project&gt;_remote_save.csh</td>
</tr>
<tr>
<td>v$datafile</td>
<td>-nobulk</td>
</tr>
<tr>
<td>shutdown abort</td>
<td>gf_setup</td>
</tr>
<tr>
<td>GeoFrame Archive</td>
<td>proj_changepasswd</td>
</tr>
<tr>
<td>Fast Backup</td>
<td></td>
</tr>
</tbody>
</table>
Exercise 20. Cold Backups

The objective of this exercise is to become familiar with the Oracle cold backup and to learn how to perform one. Additionally, you will learn to recover the database from a cold backup.

The Oracle cold backup is a ‘point in time’ backup. It is a very simple type of backup to create and recover from. It is also the first step in beginning a hot backup. Since it is a point-in-time backup, it is very important that the database is shutdown, making sure that every Oracle file saved is from precisely the same point in time. If this is not the case, recovering from the backup will cause problems.

First, locate every physical Unix file that makes up the database. This should include all datafiles, logfiles, and controlfiles. If any of the “living” files in the database are omitted from the cold backup, you will not have a viable cold backup. This list will be a complete list of all of the files that need to be part of the backup.

1. Logon as oraown and using the Server Manager, connect to the database as user sys.
2. Identify all of the files that need to be included in the cold backup, and create a text file that will prove useful for using the tar command later:

   SVRMGR> spool $HOME/livefiles.txt

   SVRMGR> select name from v$datafile;

   SVRMGR> select member from v$logfile;

   SVRMGR> select name from v$controlfile;

   SVRMGR> spool off

3. Shutdown the database.
   The next couple steps involve using vi to edit the “livefiles.txt” file to make the tar command much easier. You may find it beneficial to open another terminal window for UNIX commands.
4 Use `vi` to fix `livefiles.txt`. While in the command mode, remove all of the lines that are obviously not some sort of database file (use the `vi` command `dd`). Do not delete any lines that are an obvious database file! Next, remove all of the spaces at the end of each line in the file. This is accomplished by issuing the following `ed` command in command mode:

```
:1,$s/ //g
```

note the single space between the first and second slashes.

5 In the command mode you will need to use `:wq!` to save and exit `vi`.

6 View the `livefiles.txt` file.

The `tar` command works well for backing up the files, because in one smooth command you can create a tarfile that includes all of the database files. Also one command will restore the files to their original locations on the Oracle server. You can certainly use any method you would like, including a simple Unix copy (`cp`).

7 tar all of the files together, creating a single tarfile in your home directory. Use the following command to use the contents of `livefiles.txt` as the arguments for the `tar` command:

```
% tar cvf ~/coldone.tar -I livefiles.txt
```

8 Verify that the tarfile contains all of the files listed in `livefiles.txt`.

```
% tar tf ~/coldone.tar | diff livefiles.txt -
```

```
• • • • • •
```

**NOTE:** Step 8 should return nothing if the operation was a success. As we have used it, the Unix `diff` command will only return the difference between the output of the `tar tf` command and the contents of `livefiles.txt`.

9 Note the size of your tarfile.

```
• • • • • •
```

**NOTE:** Using the Unix command `compress` you would be able to compress this file to a fraction of it’s current size, primarily because most of the Tablespaces at this point are basically empty.

10 Return to the window in which you have started the Server Manager and startup the database.

Your database is running again, and you have a successful cold backup. As you have seen, one of the disadvantages of obtaining a cold backup is the downtime associated with the database being shutdown.
Prior to recovering from a cold backup, you can illustrate the importance of all of the files by moving or removing one so that Oracle cannot find it.

**NOTE:** If you are using this exercise as a reference, you can skip to step 21. There is no point in practicing corruption, especially if you are working with your production database.

11 With the database running, navigate to $HOME/dbs in the window where you are not running Server Manager. Once there, remove dbsoragf40.dbs (where gf40 is the database name.)


13 Remove one of your controlfiles as well. Try to shutdown the database.

14 Issue `shutdown abort` at the Server Manager prompt, to force the stopping of the database.

15 Try to restart the database. Did it work? Check for the running instance.

16 Issue `shutdown abort` again.

17 Exit the Server Manager.

**DISCUSSION:** Remember what the instance does and the difference between the instance and the database.

Recovering the database is as simple as extracting the tarfile you created earlier. Because the full path to the files was specified during the creation of the tarfile, the files will return to their original locations regardless of where you keep your tarfile, or where you run the tar command from (as they will be located from the controlfiles that should have been included in your backup.)

18 Issue the following tar command to extract your files:

```
% tar xvf ~/coldone.tar
```

19 Try to startup the database. Did it work?

**DISCUSSION:** You have just restored your database to the point in time in which the cold backup was taken. Any work done since that time is lost, unless you have been running in archive log mode, thereby having a hot backup.
20 Delete the livefiles.txt file.

NOTE: This is an important step that will prevent you from having to administer this file. Anytime you create a file such as livefiles.txt that contains vital information used to obtain a viable cold backup (in the way that we used it) you should always delete the file. Otherwise you will run the risk of having an outdated file, thereby creating incomplete backups.

The last step in obtaining a complete cold backup is to account for the GeoFrame portion. This means you should have a backup of the DSLs from the same point in time as well. Technically, the Oracle database will function properly because you will have made sure that all of the Oracle files are from the same point in time. However, the problems associated with not capturing all of the DSLs will present itself at the project level when the pointers within the Oracle tables don’t properly reflect what’s in the DSLs.

21 Identify the locations of all of your DSLs. Do you remember at what level in the GeoFrame environment you had assigned them? Are they assigned for the entire environment, or for each version of GeoFrame you have installed?

22 Connect to your baseline account in Oracle and perform the following queries to create a list of the DSL locations that you need to backup:

   prompt> sqlplus gf4_0_2/gf4_0_2
   SQLPLUS> spool $HOME/all_dsls.txt
   SQLPLUS> select STORAGE_PATH from DISKR;
   SQLPLUS> spool off

23 Prepare the all_dsls.txt file just as you did the file for the Oracle datafiles with the vi editor. The contents of this file can be used to create a tar file of the DSLs.

DISCUSSION: Who should actually create the tar file? Are there some options you need to use in order to maintain ownership of all of the files in the DSLs? When extracting the tar file? What are the implications of not maintaining this ownership?
The exercise that you just completed captures all of the living files in the Oracle database, including the controlfiles. The following exercise illustrates how to rebuild controlfiles in the event they become corrupted.
Exercise 21. Securing your Controlfiles

The objective of this exercise is to learn to produce an SQL script that will allow you to rebuild your controlfiles, and to learn how to add controlfiles to your database for redundancy purposes. Remember that all of your controlfiles are mirrored copies of each other. It is possible for them to become corrupted from time to time, and it is then necessary to have a method to rebuild them. Oracle offers the ability to do this.

Controlfiles are one of the “living and breathing” files of your Oracle database. Therefore simply maintaining a copy of the controlfiles in a separate location would not be sufficient. This is because upon restoring your controlfile, it will be from a different point in time. This exercise will show you how to use Oracle to rebuild the controlfiles it needs.

1. Login to the Oracle server as oraown. Source the Oracle_Runtime.csh file, and start the Server Manager.

2. After connecting to the database, issue the following command. This will create a trace backup of your controlfile:

   \code{SVRMGR> alter database backup controlfile to trace;}

3. You should have received a response in return saying “statement processed.” The command that you issued produced a text file in $ORACLE_HOME/rdbms/log. It will have a suffix .trc and should be the most recent trace file there.

4. Open another window, source the Oracle_Runtime.csh file and view the file you just created.

5. Using vi, edit the file and remove the unnecessary information at the top of the file and any comments. The actual script begins with “STARTUP NOMOUNT” and finishes with “ALTER DATABASE OPEN;” Save the file as control.sql which has more meaning now. (:w control.sql) Exit vi.

   The next portion of this exercise will illustrate the importance of the controlfiles. We can simulate corrupted or missing controlfiles and try to start the database.

6. In the other window, where you should still have the Server Manager running, connect to your database and shut it down.
7 Back in the other window again, navigate to $HOME/dbs and locate your controlfiles. To simulate having corrupted controlfiles, rename cntrl<your_SID>.dbf to oldcntrl<your_SID>.dbf.

8 Next, in order for Oracle to find all controlfiles, although one is corrupt, create an empty file for Oracle to find, with the same name as the controlfile you renamed in the last step:

```bash
% touch cntrl<your_SID>.dbf
```

9 From the Server Manager, startup the database. What happened? You are in a situation such that you know what has happened, and probably have a good idea of how to fix it. For this exercise; however, we will assume all of our controlfiles are corrupted, and recreate them from the script we created earlier.

10 Shutdown the database.

11 How many controlfiles did you have? Locate them and remove them.

12 List the contents of $HOME/dbs to verify that they are gone.

13 In the window with Server Manager running, execute the SQL script control.sql:

```sql
SVRMGRL> @$ORACLE_HOME/rdbms/log/control.sql
```

14 You should have a running database now. Query for your controlfiles:

```sql
SVRMGRL> select name from v$controlfile;
```

15 How many controlfiles were returned?

16 In the other window, list the contents of the $HOME/dbs directory again. Does everything look good?

DISCUSSION: How did Oracle know how many controlfiles to create? And how did Oracle know what to call them?

The discussion above brings us to the next section of this exercise, which will illustrate how Oracle knows what controlfiles exist and where. You will also learn how to add controlfiles or move the existing controlfiles.

You know by now that Oracle uses the control files to maintain a list of the physical Unix datafiles that it will look for upon starting the database (among other things.) The question becomes: Where does Oracle keep track of what controlfiles must be present? The answer is in the init<SID>.ora file.

17 View the “init.ora” file for your instance. It should be located in $ORACLE_HOME/dbs. Locate the controlfiles near the bottom of the file in the Server Manager window.

18 Shutdown the database.
Moving or changing the number of controlfiles is very similar to moving or adding datafiles. It is very important to cleanly stop the database, make your changes, and inform Oracle of the changes. Then simply open the database.

19 With the database shutdown, edit your init<SID>.ora file to reflect the movement of one of your controlfiles one directory up (to simulate moving a control file.) Also, edit the file to show a new controlfile created in your home directory (to simulate adding another controlfile.)

20 Save the changes you have made, and exit the editor. Now move the controlfile and perform the appropriate copy as per the changes you have made to your “init.ora” file.

• • • • • •

NOTE: The UNIX copy is allowed here as a method of creating additional controlfiles because the database has been stopped and all of the files will be from the same point in time with respect to Oracle.

21 In the window with Server Manager running, startup the database.

22 Query for all of the controlfiles now. Does everything look to be in order?

The importance of controlfiles is not to be underestimated. These are small, but very crucial files that are constantly written to simultaneously. It is very important that the database is shutdown cleanly whenever you intend to manipulate these files. If for any reason you ruin or lose these files, the database is ruined. In order to recover, you would need to restore a cold backup, which may result in lost work.

A good practice is to save a copy of your controlfiles prior to attempting to move them, or when attempting to add new controlfiles. This will give you a good copy to fall back on if you make a mistake. Once the database is started, however, this copy should be discarded, as it won’t be of any use to Oracle.

Anytime datafiles are added to the database, or the location of datafiles is changed, Oracle will make significant changes to the controlfiles. In this situation, where major changes have been made to the database, it is important to regenerate a new script that will recreate the controlfiles as they would now need to be created. As with any backup, the script you have learned to create in this exercise needs to be kept up to date.

23 You can exit the Server Manager and logoff of your machine if you would like.

At this point in learning to maintain adequate recovery procedures for your Oracle database, you should have a viable cold backup of the entire database, and a script that will allow you to recreate your controlfiles. There are two items that remain: the TNS_ADMIN directory, and the init<SID>.ora file. In both cases it is just a simple matter of taking a copy of these files anytime. It does not matter whether or not the database is running because the files are not left open while the database is running. They are only periodically read a different times.
On that note, the `TNS_ADMIN` and `init.ora` files can usually be rebuilt from templates. This usually can suffice in order to get your database up and running again.

The methods of securing your database in the last 2 exercises are perhaps crude; however, they are very effective. Downtime is an obvious drawback to the methods described so far. The Oracle Warm and Hot backup methods are alternatives that require a bit more administration to maintain, and much more time to recover from.

You should have a thorough understanding of the workings of an Oracle Database at this point, and know what needs to be backed up. You can choose a backup method that works best for the needs of your shop.

The next exercise also covers backups; however, at the project level.
Exercise 22. GeoFrame Backups

Cold backup and the securing of your controlfiles and TNS_ADMIN directory are strictly Oracle methods of securing your database. There is nothing ‘GeoFrame specific’ with regards to the last two exercises apart from backing up the Data Storage Locations. It is important to realize that, and to therefore realize that an Oracle cold backup still leaves a lot of data unaccounted for, with respect to having a complete backup.

GeoFrame therefore offers different methods of backing up your projects that account for both the Oracle portion of a project, and the data stored in the DSLs. Although most of the utilities and scripts about to be introduced are for obtaining project level backups, they are included in this course, because realistically the responsibility of maintaining adequate backups, project level or not, often falls on the System Administrators. The objective of this exercise is to introduce the other methods of backing up projects and data that are available as part of the GeoFrame package, and to discuss what is happening in Oracle when using each of these backups. Furthermore, you will learn which types of backups are best for certain situations, for instance, moving a project versus simply having a backup for disaster recovery.

These methods of backing up are all done from either the Project Manager or a GeoFrame xterm.

1. Logon to your machine as your GeoFrame operator.
2. Open a GeoFrame xterm and set the ADI_DEBUG environment variable to 500.
3. Start GeoFrame by typing proman at the prompt.
4. Connect to one of your standalone projects. If you don’t have a standalone project, create one now, and connect to it.
5. Click on the Project Management tab and then choose Backup...
   Note the two types of backups available to you from the Project Manager. There are differences in the amount and type of data each will backup, as well as significant differences in what is required in order to recover from each type of backup, depending on some of the options you are able to choose for each.
7. Watch the GeoFrame xterm just to get a feel for the amount of things that are going on.
8 Create a **Fast Backup** of the same project. Select **Full Backup**. Also make this backup to a file. Note the naming convention that is used by GeoFrame to indicate that you are taking a Full Fast Backup.

9 Once again, watch the GeoFrame xterm to get a feel for the amount of activity.

10 Aside from the slight difference in the length of time it took to perform the fast backup, what other event did you notice during the fast backup?

11 Create another Fast Backup, this time select **Incremental Backup**. What is an Incremental Backup? What information are you asked to provide? Once again, note the naming convention GeoFrame uses to indicate the you are taking an Incremental Fast Backup.

12 Click **Cancel** to return to the **Project Manager** window.

13 Click **Delete** to remove the project from GeoFrame. In deleting the project from the Project Manager window, what is going to be removed that hopefully wouldn’t be part of a database crash?

• • • • •

**NOTE:** Remember that it will take a while for Oracle to remove the account when you delete a project. If in the next step you get an error during creating the project, the reason is most likely because the Oracle account hasn’t been dropped yet, and therefore the project name will exist.

14 Attempt to recover your project from the **Incremental Fast Backup**. Click **Recover...** from the **Project Manager** window and select the Incremental Fast Backup file you created. What message have you received in the console of the **Project Recover** window?

15 Follow the instructions in the console window, telling you to create the project from scratch.

16 Map the DSLs for each storage type by selecting the desired storage type from the tabs in the **Map Storage Disks** window. You can increase the width of the window to see all of the storage types. Map the storage device on the left (which are those from the backup) to the available storage device on the right (which are those granted to you by the GF_DBA.)

17 Select **OK** to close the **Map Storage Disks** window when you are finished, and click **OK** in the **Project Recover** window to proceed with the project recovery. Were you able to recover the project? Why do you think you were able to recover from an incremental backup starting with nothing?

**DISCUSSION:** Which backup is best? Discuss the benefits of the Fast Backup versus the project archive for different purposes, and vice versa.
GeoFrame offers some backup commands that can be run from a GeoFrame xterm. Some of these commands are actually executed upon using the backups in the Project Manager. Knowing how to use the commands is beneficial, as there maybe a type of backup that better fits your needs, or options you can provide that allow you to “tailor” the results.

18 In a GeoFrame xterm, navigate to $GF_PATH/bin.

19 List all of the files in this directory that contain “save” in their name.

20 View the contents of the files if you can. What is the difference between remote_gen_save.csh and remote_gen_save_batch.csh?

21 Change to your home directory (cd).

22 Choose one of your standalone projects and generate backup scripts for it. Generate a “<project>_save.csh” script, and a <project>_remote_save.csh

prompt> proj_gen_save_script.csh <your project name> <your project password>

prompt> remote_gen_save.csh <your project name> <your project password>

prompt> ls -lrt

23 View each of the scripts.

24 Run each script to obtain the backup that each produces. Keep an eye on the messages going by. Did you notice a difference between the two? Did you notice any resemblance to the backups that were made from the Project Manager?

DISCUSSION:

The scripts you just ran are essentially combined in the Project Manager Fast Backup. What are the differences between a .gfb file and a .tar file?

The next portion of the exercise will allow you to quickly create a script that can be used to backup the Oracle portion of every project using the GF4_0_2 catalog. This will prove useful for whenever you need to backup multiple projects often. The remote_gen_save_batch.csh script is used here, but the proj_gen_save_batch.csh script can be used as well.

25 In a GeoFrame xterm, type remote_gen_save_batch.csh by itself to learn the usage of the script.

26 You can see that you have the option to not backup the DSLs. Run the script so that you generate a backup script for every project that is using the GF4_0_2 catalog. Use the nobulk option so DSLs are not saved.
27 List the contents of the current directory in reverse time order to verify that the scripts are there.

28 Now list only the scripts you just generated, and send the output to a file called `backupall`:

```bash
prompt> ls *remote_save.csh > backupall
```

29 Using vi, edit the file `backupall`. First insert a line at the top that contains `#!/bin/csh`. Then at the end of each line you will need to append the password of the project that will be backed up by the script on that line, followed by the storage location of the backup, each separated with a space.

**DISCUSSION:**
Discuss using a tape drive with the above script.

30 Using the `chmod` command, make your script executable. Also, while you are doing this, make the file only readable by you (`chmod 700`). This is because the passwords to all of the projects are visible in the file.

31 Run your script and check for the backups. What kind of files are they?

• • • • • •

**NOTE:** Keep in mind what it would take to update your script for new projects – or additional DSLs for that matter.

This is just one example of a script that can perform one type of backup of all of your projects that use a common catalog to disk. There are many types of Oracle backups you can perform as well. And, of course, you can design your script to perform whatever you would like. No single script or type of backup is good for all situations. You should be familiar with the GeoFrame backups at this point, as well as have a better awareness for the possibilities within Oracle in order to satisfy your backing up needs.
Exercise 23. Database Security

Maintaining adequate security in your database is the first step to avoiding corruption. So far in creating the GeoFrame environment many default passwords have been used, of which essentially anybody who has ever installed GeoFrame or Oracle is aware.

Up to this point in the installation there has been much care in setting up the proper ownership and permissions with regards to the file structure on the machines. There are some Oracle system passwords that need to be changed now, as well as other passwords in Oracle that need special attention because of their relationship to the GeoFrame environment.

The need to change from default passwords is obvious; however, after this exercise you will have a better understanding of exactly why and how certain passwords should be changed, and the implications of leaving them unchanged.

The first passwords you should change are those for the Oracle Administrator accounts: sys and system. Remember the defaults upon installing Oracle and building our database were manager and change_on_install respectively. Anybody in the world who has ever installed Oracle knows these passwords. These passwords can be changed using the standard Oracle method of changing Oracle account passwords because these accounts technically have nothing to do with GeoFrame. They are strictly Oracle accounts required for any Oracle database.

1 Open a GeoFrame xterm and start SQL*Plus as Oracle user system.

          prompt> sqlplus system/manager

2 To change passwords for Oracle accounts, the command is as follows, using the sys account as an example:

          SQL> alter user sys identified by newpasswd;
You should see a message returned: User altered.

3 Change the password for system in the same way.

   ● ● ● ● ● ●

   NOTE: Now that the passwords for sys and system have been changed, you would have to supply these new passwords whenever installing a GeoQuest product that uses Oracle. The installtool always uses manager and
change_on_install for the default passwords for Oracle users system and sys respectively. If these are NOT changed when installing, the installation procedures will fail.

DISCUSSION:
Even though the Oracle sys account has had its password changed, who will still be able to connect as sys (connect internal) without ever needing to know the password for the Oracle sys account? How is the authentication performed?

As mentioned above, most of the Unix ownership and permissions have been taken care of and is protected through the passwords for those accounts. Changing those passwords is simply done as per regular security policies and guidelines and by using normal Unix methods (passwd). The Oracle sys and system passwords have been changed as well, thereby securing the administrative accounts in our database. The rest of the passwords we will change are Oracle accounts as well; however, because of the additional GeoFrame issues, they cannot be changed using the standard Oracle method.

The Oracle accounts for GeoFrame, gf_sys and gf_dba have their passwords encrypted in a table called FINDER_ACCOUNTS owned by gf_dba. If you use the standard Oracle method of changing the passwords for these accounts, then they will no longer perform their function in GeoFrame. They will remain valid Oracle accounts; however, GeoFrame will not be able to utilize them.

DISCUSSION:
What are the privileges associated with the gf_sys and gf_dba accounts? Why should the passwords for these accounts be changed?

To change the passwords for these accounts, you must use the gf_setup utility. This is most easily run from a GeoFrame xterm.

4 Open a GeoFrame xterm.

5 Set some system environment variables that are required by the gf_setup utility:

   > setenv SYSTEM_USERID system/manager
   (or your new system password)

   > setenv FGS_USERID gf_dba/gf_dba

   > setenv PROJSYS_USERID gf_sys/pepsi
   (where “pepsi” is the new desired password for the gf_sys account.)
6 Type `gf_setup` at the prompt by itself to change the password for `gf_sys`.

```
> gf_setup
```

Changing the `gf_dba` account password in Oracle is very similar, except the `gf_setup` utility is run with the `-f` option.

7 Set and update the appropriate environment variables again:

```
> setenv SYSTEM_USERID system/manager
(or your new system password)

> setenv FGS_USERID gf_dba/cola
(where “cola” is the new desired password for the `gf_dba` account.)

> setenv PROJSYS_USERID gf_sys/pepsi
(where “pepsi” is the current `gf_sys` password.)
```

8 Type `gf_setup -f` at the prompt to change the password for `gf_dba`.

```
> gf_setup -f
```

The `gf_public` account in Oracle remains as one of the GeoFrame Oracle accounts that cannot have its password changed using conventional methods in Oracle alone. It would require the GeoFrame utility as well. However, there are additional concerns regarding the `gf_public` account and how it is utilized as part of the everyday operation of GeoFrame.

9 First, to illustrate the lack of a real need for security regarding the `gf_public` account, connect to the `gf_public` account using SQL*Plus from a GeoFrame xterm and view all of `gf_public`’s objects:

```
> sqlplus gf_public/gf_public
SQL> select * from cat;
```

10 What type of Oracle objects does `gf_public` own?

**DISCUSSION:**

How do you think you would have to change the password for `gf_public`? Do you remember if you were given the option to select a password for the `gf_public` account when you installed GeoFrame?

11 Next, to illustrate why the `gf_public` Oracle account password can not be changed, navigate to `GF_PATH/bin` in a GeoFrame xterm.
12 View the ASCII strings of the gf_users utility:

> strings gf_users

13 Identify the "gf_public/gf_public" line approximately four lines up from the end. What does this indicate? Discuss this with the class and the instructor.

The previous steps should have illustrated that the gf_public account only consists of views of tables that exist in other Oracle accounts. Therefore, the level of security required is minimal, and the account can be easily recreated anytime. Furthermore, the "gf_public/gf_public" Oracle logon is written into many of the GeoFrame utilities. Changing the gf_public password using any method will have undesirable results.

The only Oracle accounts that remain are the project accounts. These too need to have their passwords changed using methods that will account for the GeoFrame issues. If you change the password of your project account by simply using the conventional Oracle method, you will render your project unusable. As with gf_sys and gf_dba, it will still be a valid account in Oracle, and you will be able to access it using SQL, however, you will not be able to connect to it in GeoFrame.

The GeoFrame utility `proj_changepasswd` is used to change the password for a GeoFrame project. It can be performed by the project owner, or by the GeoFrame DBA (gf_dba) if the current password is not known. You must know the gf_dba password to do this however.

14 In a GeoFrame xterm, set the GeoFrame DBA password in the environment:

> setenv GFDBA_PASSWORD gf_dba

(or your gf_dba password)

15 Type `proj_changepasswd` at the prompt by itself to obtain the help message.

16 Change the password of one of your projects using `proj_changepasswd`. 


Online Help

The online help documents for this chapter are the **Oracle Run Time Installation Guide** and the **Database Administration** document, both located in the **GeoFrame System** portion of the GeoFrame Bookshelf.

As much of the chapter is purely Oracle Administration, you will be able to find much assistance in the Oracle DBA book of your choice.

Summary

In this exercise you have learned how to completely secure your database, and to take the necessary steps to prevent corruption.

Most importantly, perhaps, is that you have learned the element of the database and are aware of the significance of securing all of them, and taking the necessary precautions to prevent unauthorized users from corrupting the database.

The next chapter contains a mixture of exercises that cover a wide range of remaining GeoFrame topics, such as manually configuring the buttons in Geonet, to the Ctree Server.
Chapter 9
Other GeoFrame Issues

Overview

This chapter contains three exercises that cover some of the remaining issues that may be a part of the GeoFrame System Administrators responsibilities.

The first exercise covers the CTREE Server, including the important configuration parameters and how to configure a site CTREE Server. The next exercise introduces the workings of Geonet in greater detail and covers the methods of manually adding site specific utilities or commands to the Geonet window. The last exercise covers GeoFrame plotting. Assuming properly installed and running daemons (usually Zeh or SDI), the exercise will introduce the GeoFrame utilities that allow the user to efficiently configure GeoFrame to use them.

Keywords

The following keywords are significant in this chapter. After completing the exercises you should have a better awareness of what the keywords mean and their significance:

SITE: keyword/value pair
gstab
hcsetup
hardcopy
SiteHardcopy.hcd
PersonalHardcopy.hcd
spooler
server
printer
Exercise 24. CTREE Server

This exercise will cover some of the issues regarding the CTREE Server. The main objective of the exercise is to become more familiar with the CTREE Server and the locations and values for some of the important CTREE Server configuration parameters. Through a combination of classroom demonstrations and this practical exercise you will gain a better understanding of the CTREE Server.

There are two locations where the variable BULK_SERVER_ADDRESS is used. The first is an environment variable that GeoFrame will use to determine the nature of your CTREE server configuration, and the second is a column in a table called project in the project account in Oracle. This value in the Oracle table will specify the location of the CTREE Server daemon for GeoFrame if it is running (or where it should be running.) It is useful to know where these values are stored and how to view them. The first portion of this exercise will illustrate this.

1. Logon to your machine as your GeoFrame Operator.
2. Open a GeoFrame xterm.
3. Check for the environment Variable BULK_SERVER_ADDRESS.
   
   prompt> echo $BULK_SERVER_ADDRESS

   The value of the BULK_SERVER_ADDRESS should be what was specified during the installation of GeoFrame. Where is the value for the environment variable stored?

4. In the GeoFrame xterm, change directories to $GF_PATH, and look at the GeoFrame_Runtime.csh file.

   Information for GeoFrame about the CTREE Server configuration is stored here. You can refer to the GeoFrame Installation Guide in the GeoFrame Bookshelf for some of the other options available here as well as the required information for each type of configuration.

   The value for the BULK_SERVER_ADDRESS in the project table must also reflect the configuration of the CTREE Server.
You can check the value of the BULK_SERVER_ADDRESS in Oracle using SQL to illustrate the location within the project’s Oracle account. There is a utility provided which will be covered later in the exercise that is much easier than using SQL.

5. In the GeoFrame xterm, Start SQL*Plus and connect to one of the projects you have created.

`prompt> sqlplus <project>/<password>`

6. Describe the `project` table.

`SQL> desc project`

7. Use the following query to view the value of the BULK_SERVERADDRESS:

`SQL> select BULK_SERVERADDRESS from project;`

8. Feel free to view some of the other project parameters that are located in the project table.

If you chose "LOCAL:" when you installed GeoFrame, the BULK_SERVER_ADDRESS column in Oracle should be null, and the BULK_SERVER_ADDRESS environment variable should have a value of `LOCAL:`. In this case the CTREE Server daemon should start as needed and update the `project` table as needed to allow other users to connect to the same project and use the Bulk Data.
9 From the Geonet window, start GeoFrame, connect to a project (not a shared one), and start either IESX or Geology Office (or any application requiring the CTREE Server.) Watch the GeoFrame Application Manager window. Did you see the CTREE daemon start? You should see something similar to the following:

![GeoFrame Application Manager](image)

10 Check the BULK_SERVER_ADDRESS value in Oracle now. If you still have SQL*Plus open from the last query, exit SQL.

11 From the GeoFrame xterm type: \texttt{proj\_admin} at the prompt, to start the menu driven utility that allows you to check and update the CTREE Server and Super Server parameters. Connect to the same project you connected to in GeoFrame.

\begin{itemize}
\item \textbf{NOTE:} proj\_admin requires the Oracle account password. This is necessary in order to perform the Oracle “update” that is used to null or change the value of the BULK_SERVER_ADDRESS in the project table. This means that only the project owner should be able to manipulate the Ctree Server parameters for the project, including nulling of the BULK_SERVER_ADDRESS.
\end{itemize}

12 Select Bulk Server Operations.

13 Select 1 to view the BULK_SERVER_ADDRESS.
14 Why would you need to Nullify the BULK_SERVER_ADDRESS? Why would you need to change it? Leave the proj_admin utility running.

15 Close the GeoFrame Application that you started, and exit GeoFrame.

16 Check the BULK_SERVER_ADDRESS again using the proj_admin utility that is running in the GeoFrame xterm. Is it null? If not, wait a minute or two and check again.

17 Open another GeoFrame xterm and run `gf_users`. What connections exist to the project?

**DISCUSSION:** Which connections to the project must be terminated before the BULK_SERVER_ADDRESS will be nulled? What should you always check before clearing the BULK_SERVER_ADDRESS manually?

18 Quit the proj_admin utility and close one of the GeoFrame xterms.
You should have a thorough understanding of the parameters that need to be set in order for the CTREE Server to start, and where the values for the parameters are located. The rest of the exercise will illustrate how to configure a site CTREE Server.

19 Did you select to configure the CTREE boot-time start/stop scripts? Check in `/etc/init.d` for `ctree_startup.sh`. If you didn’t, you may need to obtain a copy from somebody.

20 View the `ctree_startup.sh` file. What are the permissions on the script? Is GF_PATH correct? What about TWO_TASK? Who is the CTREE_USER?

• • • • • •

**NOTE:** The CTREE_USER needs to have write permissions to CTREE_HOME. Although the permissions are set so everybody can execute the script, you will be switching users in order to execute the daemon; therefore, you will need to know the password of the UNIX account named as the CTREE_USER.

21 What is the CTREE_NAME set to? This one is important, as it will identify the daemon we want to connect to. Write it down here: ____________________________

22 Was the ctree daemon started automatically? Check that, using the Unix ps command:

```
% ps -ef | grep ctsrvr
```

23 If there is no other CTREE Server daemon running, run the `ctree_startup.sh` script with the start argument.

24 Watch the screen upon starting the daemon. What was the Server Name? What is the port number that was used?

25 Check for the running daemon again.
Now that you have a running CTREE daemon, it is necessary to configure your GeoFrame project to use it. For existing projects, it is only necessary to update the BULK_SERVER_ADDRESS value in the project table within the project's Oracle account.

26 In a GeoFrame xterm, run the proj_admin utility and connect to the project that you would like to use the site CTREE Server.

27 Choose Bulk Server Operations and select Change Bulk_Server_Address (option 3.)

28 The correct syntax for the BULK_SERVER_ADDRESS is as follows. Enter it according to the values you have recorded:

\[
\text{site:<CTREE_NAME>@<hostname>.<domain.name>}
\]

\[\text{TIP: You can obtain your hostname by opening an xterm and typing hostname at the prompt. Likewise, obtain your domain name by typing domainname at the prompt.}\]

29 Quit proj_admin and start GeoFrame. Connect to the project you just updated to use the site CTREE Server and run IESX.

30 Did everything start alright? How can you tell? Did you see the Application Manager try to start a CTREE daemon again? Check for another CTREE daemon using ps.

**DISCUSSION:** What would be the value of the BULK_SERVER_ADDRESS in the project table in the Oracle project account if you were to create a new project right now? Would the new project use the site CTREE Server daemon you have just configured? Why?

You can demonstrate the above discussion by creating a new project and then connecting to it and starting IESX for instance. Then check for the value of the BULK_SERVER_ADDRESS using proj_admin. Furthermore, after you have started IESX, you can check for the additional CTREE daemon process running on your machine.
Exercise 25. Manually Configuring Geonet

It is possible to manually configure Geonet to have buttons or menus that are tailored to your site or even personal requirements. The objective of this exercise is to learn how to do this, as well as to learn more about the configuration files that Geonet uses.

1. Logon to your machine as the GeoFrame Operator.

2. If you’ve configured this Unix account to use Geonet already, Geonet should have started and perhaps an xterm. If an xterm is not already open, open one.

3. View the .cshrc file located in the home directory of the user.

4. Navigate to the directory indicated by GN_dir_sun5. This should be the same as GN_DIR from the installation of GeoFrame.

5. List the contents of the directory. Change to the config directory.

6. List the contents of the config directory.
   The config directory contains the files that determine the appearance of the Geonet window. The buttons, menus, and sub-menus you see in the Geonet window are all referenced in a file called gstab. Also in the gstab file is information about what each selection will do, or if it will reference a .db file.


8. What permissions are set on the gstab file?

9. Logout of your machine and log back in as gfown.

10. Create a little script that will just open a terminal. Use vi to create the following file:

    #!/bin/csh

    /usr/dt/bin/dtterm

11. Change the permissions to make the file executable by everybody in the gqs group.
The following part of the exercise will show one method of adding items to the Geonet window. By editing the Geonet Profile in GN_DIR/config you are editing the “master” g stab file. If you have enabled the “GN_USER_DIR” functionality in Geonet, the operators who are already configured to use Geonet won’t notice any changes made here without importing the master profile again.

12 Launch the Geonet Profile editor:

% setenv GN_DIR /apps/gfown/gn5c02_00/solaris

% /apps/gfown/gn5c02_00/solaris/bin/pedit

The Profile Editor window looks like this:

13 Click Edit > Add to open the Application Properties Dialog window so you can configure Geonet.

14 Add the script that you just created to the GeoFrame 4.0 Set in the Applications:

- Set the type to Application
- enter GeoFrame 4.0 in the Set area
- Choose a Label and a Service name
- Select Script File for the Launch Option
- Click Script File... to select your script
- Click OK.
15 Next, add the command line that will open a new terminal to the list of Tools in the Geonet configuration:

- Click Edit > Add
- Set the Type to Tool
- Choose a Label and a Service name
- Select Command for the Launch Option
- Enter /usr/dt/bin/dtterm for the Command String and Confirm String
- Click OK

16 Finally, click Reconfigure Geonet to make the changes to the gstab file. Choose Yes.

17 Close the Profile Editor and Logout of the machine. Login as your GeoFrame Operator.

If you are using the GN_USER_DIR functionality, you may not see the additional buttons yet. If you aren’t using it, then all Geonet operators are using the master gstab file, and the changes will be evident, and you can use the new buttons. The next steps illustrate how to import the master profile to the account with their own gstab file.

18 Type pedit in a terminal window to launch the Profile Editor.

19 Click File > Import Profile... to select a master profile. Navigate to the gstab file located in GN_DIR/config and select it. You should begin receiving questions about duplicate services. Select Yes to update each one, or No to only get the 2 new ones.

20 Click Reconfigure Geonet to make the changes to your gstab file.

21 Exit the Profile Editor, then stop and start Geonet. Did it work?

22 If you would like you can view your gstab file again to see the changes that were made. How difficult would it be to simply edit the gstab file to add a button?

DISCUSSION: You have seen how to add buttons and menus that will run a command or a script. The other Launch Option was a db Resource File. Look at the geoframe40.db Resource File to gain an understanding of the amount of arguments used to launch some of the GeoFrame 4.0 Applications.
Exercise 26. Plotting Configuration

The objective of this exercise is to become familiar with the plotting process and to learn to configure GeoFrame to utilize the plotting software that has been installed and is running at your jobsite.

Regardless of whether you are using Zeh or SDI plotting software, the overall plotting procedure and the general chain of events is the same with a few slight differences. For both there are some required parameters ranging from the obvious - like the name of the plotter, to the name of the plot server running the vendor software. Configuring GeoFrame for plotting, first involves running the `hcsetup` utility and defining an “alias” that specifies scripts, daemons and locations which can later be referenced by the user from each application.

For the purpose of configuring the plotting environment use the following values for the parameters that will be required in the `hcsetup` utility (you will simulate a working Zeh installation during this exercise):

- **Script File**: `/bin/zpsprint`
- **Vendor**: `Zeh`
- **Metafile directory**: `/apps/gfown/zps/wrk`
- **ControlFile Directory**: `/apps/gfown/zps/daem`

The remaining variables will take values based on the following scenario: Zeh is installed on your machine and the scheduler daemon is running on your machine. The access method for moving the metafile will be `local` since the Metafile directory will reside locally on your machine. The host serving the plotter can be your machine as well. You can determine the values of any remaining variables in this exercise.

All of this information will be kept in a file called a **hardcopy definition file** which will have a suffix `.hcd`. There is typically one created for the site, `SiteHardcopy.hcd`, and optionally one for the operators personal use, `PersonalHardcopy.hcd`. You are however able to create as many of these files as you would like. Running the `hcsetup` utility allows you to create and/or edit these files.
1 Logon to your machine as your GeoFrame operator and open a GeoFrame xterm.

2 Type `hcsetup` at the prompt to start the hardcopy setup utility. You should see a window open that looks like the following:

3 Click on **File** in the Hardcopy Setup window to view the options that are available to you.

4 Select **New Personal Hardcopy Definitions File** to create a new Personal Hardcopy Definition file.
   In the **Spoolers** section of the utility you will first create a name for the Spooler that you will continue to reference during the exercise. The spooler is defined by the script file that will be run upon submitting a plot job along with the Spooler options and the name of the Vendor.

5 If you know of some options that you would like to use with the spooler it might be easiest to begin with the **Spooler Options** tab, as seen in the figure above. Select a name for the Option List and begin inputting keyword/value pairs using the **New Keyword/Value Pair** button and the **Add/Replace** button. The keywords and values that are available are dependant on the vendor (Zeh or SDI). An example of a keyword/value pair for Zeh looks like this:

```
cutlines no
transparent yes
```
Or, if you have a list of Spooler options created you can specify that in the Spoolers section.

**NOTE:** If a keyword has no meaning to the particular vendor software then the keyword/value pair will be skipped. You can refer to the ZEH or SDI user guides for complete lists of available keywords and value options.

6 Click on the Spoolers tab. You should see a dialog similar to the one below:

![Spoolers tab dialog](image)

7 Select a name for the spooler. This name will be referenced in the next portion of the exercise and will define the script file that will be run upon submitting a plot job as well as any plotting options you may specify here.

8 Enter the Script File from above and your option list if you created one.

9 Select ZEH for the vendor

10 Click Add/Replace to apply your entries.
11 Next, click on the **Servers** tab. You should see a dialog similar to the one below.

![Server Configuration Dialog](image)

12 Select a name for your server.

• • • • • •

**NOTE:** This is another alias that is defined by the actual hostname of the machine that will spool your plot job as well as the method by which it will transfer the metafile (CGM file usually) and by a spooler that was defined in the **Spoolers** window. Additionally, the locations for the metafile and ticket file directories are given.

13 Enter your host name.

14 Choose **Local** for your access method of transferring the metafile to the plot server.

15 Choose a spooler that has been previously defined.

16 Enter the Metafile Directory and ControlFile Directory from above.

17 Click on **Add/Replace** to apply your entries.
18 Click the **Printers** tab to define a printer or plotter. You should see a dialog similar to the one below:

![Printers dialog](image)

19 Enter the name of your printer in the **Printer Name** field.

**NOTE:** *This is the name that will be visible from the GeoFrame applications and the hardcopy utility which can be chosen by the operator to submit a plot job. By choosing this name when plotting, the operator will be specifying all of the parameters you have configured so far, as well as those in this dialog window.*

20 Choose a server for the plotter from the list of configured servers.

21 In the **Device Name** field should be the actual name of the printer or plotter that can be resolved from the plot server selected above.

22 Enter the width of the paper in the plotter.

23 Enter the resolution. Use **300** here. This will be used by some applications, but not all, to tailor the graphics metafile for this resolution.

24 Choose **CGM** for the file type. This is the graphics metafile type preferred by the software that prints to this printer.
25 Click **Add/Replace** to apply your changes.

26 Click **File > Save As...** to specify a name for your new hardcopy definitions file. Enter the name you wish to save the file as or select **PersonalHardcopy.hcd** to save it as that.

27 Click **File > Exit** to close the hcsetup utility.
   
   You have just configured a plotter for use by GeoFrame applications or the hardcopy utility. The operator needs only to specify the plotter name to submit a hardcopy job. You can easily test to see that your plotter is configured by starting the hardcopy utility.

28 Type **hardcopy** at the GeoFrame xterm prompt to start the **Hardcopy Spooler Utility**.

29 Move down to **Printer** and try to select the printer you had just configured. Is it there?

**DISCUSSION:** Who needs to build the **SiteHardcopy.hcd** file? Where should it reside?
Online Help

Online Help for the CTREE Server can be found in the GeoFrame Installation Guide located in the GeoFrame System portion of the GeoFrame Bookshelf. Additional assistance, including an administrator’s guide can be obtained at the Faircom Web site: http://www.faircom.com/support

The Geonet Users Guide can be opened from the Menu bar in the Geonet window under Util > doc browser. The Geonet installation guide is there as well.

Online assistance is available for plotting in various places.

For help with configuring GeoFrame to use an existing ZEH or SDI installation, you can turn to the GeoFrame Bookshelf under the Utilities portion.

Additionally, there is help on the web from each of the main vendors:

For ZEH: http://www.zeh.com

For SDI: http://www.sysdev.com

Summary

This chapter covered various issues that are relevant to the duties of a GeoFrame System Administrator.

Remember that your first line of support is the GeoFrame Bookshelf. Browse the Bookshelf documents to see the complete library of documents available to you as a GeoFrame System Administrator. From Oracle Administration to GeoFrame Plotting, it’s all there.
Appendices A, B, & C are taken from the GeoFrame Bookshelf Help Documents which would normally be referenced as a primary means of obtaining assistance.

Given the nature of this GeoFrame System Administration course, it is recognized that these online documents would not be available, as GeoFrame would not be installed on the system.

These appendices are also only a portion of the complete installation guides for Geonet, Oracle, and GeoFrame. This portion has been provided for assistance in getting GeoFrame installed, so that the complete guides can be referenced from the GeoFrame bookshelf.

Furthermore, the values for many of the parameters in the following documents are examples and should not be used as part of this course.
Appendix A; Geonet Installation

This Appendix provides general information on installing Geonet using the GeoQuest Installation Tool. Refer to the general document on the GeoQuest Installation Tool for screen displays and general details about each phase of the installation.

Provided on the Utility Kit 2.0 CD are Solaris and IRIX versions of these applications for Geonet;

— Geonet 5.2 and GeoShare launcher
— FrameViewer version 5.5
— FlexLM version 7.2e
— Browse and Edit RP66BE version 2.1
— Adobe Acrobat Reader version 4.05 for UNIX
— GeoShare Cartographic Reprojection Tool, GCRT, version 1.0

Hardware and Operating System Requirements

Solaris 2.6, 8 - IRIX 6.5.x

230 Mb of disk space for a full installation
Getting Started

Geonet is provided as part of a general GeoQuest Utility Kit 2.0 CD that also includes Oracle and other common applications. Following is a step-by-step instruction of the installation.

Insert the Utilities CD-ROM in CDROM drive. This will normally be mounted automatically. If problems are experienced with mounting the CD, see your local system administration for assistance.

Login to the account that will be the administrator of Geonet.

Use the cd command to go to the top directory of the CD-ROM and perform the following command:

```
unix> ./installtool
```

A log file called ~/.InstallLog_time_date will be created.

During the Geonet 5.2 installation, the installtool creates a directory gn5c02_00 under the selected location. Platform-dependent Geonet files will be installed under subdirectories gn5c02_00/solaris and gn5c02_00/irix so that when you install Geonet on a Solaris machine, the binaries for IRIX are also included.

An environment variable called GN_DIR will be defined, pointing to either one of those subdirectories, depending on the user’s machine platform.

The name GN_DIR will refer to either <path to>/gn5c02_00/solaris or <path to>/gn5c02_00/irix in this document.
Installation Phases

Welcome Phase

1  Click **Next** to continue.

Select Components to Install Phase

2  Turn on the Geonet 5.2 button and then click the arrow to reveal which components are selected.

   ![NOTE: Oracle may also be installed at this time, depending on whether it will be administered by the same account as Geonet. Otherwise it should be installed separately by re-starting the Installation Tool at a later time.]

3  Select the following individual components, based on your site needs:

   — **Geonet 5.2** - requires 15 Mbytes. This installs Geonet and GeoShare launcher.
   
   — **FrameViewer 5.5** - requires 85 Mbytes
   
   — **FlexLM 7.2** - requires 5 Mbytes
   
   — **Acrobat** - requires 41 Mbytes (optional) The Acrobat installation kit will be copied to $GN_DIR/acrobat, where GN_DIR is defined as described in previous section. You will have to manually complete the installation when the installtool finishes by running the script $GN_DIR/acrobat/SSOLRS.install/INSTALL. Check the ReadMe file in that directory, which has instructions on how to set a Web Browser to launch the Acrobat Reader to view pdf files directly. It is recommended to install Acrobat Reader under $GN_DIR/acrobat/.

   — **RP66BE 2.1** - requires 10 Mbytes (optional - if you plan to use the RP66 Browser and editor version 2.1. **Note: Requires a license**)  

   — **GCRT 1.0** - requires 70 Mbytes (optional - if you are going to use the GeoShare Cartographic Reprojection Tool, version 1.0. **Note: It requires a license also**).

4  Click **Next** to continue.
Select Locations to Install Components Phase

5 Select a disk location to install Geonet (the default will be at the top level of the account that started the installation). Use the Browse button to bring up the Browse Dialog to select a different location.

6 Select Next to continue.

Installation Steps Phase

7 Select the following installation step:
   — Register the Utilities Documents with the DocBrowser Menu. This step installs the Geonet and Oracle Installation Guides using the Installtool, and adds these buttons to the Doc Browser, available from the Util Pulldown Menu in Geonet:
   — Geonet Installation using the Installtool
   — Oracle Installation using the Installtool

8 Select Next to continue.

Installation Parameters Phase

9 If you have an existing version of FrameViewer 5.5 which you wish to use, toggle the button Use existing FrameViewer? to Yes and enter the path to the existing viewer/directory; otherwise the default value is No and you can leave the path to existing viewer empty. Then Frameviewer will be installed under $GN_DIR/viewer/.

10 If you selected to install the RP66be21, you have to enter the full path to a valid Flexlm license file which contains the required feature for this application. This path should be available to the users. The answer is used to define LM_LICENSE_FILE in the script file $GN_DIR/rp66be21/<platform>/rp66be.csh. You can also enter this value as <port>@<host>. In this case, the installtool gives a warning message about “path not found”, although the value is correctly registered; you can ignore that message and select Yes to continue.

11 Select Next to continue.

Ready to Extract Files Phase

12 Select Next to continue.
Extracting Files Phase

A directory gn5c02_00 is created in the selected location. The progress bar moves forward as the files are extracted under that directory.

13 Select Next to continue.

Ready to Execute Procedures Phase

14 Select Next to continue.

Executing Procedures Phase

The procedures are executed depending on what was selected to install.

15 Select Next to continue.

Ready to Validate Files

16 Select Next to continue.

Validating Files

The installed archives are checked for any inconsistencies.

Installation is Complete

17 Select Quit to exit and save the recover file.

Setting Up the User’s Geonet Environment

Once Geonet 5.2 is installed, each user’s environment must be set up in order to run Geonet. The following sections show how to set up the environment for new user accounts and existing user accounts.
Setup Procedure for New User Accounts

The Geonet installation procedure installs the generic login files in the directory 
.../gn5c02_00/gn_examples/. These files supply the necessary environment to 
start Geonet 5.2. A description of the files is included in the file README in this 
subdirectory. C-shell and Korn-shell generic login scripts are provided. There is an 
example .cshrc for C-shell and an example .profile and .gq_kshrc for Korn-shell.

If you want to set up the environment for a new user account, perform the following 
steps:

1. Login as the Geonet installation account and change directory to 
gn5c02_00/gn_examples/ in your installation. Edit the file .cshrc or .gq_kshrc located 
in this directory and set the path for GN_dir_sun5 and GN_dir_sgi to the location where 
Geonet 5.2 has been installed. A couple of lines at the beginning of the .cshrc or 
.gq_kshrc file will need editing to point to the Geonet 5.2 baseline. An example of these 
lines follows. You only need to edit those lines for Sun and Sgi platforms.

Example .cshrc

    set GN_dir_sun5 = /apps/gn/gn5c02_00/solaris ##SunOS 5
    set GN_dir_sgi = /apps/gn/gn5c02_00/irix ##SGI

Example .gq_kshrc

    GN_dir_sun5=/apps/gn/gn5c02_00/solaris ##SunOS 5
    GN_dir_sgi=/apps/gn/gn5c02_00/irix ##SGI

• • • • • •

NOTE: If you support Korn shell or a mixture of Korn-shell and C-shell user 
accounts: you should also define the GN_KSHRC environment variable in 
the .cshrc files in user accounts which support C shell and Korn shell. If this 
environment variable is not defined, Korn-shell will fail on remote 
launching of applications. The default setting for GN_KSHRC is 
“.gq_kshrc” for Korn-shell. There is no default setting for the .cshrc.

Edit the file .cshrc and uncomment the following line:

#setenv GN_KSHRC .gq_kshrc
This is not necessary if your site does not use Korn-shell user accounts.

2 Create the user account from which Geonet 5.2 will be run.

3 Become the new user with the command `su - <username>` and copy the required generic login files to the new user's HOME directory by running the script `gnc502_00/gn_examples/gncopy.sh`:

```
% cd <path to>/gnc502_00/gn_examples
% ./gncopy.sh
```

You can answer with `<Return>` to all the questions. Notice that CDE, the default Desktop Environment should not be running for the user when running this script, otherwise some of the Geonet default settings may be lost at logout.

4 Login to the system as the new user account and run Geonet.

```
% Geonet &
```
Setup Procedure for Existing User Accounts

If you already have a user account and/or are upgrading from a previous version of Geonet, perform the following steps:

1. Modify the generic logfiles in the new installed directory gn5c02_00/gn_examples/ as described above.

2. Become the existing user with the command `su - <user>` and run the script `gncopy.sh` from the new gn5c02_00/gn_examples directory:

```
% cd <path to>/gn5c02_00/gn_examples
% ./gncopy.sh
```

You can answer with `<Return>` to all the questions. This will save the previous login files in the user’s HOME directory and install the new ones.

3. Login as the user. This resets the environment to allow Geonet 5.2 to run. Be careful if you want to include settings taken from the previous .cshrc script file, particularly **do not** include:

   - any reference to modify LD_LIBRARY_PATH
   - alias names using existing Unix program names
   - lines that “echo” or “cat” messages to the screen. This could make the Unix `which` command to fail.

4. Run Geonet by typing

```
% Geonet &
```
Appendix B; Oracle Installation

This document provides information on installing Run-Time Oracle (RTO) version 8.1.6 using the GeoQuest Installation Tool. Refer to the document on the GeoQuest Installation Tool for screen displays and general details about each phase of the installation.

Use the GeoQuest Utilities CD for Solaris 8 to install the Run-Time Oracle on Sun Microsystems workstations that run Solaris 8 Operating System.

Use the GeoQuest Utilities CD for IRIX 6.5.x to install the Run-Time Oracle on Silicon Graphics workstations that run IRIX 6.5.x Operating System.

This document describes the following procedures:

— Preparing to install Run-Time Oracle
— Installing the Run-Time Oracle
— Post Installation Tasks
— Configuring Oracle for Data Management products
— Configuring Oracle for GeoFrame

During the installation, the Installtool will install the entire Oracle tree at the location that you choose. It will be referred later as ORACLE_ROOT.

After it is installed, RTO is organized into two main trees in the Solaris version: server and forms-reports. The version for IRIX has the server tree only.

The server tree contains Oracle RDBMS 8.1.6 and such utilities as Oracle import, Oracle export, SQL*Loader, SQL*Net, SQL*Plus.

The directory:

\$ORACLE_ROOT/<platform>/8.1.6/server

will be referred as ORACLE_HOME directory in this document, where<platform> is sun or sgi depending on the Unix platform of your Oracle server.

The forms-reports tree contains Oracle applications needed to access the RDBMS such as Oracle Forms, Oracle Reports, Oracle Graphics, SQL*Plus.
The directory:

$ORACLE_ROOT/sun/8.1.6/forms-reports

will be referred as ORACLE_FORMS directory in this document.

• • • • • •

**NOTE:** In RTO 8.1.6, the forms-reports bin directory does not contain copies of the following Oracle executables: `exp`, `imp`, `sqlldr`, `sqlload`. These executables only exist in the server tree.

A separate CD with the full documentation set distributed by Oracle is included in this release. Please refer to that documentation CD or a disk copy of it for additional information about pre-requisites to install and configure Oracle.
Preparing to install Run-Time Oracle

Refer to the Oracle Installation Guides and System Administration manuals for the corresponding Operating System for more information on preparing to install Oracle.

Prior to beginning the installation, you should perform complete backup of the system, including the complete backup of any databases which it serves.
Hardware and Operating System Requirements

Minimum Hardware Configuration

Sun
- Sun UltraSPARC 1 running Solaris 8.
- 512 Mb of RAM
- 1 Gb of swap space
- 650 Mb of disk space for the Oracle server tree
- 900 Mb of disk space for the Oracle forms-reports tree
- 1 Gb for the Oracle Tablespaces
- 386 Mb for installing the Oracle On-line Documentation (Optional)

SGI
- SGI Octane or other IRIX64 machine, capable of running IRIX 6.5.x in 64-bit runtime mode.
- 512 Mb of RAM
- 1 Gb of swap space
- 650 Mb of disk space for the Oracle server tree
- 900 Mb of disk space for the Oracle forms-reports tree
- 1 Gb for the Oracle tablespace
- 386 Mb for installing the Oracle On-line Documentation (Optional)

Required Operating System Patches

Sun
Oracle 8.1.6 does not require patches for Solaris 8.

SGI
Oracle 8.1.6 does not require patches for IRIX 6.5.x.
Required Operating System Packages

Below is the list of packages which are required to be installed on your system for Oracle 8.

### Sun

These Solaris 8 packages should be installed prior to installing the Oracle Server:

- SUNWarc Archive Libraries
- SUNWbtool CCS tools bundled with SunOS
- SUNWhea SunOS Header Files
- SUNWlibm Sun WorkShop Bundled libm
- SUNWlibms Sun WorkShop Bundled shared libm
- SUNWmfrun Motif RunTime Kit
- SUNWsprot Solaris Bundled tools
- SUNWtoo Programming Tools

To determine if the Solaris 2.x packages are installed, enter the following command as root user:

```
% pkginfo -i SUNWarc SUNWbtool SUNWhea SUNWlibm SUNWlibms SUNWmfrun SUNWsprot SUNWtoo
```

### SGI

These IRIX 6.5.x packages should be installed prior to installing the Oracle Server:

- IRIX 6.5.x IRIS Development option
- eoe.sw.svr4net subsystem
- compiler_dev.sw.base (required for Relinking the Oracle binaries if the Oracle owner account is not a member of the group called `dba`).
• Check if the Assembler is installed with this command:

```bash
% ls -l /usr/bin/as
lrwxr-xr-x  1 root  sys       17 Dec 18  1998
/usr/bin/as -> ../lib/driverwrap
```

```bash
% versions long | egrep '/bin/as$|/lib/driverwrap$
```

```bash
l     0     0 compiler_dev.sw.base    usr/bin/as
f 31950  185 compiler_dev.sw.base    usr/lib/driverwrap
```

### Kernel Parameters

Configure the UNIX kernel interprocess communication (IPC) parameters to accommodate the System Global Area (SGA) structure of the ORACLE database. ORACLE Server Manager will not start the database if your system does not have adequate resources to accommodate the SGA.

### Sun

Use the `ipcs -b` command to obtain a list of the current shared memory and semaphore segments and their identification number and owner. Verify the content of the `/etc/system` file, which should contain the settings described below.

#### NOTE:

Because shared memory in Solaris 2.x is dynamically loaded, running `ipcs` may return a message that the shared memory facility is not in the system. After a program such as Oracle Server is executed, the shared memory is loaded. Check your `/etc/system` file to verify that your system has been configured with enough shared memory.

Set the kernel parameters at the following maximums:

- Size of a shared memory segment
- Number of shared memory segments in the system
- Number of shared memory segments a user process may attach
- Amount of shared memory that may be allocated system-wide
The parameters shown in the following table control the allocation of shared memory. These values are optimal for one instance and are based on the default init<sid>.ora file. If you plan to have more than one instance, or to modify the init<sid>.ora file extensively, you should set these parameters higher. The Oracle Corporation recommends that you set these parameters as high as possible for your operating system.

**NOTE:** Setting these parameters at values higher than those allowed by your operating system can prevent your machine from booting up. Refer to your operating system documentation to ensure that you set these parameters within the allowable limits.

### Solaris IPC Parameter Values relevant to ORACLE

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Recommended Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHMMAX</td>
<td>At least half of Physical Memory in bytes.</td>
<td>The maximum size of a single shared memory. Should be larger than the sum of the SGA size(s), but less than the total Physical Memory.</td>
</tr>
<tr>
<td>SHMMNI</td>
<td>100</td>
<td>The maximum number of shared memory identifiers.</td>
</tr>
<tr>
<td>SHMSEG</td>
<td>10</td>
<td>The maximum number of shared memory segments that can be attached by a process.</td>
</tr>
<tr>
<td>SEMMNI</td>
<td>100</td>
<td>The number of semaphore set identifiers in the system. Each instance creates at least one semaphore set.</td>
</tr>
<tr>
<td>SEMMSL</td>
<td>equal to the value of the initialization parameter <code>processes</code></td>
<td>The maximum number of semaphores that can be in one semaphore set. Should be at least equal to the value of ORACLE <code>processes</code> parameter or the greatest value if several instances run on the server.</td>
</tr>
<tr>
<td>SEMMNS</td>
<td>2000</td>
<td>The number of semaphores in the system. This number should be greater than SEMMSL x Number of ORACLE instances.</td>
</tr>
</tbody>
</table>

The following is an example of the lines from `/etc/system`, based on 512 MB of physical memory and one Oracle instance.

The SHMMAX parameter is set as half the physical memory in bytes:
256 x 1024 x 1024 = 268435456 bytes.

```bash
set shmsys:shminfo_shmmax=268435456
set shmsys:shminfo_shmmni=100
set shmsys:shminfo_shmseg=10
set semsys:seminfo_semmni=20
set semsys:seminfo_semmsl=300
set semsys:seminfo_semmsns=600
```

Table 2 shows the default values for these and other related IPC parameters if not explicitly included in the file `/etc/system`. Use the Solaris `sysdef` command to list the current parameters.

Semaphore and Shared Memory Interprocess Communication Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>semsys:seminfo_semmni</td>
<td>10</td>
<td>Number of semaphore identifiers</td>
</tr>
<tr>
<td>semsys:seminfo_semmns</td>
<td>60</td>
<td>Number of semaphores in the system</td>
</tr>
<tr>
<td>semsys:seminfo_semmnu</td>
<td>30</td>
<td>Number of processes using the undo facility</td>
</tr>
<tr>
<td>semsys:seminfo_semmsl</td>
<td>25</td>
<td>Maximum number of semaphores, per id</td>
</tr>
<tr>
<td>semsys:seminfo_semopm</td>
<td>10</td>
<td>Maximum number of operations, per semaphore call</td>
</tr>
<tr>
<td>semsys:seminfo_semume</td>
<td>10</td>
<td>Maximum number of undo structures per process</td>
</tr>
<tr>
<td>shmsys:shminfo_shmmax</td>
<td>1048576</td>
<td>Maximum shared memory segment size</td>
</tr>
<tr>
<td>shmsys:shminfo_shmmni</td>
<td>100</td>
<td>Number of shared memory identifiers</td>
</tr>
<tr>
<td>shmsys:shminfo_shmseg</td>
<td>6</td>
<td>Segments, per process</td>
</tr>
</tbody>
</table>
How to Tune the Interprocess Communication Parameters

1. Become superuser.

2. Add a line to the /etc/system file using the syntax described above.

3. Reboot the system.

4. Verify the kernel parameter change:

   ```
   # sysdef | grep <parameter>
   ```

   The kernel parses the /etc/system file during autoconfiguration and overrides the default value for the parameters specified in this file.

SGI

Use the `ipcs -b` command to obtain a list of the current shared memory and semaphore segments and their identification number and owner. The shared memory and semaphore segments parameters are defined in the file /var/sysgen/stune. The utility ‘systune’ should be used to view or set the kernel parameters, although the IRIX default values for these parameters are normally suitable for Oracle installation and do not require to be changed.

The following is an example of the lines from `systune` output:

```
semaem = 16384 (0x4000)
semvmx = 32767 (0x7fff)
semopm = 100 (0x64)
semmsl = 100 (0x64)
semnni = 300 (0x12c)
sshmseg = 2000 (0x7d0)
shmnni = 4136 (0x1028)
shmmin = 1 (0x1) 11
shmmax = 858993460 (0x33333334) 11
```
Example of the command to change a parameter:

```
# systune -i
systune-> semmsl = 600
systune-> quit
```

If you modified the shared memory or semaphore segments parameters run the SGI autoconfiguration program and reboot the system:

```
# autoconfig -v
# reboot
```

**Multiple Oracle instances**

If another database is running on the system, it utilizes shared memory and semaphores. You can find what is being allocated with the `ipcs` command, for example:
% ipcs -b

IPC status from <running system> as of Wed Apr 18 16:07:26 CDT 2001

Message Queue facility not in system.

T     ID     KEY        MODE       OWNER    GROUP  SEGSZ  

Shared Memory:

m 1500   0xd463f24c --rw-r----- oracle8 dba 101384192

T     ID     KEY        MODE       OWNER    GROUP NSEMS  

Semaphores:

s 4128768   0x876e4110 --ra-r----- oracle8 dba 254

The SEGSZ shows the size of the associated shared memory segment and corresponds to the size of the SGA. The NSEMS shows the number of semaphores in the set associated with the semaphore entry.

For example, if the shared memory segment size is 120 Mb and the SEGSZ shows 101384192 bytes already used, creation of a new instance with SGA size of 40Mb would fail. The shminfo_shmmax value should be increased at least to 150 Mb in this example.

Memory availability

The commands `/etc/prtconf` and `/usr/sbin/swap -s` can be used for Sun and `/usr/bin/hinv` and `/sbin/swap -s` for SGI. These commands show the amount of main memory and the amount of swap space available on the current machine.
This information can be used to tune the Oracle instance after it has been created. Add more physical memory as required if you need to increase the shminfo_shmmax value.

**Current processing**

The database server machine should be checked for its current processing load. Attention should be paid to the amounts of memory used by active processes, as well as the total number of processes.

Ideally, the database server should not be loaded by other processes but Oracle processes. That would allow it to be optimized for the database access.

The machine should not be restricted by the number of processes that can be created. If the Oracle Multi-Threaded Server is not being used then every database connection leads to the creation of the process (and memory allocated to it) on the Oracle server machine.

**Available disks**

The number, size and configuration of available disks on the database server should be found. The command `df -kl` provides this information. It is strongly recommended to use local disks in the Oracle server machine for installing the Oracle and placing all the database files.

Splitting up the tablespace files between different disks generally gives better performance, as disk contention is minimized.

**Listener Service**

The service name listener does not need to be explicitly defined in the system services anymore. Previously it had to be defined in the `/etc/services` file. The only requirement now is that the tcp/ip port selected for the Listener Oracle process is free.

Verify that the port is free with the command:

```bash
% netstat -an | grep <port_number>
```
This command should return empty when the port is free. You can use the same port as an already existing listener process only if you replace/update the listener.ora TNS configuration file. Remember to select “Create SQL*Net configuration files” when using the Installtool.
Other pre-installation tasks performed by root

Create dba and oper groups

If you do not already have a group named dba, create one. The group name dba is hard-coded into several Oracle executables. If you name the DBA group something other than dba, the Oracle executables must be relinked. This can be done when installing Oracle using Installtool.

If you want to have an additional group of users with restricted OPERATOR database privileges, create a group named oper. Members of this group have separate restricted operator database privileges.

Use the operating system administration utility to create the dba and oper groups.

Create Oracle software owner

Create a UNIX login account to be used for the installation of Oracle. Having such account allows a division of administration functions and operating environments between the Oracle Database Administration and general software configuration and administration. The Oracle software owner can be called anything, but should be a member of the dba group discussed above.

Use the operating system administration utility to create an Oracle software owner account with the following properties:

- **Login name**: oracle (suggested)
- **User ID number**: free User ID between 3 and 32767
- **Login shell**: /bin/csh (recommended)
- **Default Group ID number**: Group ID of the DBA group

**NOTE:** On SGI the Oracle software owner account must be local to the machine which is used as Oracle server. If a NIS account having the same name as Oracle owner exists, create the local account with the matching User ID.
ORACLE_ROOT directory

Create or select the directory at the location where you want Oracle software to be installed. The Oracle software owner should own that directory.

Planning for new installation

New installation is used when you do not have Oracle 7.3.4 installed, or, if you choose to create a new Oracle 8.1.6 instance and then move the database from Oracle 7.3.4 using Oracle export and import utilities.

Planning for upgrade installation

Upgrade installation is used when you have Oracle 7.3.4 installed and an Oracle 7.3.4 instance running and you want to migrate this instance “in place” to Oracle 8.1.6.

Before attempting an upgrade installation perform the following procedures and checks:

Perform a cold backup (i.e. make a tar file of all the datafiles with the database shut down) of the Oracle 7.3.4 database before attempting the upgrade installation. It is recommended to perform full Oracle export of the Oracle 7.3.4 database as well. For more information on how to make a cold Oracle backup, full Export/Import, defragment a database, etc, refer to the Apendix D in this document.

• Make sure that you are familiar with the Oracle 7.3.4 instance configuration. See “Existing Oracle instances” below for information on how to find the existing Oracle databases. Study the following information:
  — User account used for 7.3.4 database (you have to use the same account for Oracle 8.1.6 installation).
  — $ORACLE_HOME path for 7.3.4 database.
  — $TNS_ADMIN path for 7.3.4 database.
  — $ORACLE_SID for 7.3.4 database.
  — $TWO_TASK for 7.3.4 database.
• Locate the initialization file, init$ORACLE_SID.ora (normally it is located at Oracle 7.3.4 $ORACLE_HOME/dbs).
• Locate all of your database files via SQL*Plus, by running the following queries when connected as SYSTEM:

SQL> select name from v$controlfile;

SQL> select member from v$logfile;

SQL> select file_name from dba_data_files;

• To validate this information perform the following tasks:
  — Start and stop Oracle 7.3.4 instance
  — Start and stop the SQL*Net listener
  — Verify that database files reside in the locations found.

• Make sure that the file init$ORACLE_SID.ora is present in Oracle 7.3.4 $ORACLE_HOME/dbs server directory. The upgrade procedure will fail if this file is not located there.

• Make sure that there is enough free space on the partition where you are planning to install Oracle 8.1.6 software since it will be installed into a separate location from Oracle 7.3.4.

Existing Oracle instances

Whether you perform new or upgrade installation you need to know if there are other Oracle instances running on the Oracle server machine. When there are multiple Oracle instances running you should consider it when configuring the kernel parameters, disk space, memory, listener port(s), database names, SQL*Net aliases (TWO_TASK).

If you have previously installed Oracle using GeoQuest Installtool, the best source of such information is from the .InstallLog_data_time file. You can find this file in $HOME of the Oracle software owner. View this logfile and find the Installation Parameter Settings portion of it: all parameters used for the previous Oracle installation are listed here.

Alternatively you can use methods listed below to gather the information about Oracle instances running on the machine.

Login into the Oracle server and list the Oracle processes:

% ps -ef | grep ora_
% ps -ef | grep lsn

oracle 2460 1 0 Apr 15 ? 4:20
/export/oracle/sun/8.1.6/server/bin/tnslsnr LISTENER - inherit

The LISTENER path usually contains $ORACLE_HOME for the Oracle server: it is one directory back from the “bin” directory; in the example above the $ORACLE_HOME is: /export/oracle/8.1.6/server

The process names for the instance contain the $ORACLE_SID, in the example above the $ORACLE_SID is: kosmos.

The Oracle software owner is the owner of the ora_ processes, in the example above the Oracle software owner is: oracle.

To find $TNS_ADMIN, go to the Oracle owner HOME directory:

% cd ~oracle

and check the script Oracle_Runtime.csh. This script has all the required environment variables for Oracle.

In $TNS_ADMIN you should find these four files:

listener.ora tnsnames.ora tnsnav.ora sqlnet.ora

From the listener.ora file you can find the following information:
• $ORACLE_HOME:ORACLE_HOME in listener.ora
• $ORACLE_SID:SID_NAME in listener.ora
• Communication port:PORT in listener.ora

While the Oracle installation procedure of the GeoQuest Utility kit does not utilize or amend the "oratab" file, it should be considered when looking for information about any databases which may be running or installed in a specific system. The oratab file, if it exists, is located in /var/opt/oracle/oratab.

The format of the oratab file is:

$ORACLE_SID:$ORACLE_HOME:Y/N {Startup at boot time flag}

For example:

oracle1:/home/jiffy/app2/oracle-7.3.4:Y

Do not add the oratab file if it does not exist already.
Installing the Run-Time Oracle

Run-Time Oracle is provided as part of a general GeoQuest Utilities CD that also includes Geonet and other common applications. The following is a step-by-step instruction of the installation.

1. Insert the GeoQuest Utilities Version 2.0 CD-ROM for the corresponding Operating System in the CDROM drive. This will normally be mounted automatically. If problems are experienced with mounting the CD, see your local system administrator for assistance.

2. Login to the Oracle software owner account. If this is remote installation set the DISPLAY environment variable:

   % setenv DISPLAY my_host:0.0

3. Verify the following in the terminal from which you are going to start the installation:
   - Check that there is enough free space on the partition where you are planning to install Oracle software, for example:

   % df -k /<directory_to_install_RTO>

   - Make sure that the Oracle software owner can write on the partition that you dedicated to the Oracle tree.

   - Check that Oracle software owner “umask” is set to 022. Use command umask to check and to modify the umask, for example:

   % umask 022

   - Check that these environmental variables: ORACLE_HOME, TNS_ADMIN, TWO_TASK or ORACLE_SID are not set before starting the installation, for example:

   % env | grep ORACLE_HOME

   % env | grep TNS_ADMIN

   % env | grep TWO_TASK

   % env | grep ORACLE_SID

   These commands should not return anything when the corresponding variables are not set. If these variables are set, unset them - having them set may have a adverse effect on the outcome of the installation and it is not recommended.
• Check that you do not have any aliases such as `rm`, `du`, `ls`, etc., set in your environment. Use the commands `alias` and `unalias` to check and remove aliases from the environment.

4 Change directory to the top directory of the CD-ROM and perform the following command:

```bash
./installtool
```

• • • • • •

**NOTE:** You must enter the startup command as `./installtool`. If not you will receive a message indicating that it is "Unable to find binaries for the installtool".

A log file called `~/.InstallLog_time_date` will be created.

**Welcome Phase**

5 Select Next to continue.

**Select Components to Install Phase**

6 Turn on the Oracle 8.1.6 button and then click the arrow to reveal which components are selected.

• • • • • •

**NOTE:** Geonet may also be installed at this time, depending on whether it will be administered by the same account as Oracle (not recommended).

7 Select the following individual components, based on the needs of the software which will use the Oracle installation:

• Oracle Server Runtime 8.1.6 (for all installations)
• Oracle Forms-reports Runtime 8.1.6 (for using it with GeoQuest Data Management products, such as Finder, AssetDB, LogDB, SeisDB)

8 Select Next to continue.
Select Locations to Install Components Phase

9 Select a disk location to install Oracle tree (the default will be home directory of the account that started the installation). Use the Browse button to bring up the Browse Dialog to select a different location. Use the button to the right of Location to create the directory, if the red icon is shown.

During the installation Installtool will create a directory at the location that you choose in this phase, and entire Oracle tree will be installed under that directory. This directory will be called sun/8.1.6 on Sun and sgi/8.1.6 on SGI.

10 Select Next to continue.

Installation Steps Phase

11 Select steps applicable to your installation. The installation steps (procedures) for Oracle are:

- Relink Oracle executables
- Upgrade an existing Oracle 7.3.4 instance to version 8.1.6
- Create the SQL*Net configuration files
- Setup Oracle system boot time startup [requires root]
- Create an Oracle instance for GeoQuest Application
- Create Data and Index tablespacess (for GeoFrame)
- Register Oracle Doc Browser in Geonet

Listed below is a brief description of the installation steps available:

Re-link Oracle Runtime Software

This procedure is usually selected for a new installation. It relinks the Oracle executables for the system it is being installed on. This must be done if the account being used to install Oracle does NOT belong to the group named dba since that name is hardcoded into the executables. The relinking procedure changes the name of the group to the name of the primary group of the account that is being used to install Oracle. If this option is selected and Oracle software owner belongs to the group named dba, the re-link will not be done. This procedure is selected by default.
Upgrade an existing Oracle 7.3.4 instance to version 8.1.6

Use this step if you have an existing Oracle 7.3.4 installation and wish to migrate it to version 8.1.6. This procedure should be executed only after having done a cold backup of the database files. For more information on backups, refer to the Appendix D in this document. To review the Oracle documentation, refer to the document Oracle8i Migration Release 2 (8.1.6) A76957-01 available in the Oracle Documentation CD. Use a web browser to view this URL:

file:/cdrom/816_docs/DOC/server.816/a76957/toc.htm

The upgrade procedure executes the following sub-procedures:

- Installing new Oracle 8.1.6 directory trees.
- Verifying the status of the 7.3.4 database.
- Moving necessary extra files from the new Oracle server tree to the old one.
- Creating new parameter and control files.
- Executing Oracle supplied migration scripts to update the tablespaces.
- Restarting the database to utilize the new executables.

When you migrate or upgrade a database, all the existing catalogs and projects are converted to the new Oracle version.

The upgrade procedure preserves the existing parameters from the previous Oracle 7.3.4 instance. Make sure that all the parameter names in the file $734_ORACLE_HOME/dbs/init$ORACLE_SID.ora are lowercase before starting the upgrade phase; this is a requirement for the installtool, not for Oracle. Some of the previous parameters are obsolete or not valid for Oracle 8. The installtool manages any previous parameter that was created with the Utility kit 1; however, if you have customized your parameters then check the list below. Any parameter from that list needs to be commented out if found in the parameter file before the upgrade procedure is started, otherwise Oracle8 does not start. After this procedure is executed, you should adjust the initialization parameters in the new file $ORACLE_HOME/dbs/init$ORACLE_SID.ora to correctly work with Oracle 8. You can take the template initialization files in $ORACLE_HOME/setup/initoracle*.ora as a reference. The database should be restarted to take the changes. Also, it is possible that you need to adjust the kernel parameters to accomodate a greater SGA.
One alternate method to accomplish the database migration is by performing a Full Oracle export/Full Import task. The installtool can be used to create a new, empty Oracle 8.1.6 instance, into which to make the Full Import. The advantages of this alternate method are:

- you get a defragmentation of your database at the same time
- you can rename your ORACLE_SID
- create bigger tablespaces and most probably better organized datafiles, i.e., get rid of small ones.
- the new database can be installed on a newer machine.

Disadvantages:

- it takes longer (mainly the Import part)
- more manual intervention, which increases the risk of errors.

For more information on how to make a cold Oracle backup, full Export/Import, defragment a database, etc, refer to the Appendix D on this document.

**NOTE:** It is not possible to create an Oracle instance and upgrade an existing Oracle 7.3.4 instance in the same installation pass.

**Create the SQL*Net configuration files**

This procedure is usually selected for a new installation; it also should be selected when the ‘Upgrade an Oracle Instance’ step is selected. It sets up or updates the SQL*Net configuration files listener.ora, tnsnames.ora, sqlnet.ora, tnsnav.ora and starts the listener. If an existing $TNS_ADMIN directory is selected in the installtool, it is recommended to check the files updated by running this step. The original existing files are saved.

**NOTE:** Do not select the $ORACLE_HOME/network/admin directory which is created as part of the Oracle tree, otherwise it will be renamed and then some directories under $ORACLE_HOME/network will be missing. The recommended location is $ORACLE_ROOT/network/admin.

This procedure is selected by default.
Setup Oracle system boot time startup [requires root]

This procedure installs the scripts necessary for Oracle to start and stop automatically when the server is rebooted. These scripts are installed in /etc/rc0.d, /etc/rc2.d and /etc/init.d directories.

Select this procedure also when you have selected Upgrade an Oracle Instance, in order to update the boot startup scripts with the new Oracle values. This procedure requires root access and you will be prompted for the root password. It is selected by default.

Create an Oracle instance for GeoQuest Application

This procedure is usually selected for a new installation. It creates an Oracle instance after Oracle is installed. This procedure is selected by default.

Create Data and Index tablespaces (for GeoFrame)

This procedure creates two tablespaces. The default names for these tablespaces are TSDATA and TSINDEX. This procedure should be selected if this is new installation of Oracle and you are creating an Oracle instance for GeoFrame. This procedure is selected by default.

Root Password Phase

12 Enter the root password for setting up the script to startup Oracle instance at the boot time if you selected Setup Oracle system boot time startup procedure. If the root password is not provided, a script is created and placed in the Oracle software owner home directory and named: GeoQuest_Utilities2_0_root_<date>.csh. This script can be run by root after installation is complete, to setup the Oracle boot time startup scripts.

13 Select Next to continue.

Installation Parameters Phase

14 Specify values for parameters listed in the “Parameters” area. This list is determined by which steps were selected during Installation Steps phase, therefore, all the parameters in table below might not be available.
Some parameters will have default values which can be changed if desired. The parameters listed in the Parameter area of the screen are divided into the following groups:

- Oracle Parameters
- Oracle Passwords
- Oracle Directories
- Oracle Datafile Sizes
- Oracle Tablespace Names

All installation parameters are described in the Table.

Run Time Oracle Installation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Oracle Parameters</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORACLE_HOME Directory for v.7.3.4</td>
<td>$734_ORACLE_ROOT/oracle-7.3.4_&lt;platform&gt;/server/app/oracle/product/7.3.4</td>
<td>$ORACLE_HOME directory for the Oracle 7.3.4 database that you are going to migrate (if any). Notice that you should use the long path here and not the soft link.</td>
</tr>
<tr>
<td>Instance Identifier</td>
<td>ora8gq</td>
<td>ORACLE_SID of the instance (8 characters max.) Try using a name that does not include the Oracle version, so it will not be confusing if it is migrated to a newer version.</td>
</tr>
<tr>
<td>Instance Alias</td>
<td>geoquest1</td>
<td>TNS alias (TWO_TASK) of the Oracle instance. This name can be the same as ORACLE_SID, but in that case it has to have 8 characters max.</td>
</tr>
<tr>
<td>Network Admin Directory</td>
<td>$ORACLE_ROOT/network/admin</td>
<td>Directory containing the Oracle SQL*Net configuration files. Do NOT select the location $ORACLE_HOME/network/admin</td>
</tr>
<tr>
<td>Listener TCP Port Number</td>
<td>1521</td>
<td>Port number on which listener process will be listening for connection requests.</td>
</tr>
<tr>
<td>Network Domain Name</td>
<td>world</td>
<td>Name of the network domain. Use the command domainname to determine the name of the network domain. Leave it empty or set to world if none is defined.</td>
</tr>
<tr>
<td><strong>Oracle Passwords</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYS account password</td>
<td>change_on_install</td>
<td>Password for SYS Oracle account. DO NOT CHANGE until finished installing GeoQuest products.</td>
</tr>
<tr>
<td>SYSTEM account password</td>
<td>manager</td>
<td>Password for SYSTEM Oracle account.</td>
</tr>
</tbody>
</table>
Run Time Oracle Installation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oracle Directories</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Default Datafile Directory 1 (DB_DEF_DISK1_DIR)</td>
<td>$ORACLE_ROOT/dbs</td>
<td>Variable DB_DEF_DISK1_DIR will be used in next parameters as a default directory for Oracle datafiles.</td>
</tr>
<tr>
<td>Default Datafile Directory 2 (DB_DEF_DISK2_DIR)</td>
<td>Default Datafile Directory 1</td>
<td>Variable DB_DEF_DISK2_DIR will be used in next parameters as a default directory for Oracle datafiles.</td>
</tr>
<tr>
<td>System tablespace directory</td>
<td>Default Datafile Directory 1</td>
<td>Directory where the datafile for the tablespace will be created.</td>
</tr>
<tr>
<td>Rollback tablespace directory</td>
<td>Default Datafile Directory 1</td>
<td>Directory where the datafile for the tablespace will be created.</td>
</tr>
<tr>
<td>Temporary tablespace directory</td>
<td>Default Datafile Directory 2</td>
<td>Directory where the datafile for the tablespace will be created.</td>
</tr>
<tr>
<td>Logfile Directory 1</td>
<td>Default Datafile Directory 1</td>
<td>Directory where the first logfile will be created.</td>
</tr>
<tr>
<td>Logfile Directory 2</td>
<td>Default Datafile Directory 2</td>
<td>Directory where the second logfile will be created.</td>
</tr>
<tr>
<td>Control file Directory 1</td>
<td>Default Datafile Directory 1</td>
<td>Directory where the first control file will be created.</td>
</tr>
<tr>
<td>Control file Directory 2</td>
<td>Default Datafile Directory 2</td>
<td>Directory where the second control file will be created.</td>
</tr>
<tr>
<td>Data Tablespace Directory</td>
<td>Default Datafile Directory 2</td>
<td>Directory where the datafile for the tablespace will be created.</td>
</tr>
<tr>
<td>Index Tablespace Directory</td>
<td>Default Datafile Directory 1</td>
<td>Directory where the datafile for the tablespace will be created.</td>
</tr>
<tr>
<td>Oracle Datafile Sizes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database Size</td>
<td>Medium</td>
<td>Size of the database in matter of default sizes of the database files and values of init.ora parameters. You can change the size of your database at later time using Oracle commands and by tuning the init.ora parameters.</td>
</tr>
<tr>
<td>System Tablespace Size (M)</td>
<td>200</td>
<td>Size of the Oracle SYSTEM tablespace in megabytes.</td>
</tr>
</tbody>
</table>
Run Time Oracle Installation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rollback Tablespace Size (M)</td>
<td>400</td>
<td>Size of the tablespace for the rollback segments in megabytes.</td>
</tr>
<tr>
<td>Temporary Tablespace Size (M)</td>
<td>50</td>
<td>Size of the temporary tablespace in megabytes.</td>
</tr>
<tr>
<td>Logfile Size (M)</td>
<td>10</td>
<td>Size of the redo logfiles in megabytes.</td>
</tr>
<tr>
<td>Data Tablespace Size (M)</td>
<td>500</td>
<td>Size of the data tablespace in megabytes.</td>
</tr>
<tr>
<td>Index Tablespace Size (M)</td>
<td>400</td>
<td>Size of the index tablespace in megabytes.</td>
</tr>
<tr>
<td>Oracle Tablespace Names</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Tablespace name</td>
<td>TSDATA</td>
<td>Name of the data tablespace.</td>
</tr>
<tr>
<td>Index Tablespace name</td>
<td>TSINDEX</td>
<td>Name of the index tablespace.</td>
</tr>
</tbody>
</table>

15 Select **Next** to continue after all parameters are correctly specified.

**Ready to Extract Files Phase**

16 Select **Next** to continue.

**Extracting Files Phase**

The files are extracted during this phase.

**Ready to Examine Prerequisite Procedures Phase**

17 Select **Next** to continue.

**Examining Prerequisite Procedures Phase**

Installtool validates parameters and environment for the installation.
Root Password Phase

18 Enter the root password for setting up the script to startup Oracle instance at the boot time. If the root password will not be provided, the special script will be created and placed in the Oracle software owner home directory and named: GeoQuest_Utilities2_0_root_<timedate>.csh.
This script can be run after installation is complete.

19 Select Next to continue.

Ready to Execute Root Procedures Phase

20 Select Next to create the Oracle instance startup script.

Ready to Execute Procedures Phase

The selected steps will be executed after this phase.

If you wish to change the parameters in the file init$ORACLE_SID.ora before the instance is created you can do it before selecting Execute. To do that open an xterm, login as Oracle software owner and change directory to the $ORACLE_HOME/setup. Database block size is the parameter which can not be changed once the instance is created.

The default database block size with which the Oracle instance is created is 8192. To change the database block size modify the value for db_block_size parameter in one of the following files located under $ORACLE_HOME/setup directory: initoracle_large.ora, initoracle_medium.ora, initoracle_small.ora depending on whether you chose Large, Medium or Small Database Size parameter.

If you want to set db_block_size to the value different from the default value, consult with Oracle Reference Addendum for Sun SPARC Solaris 2.x or Oracle Reference Addendum for MIPS ABI Systems (SGI).

When you change the database block size, the values of other parameters which are measured in the database blocks, might need to be adjusted. For example, the db_block_buffers parameter is set in this manner. If you double the block size, you might need to cut the db_block_buffers
parameter in half. Failing to do this will double the size of the data buffer cache, possibly causing problems with the memory management on your server.

21 Select **Next** to execute selected installation steps.

**Executing Procedures Phase**

Installtool executes the steps selected in the Installation Steps phase.

⚠️ 

**NOTE:** Never proceed to this step before making a cold backup of your database if you are doing in-palce migration.

Review the logfile displayed in the Message area to confirm that procedures are executed successfully.

**Ready to Validate Files Phase**

22 Select **Next** to validate the checksums for the installed files.

**Validating Files Phase**

Installtool validates that the checksums are correct for the installed files.

**Installation is Complete Phase**

23 Select **Quit** to exit and save the restart file.
Post Installation Tasks

This chapter discusses tasks that may need further explanation after the basic Oracle installation is completed.

Installation Verification

Oracle Database Processes

There are two type of Oracle processes. Processes of first type run the database and the other listens for the SQL*Net connections to the database.

The Oracle processes running on the database server can be seen if you type:

```
% ps -ef | grep ora_
```

```
oracle   684     1  0 May 19 ? 0:02 ora_pmon_ora8gq
oracle   686     1  0 May 19 ? 1:26 ora_dbwr_ora8gq
oracle   688     1  0 May 19 ? 2:19 ora_lgwr_ora8gq
oracle   690     1  0 May 19 ? 9:15 ora_smon_ora8gq
oracle   692     1  0 May 19 ? 0:01 ora_reco_ora8gq
```

Shutting down an Oracle database can be achieved in several ways. It is important for an Oracle database to be shut down cleanly. A clean shutdown is accomplished by using sqlplus, as illustrated below, or by using the supplied scripts. These allow the user to manually execute the scripts or, with proper setup, allow you to use the scripts during the re-boot process to cleanly stop and start the Oracle database.

Notice that svrmgrl is being dropped by Oracle and sqlplus should be used instead. The /nolog qualifier allows to use previous svrmgrl syntax within sqlplus.
Listed below is a sample procedure used to manually shutdown the database using sqlplus:

% sqlplus /nolog

SVRMGR> connect internal

SVRMGR> shutdown immediate

SVRMGR> exit

Listed below is a sample procedure used to manually startup the database using sqlplus.

% sqlplus /nolog

SVRMGR> connect internal

SVRMGR> startup

SVRMGR> exit

The listener processes running on the database server can be seen if you type:

% ps -ef | grep tns

```
oracle8 8147 1 0 May 20 ? 0:02
/apps/sun/8.1.6/server/bin/tnslsnr LISTENER -i
```

Listed below is a sample procedure used to start the SQL*Net listener:

% lsnrctl start

Listed below is a sample procedure used to stop the SQL*Net listener:

% lsnrctl stop

Listed below is a sample procedure used to get the status of the SQL*Net listener:

% lsnrctl stat

Listed below is how to get help on Lsnrctl.

% lsnrctl help
Instance verification

The database installation should be verified by using the Oracle installation account and performing the following checks and verifications.

During the installation of Oracle 8.1.6, a file is created named ORACLE_ROOT/Oracle_Runtime.csh, which has the necessary environment settings to start, examine and stop the newly created or upgraded database instance.

Change directory to ORACLE_ROOT and execute the command:

```bash
% source Oracle_Runtime.csh
```

To check that Oracle instance is up and running you can:

```bash
% sqlplus system/<password>
```

This query should return results similar to the output below, which shows the ORACLE_SID of the database:

```
SQL> select name,created,open_mode from v$database;

NAME      CREATED   OPEN_MODE
--------- --------- ----------
ORA8GQ 14-APR-01 READ WRITE
```

```
SQL> show sga;

Total System Global Area  100859584 bytes
Fixed Size                 69312 bytes
Variable Size             74035200 bytes
Database Buffers         26214400 bytes
Redo Buffers              540672 bytes
```
If none of the above work, check that the instance is running on the database server and that there is also a listener process available.

Network connections can be verified by using the `tnsping` command.

```
% tnsping $TWO_TASK

TNS Ping Utility for Solaris: Version 8.1.6.3.0 -
Production on 22-APR-2001 11:02:50

(c) Copyright 1997 Oracle Corporation. All rights reserved.

Attempting to contact
(ADDRESS=(COMMUNITY=TCP.world)(PROTOCOL=TCP)(Host=sc-krypton)(Port=1521))

OK (10 msec)
```
To further check that the database and the listener are running and configured correctly, enter the following command to connect to the database as the system account. By specifying TWO_TASK it will connect to this instance via SQL*Net:

```
% sqlplus system/<password>@TWO_TASK
```

The database instance can be checked for completeness. Verify that the tablespaces have been created during the installation:

```
% sqlplus system/<password>
```

```
SQL> select TABLESPACE_NAME, STATUS from DBA_TABLESPACES;

TABLESPACE_NAME    STATUS
---------------------  ---------
  SYSTEM              ONLINE
  ROLLBACK            ONLINE
    TEMP              ONLINE
   TSDATA             ONLINE
   TSINDEX            ONLINE
   GMDATA             ONLINE
   GMINDEX            ONLINE

7 rows selected.
```
You can view the Oracle datafiles using query:

```
% sqlplus system/<password>
```

```
SQL> select FILE_NAME from DBA_DATA_FILES;
FILE_NAME
--------------------------------------------------------
/data/disk3/ora_816_tbls/GMDATAora8gq.dbf
/data/disk4/ora_816_tbls/GMINDEXora8gq.dbf
/data/disk1/ora_816_tbls/dbsora8gq.dbf
/data/disk1/ora_816_tbls/rollbackora8gq.dbf
/data/disk4/ora_816_tbls/tempora8gq.dbf
/data/disk3/ora_816_tbls/TSDATAora8gq.dbf
/data/disk4/ora_816_tbls/TSINDEXora8gq.dbf
/data/disk4/ora_816_tbls/tstindexora8gq.dbf
```

8 rows selected.

If you have performed a database migration by selecting the Upgrade Oracle Instance step, you have to disable Oracle 7.3.4 from starting again.

Change directory to /etc/rc2.d and check the boot startup script for Oracle, verifying that it was updated to use the new ORACLE_HOME and new syntax. Remove all startup scripts related to the previous Oracle 7.3.4 installation from /etc/rc2.d, /etc/rc0.d and /etc/init.d.
Appendix C; GeoFrame Installation

This document provides general information and describes various tasks and procedures necessary for a successful installation of GeoFrame.

GeoFrame 4.0 is a full baseline release.

Major components related to the release of GeoFrame 4.0 are;

- Support for GUCE 6.0 (Solaris 8, IRIX 6.5.x, Oracle 8.1.6, Flexlm 7.2e)
- New functionality for applications and other application bug fixes. Please see Product Release notes for details.

This document assumes that the proper environment is installed and functional and describes the following procedures:

Pre-Installation Tasks—Describes the tasks that are needed before the installation can proceed. Checks OS patches, and setup user accounts.

Install GeoFrame—Provides step-by-step instructions for installing GeoFrame with the Install Tool and different installation scenarios.
Pre-Installation Tasks

System Requirements

Software Space Requirements

The space required for a GeoFrame 4.0 installation is as follows:

A full GeoFrame installation requires approximately 2.36 gigabytes of disk space for a Solaris installation and 2.28 gigabytes for an SGI installation. This does not include the space required for the Oracle software or database. The following table gives the approximate disk requirements for individual GeoFrame products. Please see note concerning sizes below.

<table>
<thead>
<tr>
<th>Component</th>
<th>Solaris Size (Mbytes)</th>
<th>SGI Size (Mbytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeoFrame Kernel</td>
<td>481</td>
<td>459</td>
</tr>
<tr>
<td>Geology</td>
<td>776</td>
<td>743</td>
</tr>
<tr>
<td>Charisma</td>
<td>734</td>
<td>772</td>
</tr>
<tr>
<td>IESX</td>
<td>770</td>
<td>693</td>
</tr>
<tr>
<td>InDepth</td>
<td>732</td>
<td>746</td>
</tr>
<tr>
<td>Petrophysics</td>
<td>682</td>
<td>617</td>
</tr>
<tr>
<td>Mapping &amp; Visualization</td>
<td>1376</td>
<td>1342</td>
</tr>
<tr>
<td>Reservoir</td>
<td>553</td>
<td>675</td>
</tr>
<tr>
<td>Utilities</td>
<td>498</td>
<td>472</td>
</tr>
<tr>
<td>GeoQuest Internal Software</td>
<td>915</td>
<td>n/a</td>
</tr>
<tr>
<td>GeoFrame Developer Kit (GFDK)</td>
<td>581</td>
<td>548</td>
</tr>
</tbody>
</table>
NOTE: The sizes shown in the above table are those sizes shown by the Installtool. The values for each Component reflect the GeoFrame Kernel and common libraries necessary for that specific Component. Because of the common libraries, it is not valid to determine the exact sizes of an installation by simply adding the sizes shown.

You can approximate the size of your installation by using the size of the Components above, removing the GeoFrame Kernel and subtracting this from a full installation. For example if you wanted to install everything except for the GeoFrame Developer Kit (GFDK), take the size of the GeoFrame Developer Kit (GFDK), 581Mbytes minus the Kernel 358 Mbytes, which gives you approximately 223 Mbytes and subtract this from the full installation of 2.353 Gbytes, which gives 2.13 Gb.

Some sites are expected not to need all applications, as this includes both IESX and Charisma. Only Geoquest Internal processing sites would require the GeoQuest Internal Software.

The following table gives the current disk requirements for Geonet 5.2 and Oracle 8.1.6 Run time products. Please note that these figures are approximate.

<table>
<thead>
<tr>
<th>Component</th>
<th>Solaris Size (Mbytes)</th>
<th>SGI Size (Mbytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geonet 5.2 (190Mbytes for all)</td>
<td>190</td>
<td>190</td>
</tr>
<tr>
<td>Oracle 8.1.6 Run Time (Server)</td>
<td>540</td>
<td>550</td>
</tr>
<tr>
<td>Oracle 8.1.6 Forms &amp; Reports</td>
<td>889</td>
<td>900</td>
</tr>
</tbody>
</table>

Additional space is needed if a local site baseline is created. It is recommended you have a local baseline in order to store and manage site customizations. The amount of space required should not be great.

In a mixed Sun and SGI environment both architecture versions of Oracle binaries are usually needed to support the client-side functions. Only one (the native) is required for the server-side (database) functions. GeoFrame 4.0 provides a way to avoid the requirement of installing both Oracle binary sets. This is explained later in this document.
Machine Configuration

Many applications will not function adequately using a minimum hardware configuration because of graphics considerations and performance. See Preferred configuration.

Minimum

- Sun UltraSPARC 1 / 512Mb of RAM / 1 Gb of swap
- IRIX64 machine / 512Mb of RAM / 1 Gb of swap

Preferred

- Sun Ultra 80 Dual Processor / 1Gb of RAM / 2Gb of swap / Expert3D graphics
- SGI Octane Dual Processor / 1Gb of RAM / 2Gb of swap space / MXE Graphics

Memory availability

The commands ‘/etc/prtconf’ and ‘/usr/sbin/swap -s’ can be used for Sun and ‘/usr/sbin/swapon -s’ for SGI. These commands show the amount of main memory and the amount of swap space available on the current machine.

Required Operating System and Patches

Solaris

GeoFrame 4.0 requires Solaris 8 /January, 2001
Use ‘showrev -p’ to list the patches for your system.

The following is a list of the patches required of Solaris by GeoFrame:

- 108652-28 - X11 6.4.1: Xsun patch(this is a Finder 9.0 recommended patch)
- 108940-19 - Motif 2.1.1: Runtime library patch for Solaris 8
- 109543-13 - OpenGL 1.2.1: OpenGL Patch for Solaris 2.5.1/2.6/7/8 (32-bit)
- 109544-13 - OpenGL 1.2.1: OpenGL Patch for Solaris 7/8 (64-bit)
SGI

GeoFrame 4.0 requires IRIX 6.5.11 or later (64-bit is required). Your SGI server should report as IRIX64 when using the unix command ‘uname -s’.

Use ‘versions -b | grep patch’ to list the patches for your system.

GeoFrame does not require any additional patches for IRIX.

Compilers

The GeoFrame Developers Kit and Application builder requires Fortran and C compilers.

Solaris requires C/C++ version 5.2 and Fortran version 4.2.

SGI requires Fortran 77 and C/C++ version 7.2.

Geonet

GeoFrame 4.0 has been tested and verified using GUCE 6.0, Geonet 5.2

Flexlm 7.2e is required. GeoFrame 4.0 does not work with previous Flexlm versions. An existing license file for other GeoQuest products will work with this Flexlm version.

Oracle

GeoFrame 4.0 has been tested using Oracle 8.1.6.
Pre-installation tasks performed by root

Set Up UNIX user accounts

It is recommended that you create a UNIX login account to be used for the installation of GeoFrame. While this is not mandatory, it allows a division of administration functions (and their operating environments) between the Oracle Database Administration and general GeoFrame software configuration and administration. The GeoFrame user should be a member of the `dba' group, specified on the machine that will be running the database instance.

For the correct Geonet Integration of GeoFrame, the files under GN_DIR should belong to the same group as GeoFrame files and have write permission granted to the GeoFrame installer account.

For the C-Tree Server process, if you select the SITE mode, it is required to assign a user account which will start the Site C-Tree Server process and hold the special files for that process. This username should exist and belong to the same unix group as all the users that will connect to that C-Tree server process.

Solaris

You can use the system administration tool `sysadm' to create this user account. You can also manually edit the `/etc/passwd' file on Solaris using your editor of choice. If you manually edit the `/etc/passwd' file you must run the `pwconv' command to update the `/etc/shadow' file with the information from `/etc/passwd'. The GeoFrame account should be a member of the `dba' group, and the C-Tree Server owner (for SITE mode) should be a member of the same group as the GeoFrame users as discussed above.

SGI

You can use the system administration tool from the "toolchest" GUI to create this user account. You can also manually edit the `passwd' and `group' files on the SGI using your editor of choice. The GeoFrame account should be a member of the `dba' group, and the C-Tree Server owner (for SITE mode) should be a member of the same group as the GeoFrame users as discussed above.
Planning for installation

If a previous version, for example GeoFrame 3.8.1, is already installed:

- Make sure that you are familiar with the Oracle instance configuration. See Oracle Installation guide for details.
- See the Section "Installing a GeoFrame 4.0 catalog if GeoFrame 3.8.1 already exists", described later in this document.

Planning for Combination SUN and SGI installation

There is a parameter requiring special attention for cross platform execution. It is called "GF_SGI_PATH" during the SUN installation and "GF_SUN_PATH" during the SGI installation. This should be correctly defined during the installation; also check the GeoFrame_Runtime.csh script in $GF_PATH after the installation is completed.

Please see the discussion on Installing Solaris and SGI baselines on the same network, described later in this document.

Solaris

"GF_SGI_PATH" should be set to the $GF_PATH of the SGI baseline.

A soft link should be created in the SUN $GF_PATH which links "sgi" to the SGI $GF_PATH. Example of the soft link:

    sgi -> /some/path/geoframe_40_sgi

Example of the required environment in the GeoFrame_Runtime.csh script for Sun:

    setenv GF_SGI_PATH "/some/path/geoframe_40_sgi"

SGI

"GF_SUN_PATH" should be set to the $GF_PATH of the SUN baseline.
A soft link should be created in the SGI $GF_PATH which links "sun" to the SUN $GF_PATH. Example of the soft link:

```
sun -> /some/path/geoframe_40_sun
```

Example of the required environment in the GeoFrame_Runtime.csh script for SGI:

```
setenv GF_SUN_PATH "/some/path/geoframe_40_sun"
```

## Configuring GeoFrame for Multiple Oracle databases

To configure a GeoFrame installation to use multiple Oracle databases, see the section on Configuring for Multiple Oracle databases in "Installation scenarios".

### Prerequisite Tasks and Checks

- Use a regular xterm to install GeoFrame4.0, not a previous GeoFrame Xterm.
- Check that the ‘umask’ is set to 022. To verify the current setting for umask, use the command ‘umask’ without any parameters in a xterm.
- Check that the installer user account has write permission in the directories where you are planning to install the software, including the GN_DIR/bin and GN_DIR/config directories and files in Geonet. Login as the GeoFrame installer and try to create and delete a **non-existing** file in those directories.
- Check that you do not have any variables set that are for an existing Oracle or GeoFrame installation. Use the command ‘env | grep ORA’ and ‘env | grep GF’ and check that you have no variables set in your environment. If you do, remove them from the .cshrc or .login scripts and logout and login again to ensure that you have a clean environment.
- Check that disk space is available for the planned installation including space for new tablespaces.
- Check that you do not have any aliases set in your environment. To check this, use the command ‘alias’. Remove any alias such as rm, du, etc. After you remove them, logout and login again to ensure that you have a clean environment.
- Read the appropriate sections on planning your installation and complete the Installation Planning Checklist.
• Make sure that you have the media containing your GeoFrame software. GeoFrame software is distributed on a single CD-ROM for each supported platform. The media label lists the GeoFrame baseline version contained on the CD, such as gf40.

• Make sure you have your product licenses. These are normally supplied with your GeoFrame software
Installing GeoFrame

Below is a step-by-step instruction of installing GeoFrame. Please see the following Guides for additional information.

- Installation Tool instructions
- Geonet Installation guide
- Run Time Oracle installation

The **GeoFrame Installation Tool** executable is called from the "installtool" script which is located in the top level directory ‘gf40’ of the CD-ROM.

Performing an installation:

1. Insert the GeoFrame 4.0 CD-ROM for the corresponding Operating System in the CDROM drive. This will normally be mounted automatically. If problems are experienced with mounting the CD, see your local system administrator for assistance.

2. Log into the **GeoFrame** Manager account from the login window. If this is remote installation set the DISPLAY environment variable:

   ```bash
   setenv DISPLAY host_name:0.0
   ```

   Note: If this is a remote installation, make sure you type: `xhost +` before connecting to the remote node.

3. Verify the following in the terminal from which you are going to start the installation:

   - Check that there is enough free space on the selected partition if you are planning to install Oracle software, for example:
     ```bash
     # df -k /directory/to/install
     ```
   - Make sure that the GeoFrame owner has write permission on the partition that you designated for the Oracle tables. For example:
     ```bash
     > touch /directory/to/install/temp
     > rm /directory/to/install/temp
     ```
   - Check that GeoFrame owner “umask” is set to 022. Use command `umask` to check and to modify the umask, for example:
     ```bash
     > umask 022
     ```
— Check that environmental variables ORACLE_HOME, TNS_ADMIN, TWO_TASK or ORACLE_SID are not set before starting the installation, for example:

```bash
> env | grep ORACLE_HOME
> env | grep TNS_ADMIN
> env | grep TWO_TASK
> env | grep ORACLE_SID
```

These commands should not return anything when the corresponding variables are not set. If these variables are set, please unset them. Having them set may produce an adverse effect on the outcome of the installation.

— Check that you do not have any aliases such as rm, du, ls, mv, cp set in your environment. Use the commands `alias` and `unalias` to check and remove aliases from the environment.

4 Start the Installation Tool by typing the following at the unix prompt.

```
unix> cd /cdrom/gf40
unix> ./installtool
```

⚠️ **NOTE:** You must enter the startup command as "./installtool". Otherwise you will receive an error message indicating that it is "Unable to find binaries for the installtool".

A log file named ~/.InstallLog_time_date will be created.

Part of the log file is displayed in the "Messages" text box of the Installtool interface.

⚠️ **NOTE:** For details about the Installation Tool, see the GeoQuest Installation Tool document.

After several minutes of analyzing the installation information found on the CD-ROM, the Installation Tool displays the "Welcome Phase" menu:

**Welcome Phase**

5 Select Next to continue.
The Mode of Installation Phase

6 Two options are presented: Default Mode or Custom Mode. The recommended method is Custom Mode, then select Next to continue.

The Default Mode allows for a minimum amount of choices and parameters. It displays fewer dialogs which allow you to select the components you wish to install and the location of these directories and files only. The Installation Tool then uses a predefined set of Oracle and Geoframe parameters:

- TWO_TASK = geoquest1
- ORACLE_HOME = $HOME/geoframe_40_xxx (where xxx = sun or sgi)
- TNS_ADMIN = $HOME/network/admin
- LM_LICENSE_FILE = /usr/local/flexlm/licenses/license.dat
- CHARISMA HOME = $HOME/charisma
- BASELINE_ACC = GF4_0

NOTE: Currently the Default Mode of installation is not recommended unless all the predefined parameters are valid, which is almost never the case.

The Custom Mode allows you to select the components you wish to install and the locations of these directories and file; it also allows user selection of all available database parameters, tablespace sizes and file locations. It permits user control of the "init.ora" parameters by selecting Small/Medium/Large sizes of databases as defined in the Administration Guide.

The Feature Key File Phase

7 If you do not have a Feature Key file for GeoFrame yet, ignore this phase and select Next.

The Feature Key File is a file supplied by GeoQuest that contains the new license keys for the products you are installing (the license key file).

The Installation Tool searches for this file in several default locations ( $HOME or /usr/local/flexlm/licenses). You may override the location that is displayed, by selecting the Browse button. It is convenient to start with this file in $HOME.

A warning message will be displayed if a key file can not be found. The user can reply Yes to continue and ignore the warning or No and find a proper key file.
If a license file is found, the installtool selects the components to install based on the file contents (the licensed Features). You can override this selection in the next phase.

Other purpose for this file is to provide a way to automatically update an existing FlexLM license with its contents. This is an optional step during the installation.

**Select Components to Install**

8 Select the components to install by selecting the button to the left of the component name and select **Next** when you are ready to continue.

In a later menu you will be requested to enter the parameters based on your Component selections.

**NOTE:** In general, it is recommended to install everything. The exception is the GeoFrame Developer Kit which can be safely deselected. If this component is selected, you have to provide a path to the required compilers.

**Select Locations to Install Components Phase**

9 Select the Location to Install GeoFrame and click on **Next** to continue.

This menu shows you the GeoFrame product for installation, the required and available space, and the default location, which is $HOME.

You may change the Selected Location with the "Browse..." button. If the location does not exist, the icon to the right of the location will appear red. By clicking on the icon, the directory is created and the icon becomes green. This location is known to GeoFrame as $GF_HOME.

An error message is displayed if there is insufficient space in the directory for the selected components.

**Installation Steps Phase**

10 Select the steps you wish to perform and select **Next** when you are ready to continue.

The **Installation Steps** phase is only applicable in the **Custom** installation mode. Some of the components that you have selected for installation require additional installation steps. You can select or deselect any of these steps at this time.
If you are using the **Default** installation mode, the **Installation Steps** menu is not displayed. Listed below are the pre-defined **Installation Steps** that would be executed for a **Default** installation of GeoFrame:

- Install GeoFrame Database Schema
- Setup FlexLM startup files (requires root)
- Integrate GeoFrame with Geonet

### Installation Steps Description

Listed below is a brief description of the installation steps available for the **Custom** installation mode:

**Create an Oracle Instance for GeoFrame** - Use this option to create an Oracle instance using an existing Oracle installation. If the Oracle instance is already running, you do not want to select this option.

**Install GeoFrame Database Schema** - Selecting this will import the GeoFrame schema into the Oracle database. This is often called the **catalog** or BASELINE ACCOUNT. This option is usually selected for a new **GeoFrame** installation.

**Copy DSL paths from previous catalog to new catalog** - Selecting this you will be expected to enter the name of a previous catalog and the password for this catalog. This should only be selected if **Install GeoFrame Database Schema** was selected and if you had created a GeoFrame catalog in the same Oracle instance that you are using now. Do a `gf_accounts` command from a previous GeoFrame Xterm, to verify if the source catalog does exist and use sqlplus `<catalog_name/passwd>` to verify this entry.

**Install Uniras** - This option installs the plotting Software for **Charisma** scaled hardcopy. Select this option for GeoFrame 4.0. It uses a new version of UNIRAS software.

**Install GM_SITE_DIR** - This option is usually selected for a **Charisma** installation. Do not overwrite previous GM_SITE_DIR location.

**Copy the new License Keys to license file** [Requires root Password] - This optional step merges the new FlexLM license keys contained in the new Feature Key file into an existing FlexLM license key file. When you select this option, put the New keys license file in `$HOME` and set the `FLEXLM_LICENSE_FILE` pointing to the currently active license.dat file. Whether you select this option or not, the `LM_LICENSE_FILE` should point to the currently active license file if that path is mounted from the client machines, or set in the form of `<port>@<host>`.
Setup FlexLM startup files [Requires root Password] - This option will install the scripts necessary to automatically stop and start the FlexLM daemons when the server is rebooted. If FlexLM is already running, you may not want to select this. This option is usually selected for a new GeoFrame installation.

Setup Site C-Tree Server boot time startup [Requires root password] - This optional step is included to create the required scripts to start a SITE C-Tree Server at boot time. Normally this step is not required unless special circumstances make it more convenient to set the Ctree Server operation in SITE mode. The default mode is LOCAL.

If this option is selected, the installtool will request additional input to complete the Ctree Server configuration in SITE mode.

Refer to the CTREE SERVER section later in this document for more details about the Ctree Server configuration.

This option configures the startup of a Ctree Server process in SITE mode to be run on the same host as the GeoFrame installation. You can copy the created scripts to other server if required.

Integrate GeoFrame with Geonet - This option will copy and integrate new launch scripts for Geonet to be used by GeoFrame4.0. A new GeoFrame 4.0 button will be added to Geonet. For this step, make sure that the GeoFrame owner has write permission granted to the files in GN_DIR/bin and GN_DIR/config.

Root Password Phase

To continue, enter the Root password and you will then be asked to re-enter it for verification. If you choose not to enter the Root password, the Installation Tool will write a file named GeoFrame4_0_root_time_date.csh in $HOME. This script can be executed afterwards by anyone with root permissions in order to complete the installation.

Several steps that you selected earlier may require a Root password. Those steps may be any one or all of the following:

- Setup FlexLM startup files
- Copy the new license keys to license file.
- Setup Site C-Tree Server boot time startup.

11 Select Next when you are ready to continue.
Installation Parameters Phase

12 Enter your parameter values and select Next when you are ready to continue. The Installation Parameters menu only appears for Custom installation mode.

NOTE: You are better to not use Default installation mode: The predetermined conditions are not generally useful.

The parameters listed in the Parameter area of the screen are divided into the following groups:

- Oracle Parameters
- Application Parameters
- General Parameters

This menu shows the list of parameters that are used by the installation procedures to be executed. The displayed parameter fields are: the Parameter Description, the Environment variable name and the assigned value, which usually appears with a default or example value.

The icon to the right of a location indicates whether the location does exist (green folder icon) or not (red icon). The "Browse..." button at the right opens a file selector window. The red icon may be selected to create the directory after selecting the correct location; then it becomes green if the directory was successfully created.

During the GeoFrame 4.0 installation you may want to set ORACLE_HOME the same ast GF_PATH e.g., $<GeoFrame selected location>/geoframe_40_sun or _sgi. This is so, because some Oracle programs and libraries have been included in the baseline. One advantage of this approach is that you only need to install Oracle on your database server and not for each supported platform. Also, the Oracle server tree does not need to be available to the clients, only three SQL*Net configuration files.

- tnsnames.ora
- tnsnav.ora
- sqlnet.ora

The ‘network’ file listener.ora is relevant to the Oracle server processes and does not need to be available to the clients.

This is a list of the Oracle 8.1.6 files that are included in the GeoFrame 4.0 baseline:
bin/exp
bin/imp
bin/sqlldr
bin/sqlplus
bin/tnsping

On Solaris:
lib/libclntsh.so.8.0
lib/libwtc8.so

On Irix:
lib/libclntshcdk.so.8.0
lib/libclntsh.so.8.0
lib64/libclntsh.so.8.0
lib64/libwtc8.so

on both:

sqlplus/mesg/*.msg
sqlplus/mesg/*.msb
rdbms/mesg/*.msg
rdbms/mesg/*.msb
network/mesg/*.msg
network/mesg/*.msb

New parameters for C-Tree Server configuration in SITE mode (Optional step)

The additional input required to configure the Site C-Tree Server startup if you selected that optional step is described below:

SITE_CTREE_USER which is the Site C-Tree Server Owner. This is a unix account that should exist and is selected as the administrative user for the Ctree-Server functions and archives. The C-Tree Server administrator account should be a member of the same unix group as the GeoFrame users that will be using this C-Tree Server so that it has write access to all the DSL’s for the users.

This group is usually different from the GeoFrame or Oracle owner group.

SITE_CTREE_NAME. This is the particular name that you can give to the Site C-Tree Server. The default name is FAIRCOMS. The name should be a unique, one-word, upper-case ascii name with no special characters. Select a name shorter than 15 characters.
SITE_CTREE_HOME. This is the Site C-Tree Server Base Directory, which should be created on a LOCAL partition and owned by SITE_CTREE_USER. This directory is going to hold the C-Tree Server Files, which are special system files to keep track of the Server status, error conditions and the required information for automatic recover and restart the C-Tree Server process. You can create this directory either as the GeoFrame installer, as the Site C-Tree Server Owner (SITE_CTREE_USER) or as root; it will be automatically changed to be owned by the Site C-Tree Server owner, but make sure that the directory is created on a local partition, otherwise this installation step will fail to complete.

The recommended practice is to create the SITE_CTREE_HOME directory on a file system structure similar to the one for Flexlm. For example, you can create this directory: /usr/local/ctree_server. This directory does not require to be exported to the clients.

The required space for this directory is very small (less than 10 MB), but the files that the C-Tree Server process will create there are very important in Site mode!

BULK_SERVER_ADDRESS. This environment variable is not required to complete the configuration of Site C-Tree Server boot startup scripts, but it is mentioned here to clarify the required syntax in case the C-Tree Server is used in Site mode. In this case it should be set as

"SITE:<servername>@<host>"

in which <servername> is the value given to SITE_CTREE_NAME, which normally is FAIRCOMS, and <host> is the name of the machine that will run the Site C-Tree Server process.

**Complete Parameter list**

The complete list of installation parameters for GeoFrame 4.0 is described below.

Not all of these parameters will be displayed, only the required values, depending on your selections in previous steps of the installation.

Some of the default values as they appear in the installtool dialog boxes, should be considered as mere examples and not literally as the recommended values.

Some of them include the Oracle Service name, instance identifier (TWO_TASK, ORACLE_SID), new and previous catalog names/passwords, etc.
If you plan to start or continue using the Update GeoFrame Schema (catalog) option, it is recommended to use a catalog name which is NOT tied to the GeoFrame Version, in order to avoid confusion.

## Installation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORACLE INST_DIR</td>
<td>$HOME/geoframe_40_sun or _sgi</td>
<td>Path to the existing Oracle 8 installation (when you want to add a second Oracle instance)</td>
</tr>
<tr>
<td>ORACLE_HOME directory</td>
<td>$HOME/geoframe_40_sun or _sgi</td>
<td>$ORACLE_HOME path, which may be set to point to $GF_PATH.</td>
</tr>
<tr>
<td>Oracle Network Admin Directory (TNSADMIN)</td>
<td>$HOME/network/admin</td>
<td>Directory containing the Oracle SQL*Net configuration files.</td>
</tr>
<tr>
<td>Default Oracle Datafile directory (DB_DEF_DISK1_DIR)</td>
<td>$HOME/dbs</td>
<td>Variable DB_DEF_DISK1_DIR will be used in other parameters as a default directory for some Oracle data files.</td>
</tr>
<tr>
<td>Default Oracle second Datafile directory (DB_DEF_DISK2_DIR)</td>
<td>$DB_DEF_DISK1_DIR</td>
<td>Variable DB_DEF_DISK2_DIR will be used in other parameters as a default directory for some Oracle data files. If possible, set it to a different disk to improve efficiency.</td>
</tr>
<tr>
<td>Oracle Log File Directory</td>
<td>$DB_DEF_DISK1_DIR</td>
<td>Directory containing the Oracle Log files</td>
</tr>
<tr>
<td>Oracle System Tablespace Directory</td>
<td>$DB_DEF_DISK1_DIR</td>
<td>Directory containing the Oracle SYSTEM tablespace</td>
</tr>
<tr>
<td>Oracle System Tablespace Size</td>
<td>150</td>
<td>Oracle SYSTEM tablespace size in Mb.</td>
</tr>
<tr>
<td>Oracle RollBack Directory</td>
<td>$DB_DEF_DISK1_DIR</td>
<td>Directory containing the Oracle Rollbacks</td>
</tr>
<tr>
<td>Oracle RollBack Size</td>
<td>200</td>
<td>Oracle Rollback size in Mb.</td>
</tr>
<tr>
<td>Oracle DATA Tablespace Directory</td>
<td>$DB_DEF_DISK2_DIR</td>
<td>Directory containing the Oracle Data tablespace.</td>
</tr>
<tr>
<td>Oracle Data Tablespace Size</td>
<td>200</td>
<td>Oracle Data Tablespace size in Mb.</td>
</tr>
<tr>
<td>Oracle INDEX Tablespace Directory</td>
<td>$DB_DEF_DISK1_DIR</td>
<td>Directory containing the Oracle Index Tablespace</td>
</tr>
<tr>
<td>Oracle Index Tablespace size</td>
<td>150</td>
<td>Oracle Index Tablespace size in Mb.</td>
</tr>
<tr>
<td>Oracle TEMP Tablespace Directory</td>
<td>$DB_DEF_DISK1_DIR</td>
<td>Directory containing the Oracle TEMP Tablespace</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Oracle TEMP Tablespace size</td>
<td>50</td>
<td>Oracle TEMP Tablespace size in Mb.</td>
</tr>
<tr>
<td>Oracle Service Name</td>
<td>geoquest1</td>
<td>TNS service name (TWO_TASK) of the instance.</td>
</tr>
<tr>
<td>Oracle instance identifier</td>
<td>ora8gq</td>
<td>ORACLE_SID of the instance.</td>
</tr>
<tr>
<td>Oracle sys account password</td>
<td>change_on_install</td>
<td>Password for SYS Oracle account. This Password is used to connect to the SYS account to import the catalog. Do not change this password during the installation.</td>
</tr>
<tr>
<td>Oracle system Account Password</td>
<td>manager</td>
<td>Password for SYSTEM Oracle account. This Password is used to connect to the SYSTEM account to import the catalog.</td>
</tr>
<tr>
<td>Index Tablespace name</td>
<td>TSINDEX</td>
<td>Name of the Index tablespace used by GeoFrame. This tablespace will be used to store the catalog.</td>
</tr>
<tr>
<td>Data Tablespace name</td>
<td>TSDATA</td>
<td>Name of the Data tablespace used by GeoFrame. This tablespace will be used to store the catalog.</td>
</tr>
<tr>
<td>Baseline account name</td>
<td>GF4_0 (change)</td>
<td>Name for a new GeoFrame catalog.</td>
</tr>
<tr>
<td>Baseline account password</td>
<td>GF4_0 (change)</td>
<td>Password to connect to the new GeoFrame catalog.</td>
</tr>
<tr>
<td>GeoFrame system password</td>
<td>gf_sys</td>
<td>Password for the Oracle user account &quot;gf_sys&quot; which is the Geoframe system account</td>
</tr>
<tr>
<td>Copy DSLs From TWO_TASK</td>
<td>geoquest1</td>
<td>Oracle service name of instance with a catalog to copy DSLs from. This applies when creating a new catalog when other exists already.</td>
</tr>
<tr>
<td>Copy DSLs From Account Name</td>
<td>GF3_8</td>
<td>Previous Catalog from which you want to copy project Disk Storage Locations.</td>
</tr>
<tr>
<td>Copy DSLs From Account Password</td>
<td>GF3_8</td>
<td>Oracle Password of the existing catalog from which you want to copy project Disk Storage Locations.</td>
</tr>
<tr>
<td>Application Parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GeoFrame home directory</td>
<td>$HOME</td>
<td>This value is set previously when you selected the Location to install. Normally it should not be changed from the displayed value.</td>
</tr>
<tr>
<td>Charisma / InDepth-Uniras home directory (UNIHOME)</td>
<td>$HOME/charisma/uniras</td>
<td>The directory containing the Uniras software used by Charisma Scaled hardcopy in which the new Uniras software will be installed. Specify a new location for GeoFrame 4.0.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Charisma / InDepth Site Directory</td>
<td>$HOME/charisma/client</td>
<td>The directory for site specific Charisma files, such as the plotting scripts, plotter definitions and mapping application parameters. Specify a separate location from a GeoFrame 3.x installation or the directory into which the files are to be installed if this is a new installation.</td>
</tr>
<tr>
<td>Charisma / InDepth project directory (DB_GMHOME_DIR)</td>
<td>$HOME/charisma/projects. Note, the DB_GMHOME_DIR environmental is only used for installation and is stored in the database. It is not a runtime environmental setting.</td>
<td>The directory used by Charisma for project information other than bulk data. In GeoFrame 4.0 this value is used during installation only. You can use the same path of an existing GeoFrame 3.x Charisma installation. Each GeoFrame 4.0 user needs to define a DSL for his/her Charisma Projects.</td>
</tr>
<tr>
<td>Site C-Tree Server Owner</td>
<td>Unix user designed to be the owner of the C-Tree Server system files in Site mode.</td>
<td></td>
</tr>
<tr>
<td>Name of Site C-Tree Server</td>
<td>FAIRCOMS</td>
<td>C-Tree Server name in SITE mode.</td>
</tr>
<tr>
<td>Site C-Tree Server Base Directory</td>
<td>Site C-Tree Home. This location contains the SITE C-Tree Server system/status files.</td>
<td></td>
</tr>
<tr>
<td>BULK_SERVER_ADDRESS</td>
<td>LOCAL:</td>
<td>Variable defining the mode to use the C-Tree Server from the GeoFrame applications. The default is the value for local mode. For SITE mode, it should be set to SITE:&lt;servername&gt;@&lt;host&gt;, where &lt;servername&gt; is usually FAIRCOMS and &lt;host&gt; is the machine running the Site C-Tree server.</td>
</tr>
<tr>
<td>Allow private project disk</td>
<td>off</td>
<td>Select this &quot;on&quot; for the option of &quot;Private Disks&quot; on the Proman disks menu. Private disks allows users to use any disk which is available for their use. This is not suggested for structured organizations. Also it does not work for Charisma extensions.</td>
</tr>
<tr>
<td>Geonet Home Directory</td>
<td>/usr/local/Geonet/gn5c02_0/solaris</td>
<td>This is the GN_DIR location to be used to perform the Geonet integration.</td>
</tr>
<tr>
<td>Charisma scratch directory</td>
<td>$HOME/charisma/scratch</td>
<td>The directory to be used by Charisma as a scratch directory for saving CGM files and for backup and restore operations. It can be the same location of an existing GeoFrame 3.x installation. Any GeoFrame user should be allowed to write in this directory.</td>
</tr>
<tr>
<td>Charisma/InDepth product</td>
<td>InDepth</td>
<td>It seems that this value is irrelevant now.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GF_HARDCOPY_DIR</td>
<td>$GF_HOME/hardcopy_files</td>
<td>Path to use for GeoFrame scratch plot area. Any GeoFrame user should be allowed to write in this directory.</td>
</tr>
<tr>
<td>Directory containing compilers</td>
<td>/opt/SUNWpro/SC5.2 - Solaris</td>
<td>This is the path to the installed compilers. It is used by the Installtool to build baseline and also used by Application builder and by GFDK when selected. The required compilers version for GeoFrame 4.0 is SC5.2</td>
</tr>
<tr>
<td>General Parameters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLIS data file path</td>
<td>$GF_HOME/dlis_files</td>
<td>This is the path to create the test dlis files which can be used to test installation. Use &quot;test_archive.dlis&quot; for a dlis load and &quot;test_backup.dlis&quot; for a project restore.</td>
</tr>
<tr>
<td>FlexLM License File</td>
<td>/usr/local/flexlm/licenses/license.dat</td>
<td>This environment will be put in the $GF_PATH/GeoFrame_Runtime.csh script. This should be the absolute path to the currently active license.dat file if mounted from the clients, or set in the form of &lt;port&gt;@&lt;host&gt;.</td>
</tr>
<tr>
<td>FlexLM bin directory</td>
<td>/usr/local/flexlm/bin</td>
<td>This is used to specify the FlexLM bin directory to the boot time startup script.</td>
</tr>
<tr>
<td>FlexLM license file to be updated</td>
<td>/usr/local/flexlm/licenses/license.dat</td>
<td>License file to be updated if &quot;Copy the new License Keys to license file&quot; step is selected. This is used to specify the FlexLM license file to the boot time startup script, and should be the absolute path to the currently active license.dat file.</td>
</tr>
<tr>
<td>FlexLM New License Keys file</td>
<td>$HOME/license.dat</td>
<td>This is the path to your License File with the New Features to be copied to the &quot;FlexLM license file to be updated&quot;. It is better to put this temporary file in $HOME.</td>
</tr>
<tr>
<td>Openwindows home directory</td>
<td>/usr/openwin</td>
<td>Solaris only, this is the home directory for Openwindows libraries.</td>
</tr>
<tr>
<td>Geoframe Environment Umask</td>
<td>002</td>
<td>This is the umask setting which will be used by Geoframe to create project directories and files.</td>
</tr>
<tr>
<td>Motif home directory</td>
<td>/usr/dt/</td>
<td>Solaris only, this is the home directory for Motif libraries.</td>
</tr>
<tr>
<td>Is GF4.0 for Solaris/SGI installed</td>
<td>NO</td>
<td>Enter YES or NO, then enter the correct path below.</td>
</tr>
</tbody>
</table>
Ready to Extract Files Phase

If you are ready to proceed with the file extraction process, select Next.

Until now you have been entering parameters and making selections. You have given the Installation Tool all the necessary information to perform the installation for the products shown. If you wish to make changes, select Back to return to the appropriate dialog.

NOTE: When selecting Back over the Parameter Phase and backwards, some of the parameter values already entered are reset to the default value, so re-check all the parameters again.

The installtool will now uncompress and untar the Install kit and populate the directories and files for the products selected.

Re-Extract Previously Extracted Archives- Normally you do not want this toggle turned on. This toggle option allows the user to un-compress and un-tar those archives that have already been done. During the Extraction process, the restart capability keeps track of which archives have been completed and which are not. If you quit in the middle of the extraction, if the restart file is created, you can restart and leave this option toggled off and it will only un-pack those files that have not been done yet. However, if there is any doubt because there was any file deletion or
corruption, it would be the best "to un-pack" all the files again. The recommended method is to delete the failed extracted files completely including the base directory and start the installtool again without using the `-restart` option.

You can recover a GeoFrame installation by re-extracting all the baseline files only, as long as the Oracle datafiles are intact. For this purpose, you can run the `Installtool` and **deselect all the installation steps**. Then proceed to the extraction phase, having previously deleted any remains of the damaged or suspected baseline directory, except may be the GeoFrame_Runtime.csh script. Even if you have deselected all the installation steps, you have to run the Execution and Validation phase, which complete some required procedures. You only need to keep track of the parameters used in the original installation, which normally can be found in the script `GeoFrame_Runtime.csh` located in `GF_PATH` or in `GN_DIR/geoframe40.db`.

**Extracting Files**

14 The **Extracting Files** dialog opens, showing a progress slider bar.

This phase will take a some time, depending on what products and parameters you have selected.

**Examine Prerequisites Phase**

15 Select **Next** to continue.

The **Examine Prerequisites** phase checks to see if shared memory is set up, if Oracle binaries are available, and if a connection to an existing Oracle instance is successful.

It also verifies that the Fortran compiler (required for Application Builder and GeoFrame Developer Kit) is installed. You may ignore the Warning if Fortran Compiler is not found and continue, if you do not plan to use these applications.

**Ready to Execute Root Procedures Phase**

This phase is run if there is a selected step requiring of the root password in order to execute.

**Ready to Execute Procedures Phase**
Select **Next** to execute the installation procedures for the components you selected. If you would like to do this later, select **Quit**. If you choose to **Quit**, you can return to this step by running the **Installation Tool** with the **-restart** option.

**Executing Procedures**

Installtool executes the steps selected in the Installation Steps phase. Review the logfile displayed in the **Messages** area to verify that there are no errors, even if it tells you that the procedures are executed successfully.

To perform this operation, select **Next**.

During this operation, the **Executing Procedures** dialog opens, once again providing a progress slider bar.

**Validating Files**

A **Validating Files** dialog opens, displaying a progress slider bar.

This validation makes a comparative checksum of the installed files against a known set of checksums on the installation media. This is a guard against any file corruption.

**Installation Completed**

The installation is finished and a dialog box opens indicating completion.

You may select the **Quit and Run** button, which terminates the **Installation Tool** and runs **GeoFrame**.

If you do not want to run **GeoFrame** at this time, press **Quit** to exit the **Installation Tool**.