Oracle® Internet File System

Setup and Administration Guide

Release 1.1

September 2000

Part No. A81197-05

ORACLE®

Setup and Administration Guide, Release 1.1

Part No. A81197-05

Primary Author: Aimee Nicklin

Contributing Authors: Jeff Backes, Dennise Brown, Edmond Chui, Dennis Dawson, Vijay Devadhar, Kevin Gillett, Tom Grant, Mei Hong, Francine Hyman, Vasant Kumar, Tim Lockwood, Dave Long, David Pitfield, Karthik Raju, Syliva Perez, Tina Seto, David Tom

Contents

2 Starting Oracle *i*FS

3 Using Oracle *i*FS Manager

4 Task 1: Set Up the Folder Hierarchy

5 Task 2: Add Users and Groups

6 Task 3: Load Files into Oracle *i*FS

7 Task 4: Set Up Security

8 Using Server Manager to Start and Stop Servers

9 Configuration and Back Up

10 Using Oracle *i*FS Manager for Development Tasks

11 Oracle *i*FS Repository Views

12 What Your Users Need to Know

13 Oracle *i*FS Log Files and Troubleshooting Information

A Command Line Utilities Reference

B Secondary Properties

Index

Send Us Your Comments

Oracle Internet File System Setup and Administration Guide, Release 1.1

Part No. A81197-05

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

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

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

- E-mail: ifsdocteam@us.oracle.com
- FAX - 650.506.7104 Attn: Documentation Manager for Oracle Internet File System
- Postal service:
  Oracle Corporation
  Oracle Internet File System, Attn: Documentation Manager
  500 Oracle Parkway, Mailstop 5op4
  Redwood Shores, CA 94065
  USA

If you would like a reply, please give your name, address, and telephone number below.

____

____

____

If you have problems with the software, please contact your local Oracle World Wide Support Center.

Preface

Oracle Internet File System (Oracle *i*FS) is a file system and development platform that runs as part of the Oracle8*i* database. From the perspective of an end user, Oracle *i*FS functions similarly to a standard file server, organizing files into hierarchies of folders.

Oracle *i*FS combines file, Web, e-mail, and database servers into a single server application and a single repository. Data previously stored on separate machines, managed by separate server applications, can be integrated into a single repository. Users and applications can access the contents of Oracle *i*FS through several network protocols, such as FTP, Windows (SMB), and HTTP, as well as through relational database operations. As an administrator, you can secure, back up, restore, and monitor all of this data using different administration tools.

Intended Audience

This manual is intended for system administrators. It provides an introduction to Oracle *i*FS and describes the configuration, customization, and management tasks you will perform as an Oracle *i*FS server administrator.

Skills Required to Administer Oracle *i*FS

You need the following skills to perform basic administrative functions with Oracle *i*FS:

- Experience administering systems running the networking protocols used in Oracle *i*FS. These include Windows (SMB) / CIFS (the protocol for Windows networks), FTP, HTTP, IMAP4, and SMTP.

- Familiarity with Internet technologies. Since Oracle *i*FS serves web pages representing the file system, relies on XML for system configuration data, and uses Java as the language of server-side application development, familiarity with these technologies is essential.
- Experience administering an Oracle database. The administrator needs to be acquainted with relational database principles as well as the tools used to administer an Oracle database.
- Familiarity with the Java Runtime Environment (JRE) and the Java Developer's Kit (JDK).

Structure of this Guide

This manual contains thirteen chapters and two appendices:

Chapter 1	Getting Started with Oracle iFS	Introduces Oracle *i*FS, the administration tools, out-of-the-box and customized setup, and maintenance issues.
Chapter 2	Starting Oracle iFS	Describes the necessary steps to starting Oracle *i*FS.
Chapter 3	Using Oracle iFS Manager	Describes using Oracle *i*FS Manager, the tool used to administer Oracle *i*FS as an extension of Oracle Enterprise Manager.
Chapter 4	Task 1: Set Up the Folder Hierarchy	Describes the folder hierarchy in Oracle *i*FS, how to plan your folder hierarchy, and how to use the administration tools to create and define security for folders.
Chapter 5	Task 2: Add Users and Groups	Describes users and groups in Oracle *i*FS, and the administration tasks used to create, modify, and delete them.
Chapter 6	Task 3: Load Files into Oracle iFS	Describes the five ways to load files into Oracle *i*FS, how to assign access permissions, and how to upload files.
Chapter 7	Task 4: Set Up Security	Describes user and file security, including Access Control Lists (ACLs), and how to define them.

Chapter 8	Using Server Manager to Start and Stop Servers	Describes starting and stopping the servers using Server Manager and working with the protocol servers.
Chapter 9	Configuration and Back Up	Describes configuration and maintenance functions.
Chapter 10	Using Oracle iFS Manager for Development Tasks	Describes the various development tasks which can be performed using Oracle *i*FS Manager, such as creating class objects and value defaults.
Chapter 11	Oracle iFS Repository Views	Shows how to use the Oracle *i*FS repository views to administer Oracle *i*FS.
Chapter 12	What Your Users Need to Know	Describes basic information your users need to get started using Oracle *i*FS.
Chapter 13	Oracle iFS Log Files and Troubleshooting Information	Describes troubleshooting information for problem solving.
Appendix A	Command Line Utilities Reference	Describes using the Command Line Utilities for administering Oracle *i*FS.
Appendix B	Secondary Properties	Describes Oracle *i*FS properties and provides example of settings.

Related Documents

For more information on Oracle *i*FS, see the following manuals which are included with Oracle *i*FS:

- *Oracle Internet File System Installation Guide*
- *Oracle Internet File System Quick Tour*
- *Oracle Internet File System User's Guide*
- *Oracle Internet File System Developer's Guide*
- *Oracle Internet File System XML Reference*

For more information on the Oracle8*i* database, see the documentation for Oracle8*i*.

The following developer documentation is available in the Documentation section of the Oracle *i*FS listing on OTN (Oracle Technology Network):

Oracle Internet File System Javadoc	Describes the packages, classes, and methods of the Oracle *i*FS API.
Oracle Internet File System Class Reference	Provides a listing of the class hierarchy and describes the attributes of the Java classes.

1

Getting Started with Oracle *i*FS

Oracle *i*FS provides a mechanism for creating, storing, and managing various types of information in a common repository for users to access and update. This chapter gives you a brief look at Oracle *i*FS, how it is used out-of-the-box, customization features, the various administration tools, and the tasks an administrator needs to perform to set up, customize, and maintain Oracle *i*FS. Topics include:

What is Oracle *i*FS?

Oracle *i*FS is a file system running on top of a database. In essence, Oracle *i*FS provides the paradigm translation between file systems and databases. These two separate worlds of data storage meet in Oracle *i*FS. For file system users, Oracle *i*FS behaves as if it were a standard file server, organizing files into hierarchies of folders. Users may not realize that the data is stored in a database, since Oracle *i*FS behaves just like any file, web, or e-mail server and the end user does not have to interact directly with the database.

Oracle *i*FS actually stores files in a relational database, rather than on a local hard drive. Administrators can perform many tasks using Oracle *i*FS that are not possible using standard file systems.

The following definitions of Oracle *i*FS components may help you to better understand Oracle *i*FS.

Benefits of Using Oracle *i*FS

The benefits of using Oracle *i*FS include simplified administration, universal access to information for end users, application development, and National Language Support. Each section below describes these benefits.

Simplified Administration

For administrators, Oracle *i*FS has several advantages, resulting from storing data and running services within the Oracle8*i* database and supporting standards:

- Simplified system maintenance—Access and performance for both the Oracle database and Oracle *i*FS improves when you scale the system to new machines, add processors for performance, or perform other maintenance tasks. File storage and access, e-mail, and application performance automatically improve.
- Simplified system security—Administrators can manage security for files, e-mail messages, web pages, and relational data within the same system, using the same tools.
- Simplified backup—Backing up the Oracle8*i* database automatically backs up files, messaging, and other contents of Oracle *i*FS.

End Users

End users get the following benefits from Oracle *i*FS:

- Universal Access to Information—Users can access the contents of Oracle *i*FS through Windows Explorer, any Web browser, e-mail clients, and FTP clients. Users can view a file directly in Oracle *i*FS by double-clicking it in Windows Explorer.
- Different Clients, Same Functionality—The Web interface provides the functionality for checking out files, creating new versions, and searching for files.
- Advanced Searches—The search capabilities of Oracle *i*FS far surpass anything possible in file systems. Not only is it easier to find a single file by searching for its external attributes and its content, but users can perform searches across different file types. For example, a user could search for all the Word files that contain references to Project X, as well as presentations, images, e-mails, web pages, and other files that relate to that project. Oracle *i*FS indexes all content automatically; searching is faster because Oracle *i*FS searches the index, not the files' content.
- File Management and Security—If users need to share files while still controlling access to them, users can enforce check in/check out procedures, as well as control the privileges of other users to view, edit, or delete files.

  Users can also lock or unlock files. Locking a file prevents anyone from modifying the same file.
- Backing Up Data—Files stored in Oracle *i*FS are backed up as often as the database itself, so that they can be restored from backups when needed.

Application Development

Extending Oracle *i*FS Classes To customize the system to meet specific needs, developers can extend the Oracle *i*FS classes to add custom attributes.

XML Support This structured format is entirely composed of attributes. The SimpleXMLParser, which comes with Oracle *i*FS, enables developers to create documents with custom attributes without writing Java code.

National Language Support

Oracle *i*FS supports storing documents of different character sets. For example, an international company can store German and Japanese documents in a single file system. More importantly, you will be able to search on these multi-lingual documents.

Files: A Definition

A file in Oracle *i*FS is any type of electronic data that end users store in Oracle *i*FS, including all types of word processing files, presentations, graphic images, HTML web pages, spreadsheets, e-mail messages, and so on. Files may be complex in nature, consisting of both structured data components, such as attributes, and unstructured data components, such as document content. Files loaded into Oracle *i*FS are parsed without customization. Files are also parsed to extract attributes.

Oracle *i*FS Service

An Oracle *i*FS process runs in a single Java Virtual Machine (JVM). An example of an Oracle *i*FS process is an File Transfer Protocol (FTP) server or the Server Message Block (SMB) server. Each Oracle *i*FS process has an Oracle *i*FS service which manages the interaction with the database.

An Oracle *i*FS service is configured by a service properties file which contains information such as connection pool sizes and cache sizes. Multiple Oracle *i*FS services can share the same properties file.

If you use Oracle *i*FS out of the box:

- Each protocol server has its own Oracle *i*FS service.
- Each of these Oracle *i*FS services has its own name and therefore, service properties file.
- Each of these service properties files points to the same database instance and database schema.

The Oracle HTTP Server

The Oracle HTTP Server is a component of Oracle8*i*, Release 3 (8.1.7) and the Oracle Internet Application Server (iAS). This HTTP server provides an Oracle implementation of the Apache Web Server, version 1.3.12 that works in conjunction with Jserv 1.0. In addition, Oracle HTTP Server also includes the Oracle Java Server Pages (JSP) engine.

Although Oracle *i*FS ships the Java Web Server (JWS), version 2. 0, and is the default Web server, Oracle *i*FS has also been certified to work with the Oracle Apache Web Server. The Oracle Apache Web Server is an additional service provided to facilitate customers who want to use Apache as their Web server.

Apache JServ is a 100% pure Java servlet engine fully compliant with the 2.0 specification of the JavaSoft Java Servlet API.

See Also

- For more information on service properties files, see Appendix B, "Secondary Properties".
- For more information on how to configure the Apache Web Server and the Jserv Servlet engine, see the *Oracle HTTP Server Release Manual, version 1.3.12*.
- For more information on application development, see Chapter 10, "Using Oracle iFS Manager for Development Tasks".
- For more information on the language and character sets that Oracle *i*FS supports, see the *Oracle Internet File System Installation Guide*.

Oracle *i*FS Components

The Oracle *i*FS system is divided into three layers. Figure 1-1 demonstrates these concepts.

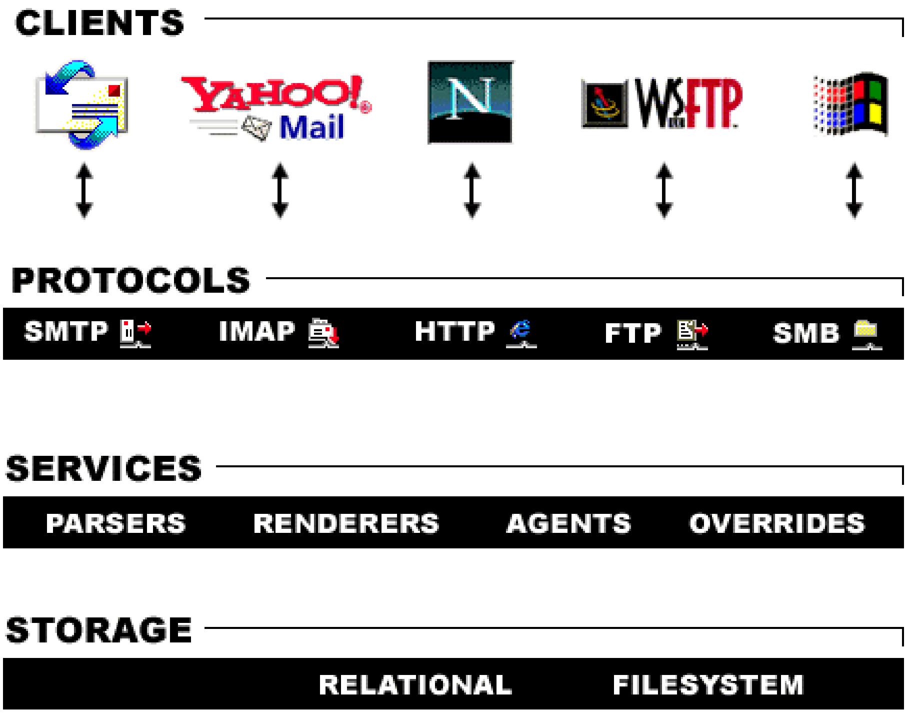
Protocols—The protocols are the familiar interfaces with which the end user interacts. Because these are standard protocols, users are not even aware that their files are being stored in a database.

Services—This layer provides actions on the repository, such as parsing and rendering. Developers can extend the repository’s behavior by adding their custom business logic. For example, a developer may write his or her own custom agent or override.

Storage—The repository is the Oracle *i*FS foundation, storing both relational data and file information.

Note: Remember that the Oracle *i*FS repository is owned by `ifssys` user.

Figure 1–1 Oracle iFS Components



Oracle *i*FS Uses Three Types of Objects

Oracle *i*FS is based on three main objects. Users will be aware of these objects; however, as the administrator, you will see the underlying ways in which Oracle *i*FS actually classifies and stores files, folders, and other file system components. Oracle *i*FS uses a combination of relational and object-oriented concepts in defining the repository.

- Public Objects—These objects are the most common objects in the Oracle *i*FS repository and may be manipulated by end users. Public objects can be placed in folders. Access to them is controlled by Access Control Lists (ACLs).
- Schema Objects—These objects define the class hierarchy, the object attributes, and validation. The class hierarchy shows which child classes inherit from

which parent classes. Only an administrator can create, modify, or delete schema objects. Examples of schema objects are class objects and class domains.

- System Objects—These objects pertain to the overall Oracle *i*FS system. Only an administrator can create, modify, or delete system objects. Examples of system objects are formats and policies.

See Also

For more information on the folder hierarchy, see Chapter 10, "Using Oracle iFS Manager for Development Tasks".

Setting Up Oracle *i*FS is a Four-Step Process

To set up Oracle *i*FS, you must complete four steps:

1. Plan your system and set up the folder hierarchy: Before using Oracle *i*FS, define the structure of the folder hierarchy based on organization structure, information or workflow, or user recommendations.
2. Add users and groups: After planning and setting up your folder hierarchy, add the users and define groups of users to make administration and the applications of ACLs to files easier.
3. Load files into Oracle *i*FS: Migrate your data into Oracle *i*FS from your legacy file systems.
4. Set up security: Security covers access to the Oracle *i*FS system and its various functions, and access to particular folders and files. Identify those users who need administration privileges to manage their department and folder hierarchy. Define the basic ACLs to apply to files. As you add users, you assign passwords.

Note: It is recommended that you change the system user's password because it is published in this guide. Anyone can access the administrator account if the password is not changed. For information on changing the password, see Chapter 5, "Task 2: Add Users and Groups".

Task 1: Plan Your System and Set Up the Folder Hierarchy

The folder hierarchy is a very important component of Oracle *i*FS. It is a representation of how the files in the repository are organized. It is familiar to all users and makes organizing and browsing information easy. An important prerequisite is to model how you want to organize the file hierarchy. For example:

```
MyHome
    Correspondence
    Conferences
        Oracle Open World
        Oracle Applications Users Group
    Plans
        Presentations
Partners Program
    Partner Information
    Program Plan
    Technical Whitepapers
Public
    Product Information
        Oracle iFS
                Manuals
                Software Downloads
        Oracle JDeveloper
        Oracle8i Server
    Standards and Procedures
        Company Policies
        Procedure Manuals
    home/
    public/
    root/
```

When constructing your folder hierarchy, answer these questions:

- Who will be using Oracle *i*FS?
- What protocol servers do you need to run?
- What kinds of users will be using Oracle *i*FS? Do you want to accommodate guest users, or give administrative permissions to some users or developers?
- Do users naturally group together? Should you set up groups within groups?
- What kinds of access should be allowed to files?
- What types of files do you have? Do they include .doc, .txt, .html, or .xml?

Task 2: Add Users and Groups

After setting up the folder hierarchy, create users and groups. Oracle *i*FS users are the end users who use Oracle *i*FS as a file management system. Groups are composed of users that are related in some way. For example, the Marketing group could contain all of the members of the corporate marketing organization.

There are no groups provided with a new Oracle *i*FS installation.

Task 3: Load Files into Oracle *i*FS

It may be necessary to migrate your data from your legacy systems. This process requires you to manually transfer the data to the new environment.

There are four different ways to load files into Oracle *i*FS:

- Drag and drop—In Windows Explorer (if the SMB or NTFS server is running) and the Web interface (if the Web server is running), you can drag and drop files into Oracle *i*FS as well as build the folder hierarchy.
- FTP—If the File Transfer Protocol (FTP) server is running, use any FTP client to load files into Oracle *i*FS. FTP is the most efficient protocol to use for bulk file loading.
- Upload by Browse—Use the Web interface to select a single file to upload to Oracle *i*FS. This method allows you to upload only one file at a time.
- Command Line Utilities—Use the Command Line Utilities to load one file at a time into Oracle *i*FS.

Task 4: Set Up Security

Now that you have established the folder hierarchy, created users and groups, and loaded files into Oracle *i*FS, you can set up security for these items in Oracle *i*FS. Security in Oracle *i*FS is defined using Access Control Lists (ACLs). An ACL is a list of users and groups with a set of access permissions. The ACL is applied to a file or folder to grant the defined access settings for the users and groups in the list.

The following ACLs are supplied out-of-the-box:

- PUBLIC—All users have all permissions.
- PUBLISHED—All users have read permissions, and can view, but not edit, the attributes or content. The owner has all permissions.
- PRIVATE—The owner has all permissions; no other users have access. This is the default ACL.

- PROTECTED—To be used for folders; users can add and remove files, but they cannot rename or delete the folder in which the file resides. The owner has all permissions.

See Also

- For more information on planning your file system and setting up your folder hierarchy, see Chapter 4, "Task 1: Set Up the Folder Hierarchy".
- For more information on creating users and groups in Oracle *i*FS, see Chapter 5, "Task 2: Add Users and Groups".
- For more information on loading files into Oracle *i*FS, see Chapter 6, "Task 3: Load Files into Oracle iFS".
- For more information on setting up security in Oracle *i*FS, see Chapter 7, "Task 4: Set Up Security".

The Five Administration Tools

Oracle *i*FS provides administrative capabilities in four tools:

- Oracle *i*FS Manager
- Web interface
- Command Line Utilities
- XML configuration files
- Server Manager

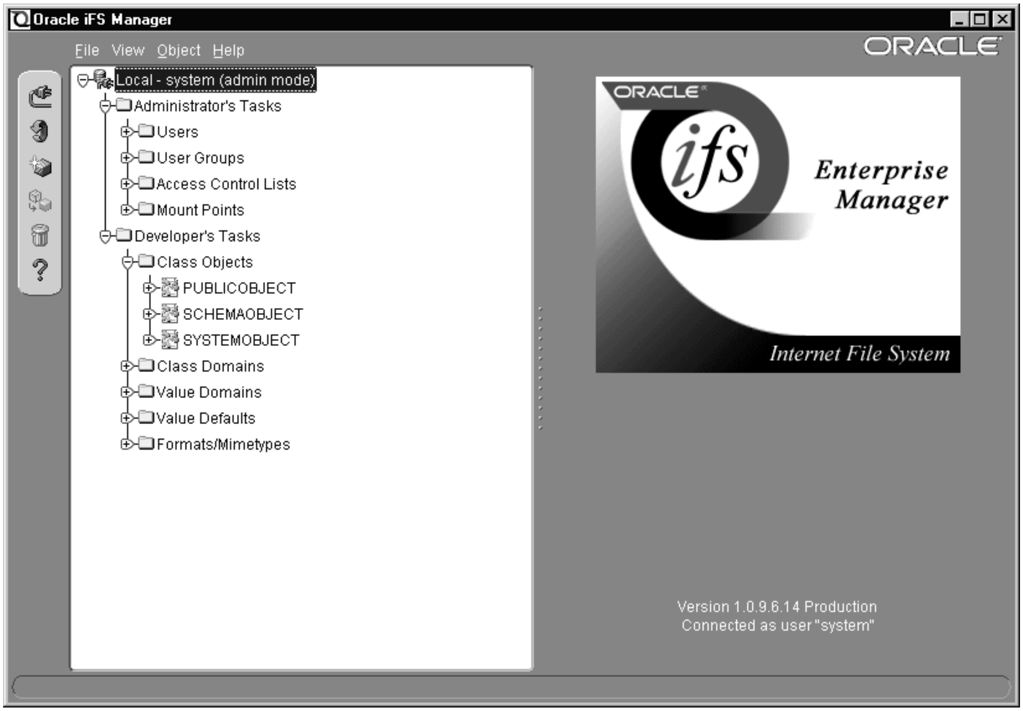
Oracle *i*FS Manager is the tool to use for most administration tasks. The Web interface is the easiest tool to use, though limited in its administrative functionality. The Command Line Utilities are used for administrators familiar with the command line. XML configuration files can be used to streamline the creation of large numbers of repository objects. You use Server Manager to manage and monitor the servers and agents for Oracle *i*FS.

Use Oracle *i*FS Manager As An Extension of OEM

Oracle *i*FS Manager uses the OEM toolkit and framework to provide a single-point administration tool. For an illustration of the Oracle *i*FS Manager interface as it looks after logging in, see Figure 1-2, Oracle *i*FS Manager.

Using Oracle *i*FS Manager, you can perform both administrative tasks and development tasks. Oracle *i*FS Manager also contains a built-in file system browser to navigate the folder hierarchy of Oracle *i*FS and includes access to the Server Manager interface.

Figure 1–2 Oracle iFS Manager



Version 1.0.9.6.14 Production
Connected as user "system"

Administrative tasks include creating and modifying:

- Users—Users are the end users who use Oracle *i*FS as a file management system.
- Groups—By assigning users to groups, administration and maintenance is easier. Instead of adding each user to an Access Control List interface for a file or folder individually, you can add a group of users all at the same time and manage multiple users a single group.
- ACLs—Security for all objects, including folders and files, is maintained through ACLs. An ACL specifies the permissions granted to or revoked for a particular user or group.
- Mount Points—Mount points in Oracle *i*FS are named directories which can be accessed through Windows (SMB) and the Web interface.

Development tasks include creating and modifying:

- Class Objects—A class object represents a class with instances that are persistently stored and managed by the Oracle *i*FS server.
- Class domains—A class domain limits specific classes for attributes that point to objects.
- Value domains—A value domain specifies valid attribute values. A value domain can be a list or a range.
- Value defaults—A value default specifies an initial value for an attribute if none is given.
- Formats/Mimetypes—The format of a document specifies the way the document information is stored, indexed or not.
- Register parsers—A custom parser and a custom XML parser can be registered to a file extension.
- Register renderers—A custom renderer can be registered to a specific class.
- Register Java Server Page (JSP) Lookups—A custom Java Server Page (JSP) can be registered to a specific class.

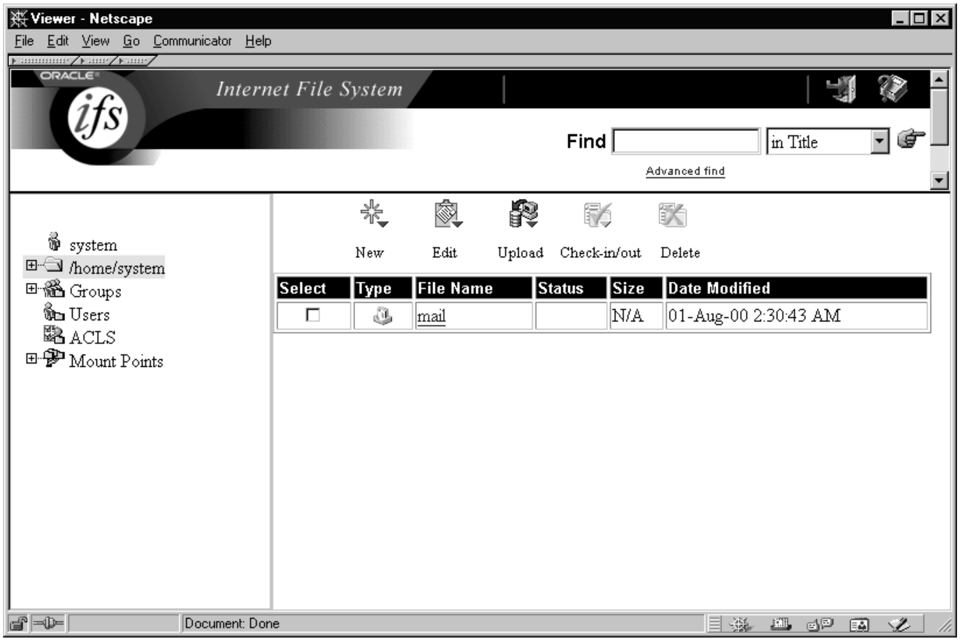
Use the Web Interface for Basic Tasks

The advantage to using the Web interface to administer Oracle *i*FS is that it can be used on any platform and is accessible from any machine that has a browser. For an illustration of the Web interface as it looks after logging in, see Figure 1-3, Web Interface.

The Web interface is the best tool available when completing the following simple tasks:

- Creating a single user or group.
- Displaying all users or groups.
- Editing a user's profile.
- Changing a user's password.
- Modifying ACLs by adding or removing Access Control Entries (ACEs) for users and groups.
- Applying an Access Control List (ACL) to a document or folder.

Figure 1–3 Oracle iFS Web Interface



Select	Type	File Name	Status	Size	Date Modified
□		mail		N/A	01-Aug-00 2:30:43 AM

Use XML Configuration Files to Create Repository Objects

The XML configuration files provide a convenient shortcut for creating users, ACLs, and other repository components. Using XML, you can create the same Oracle *i*FS setup on another server.

For example, this XML file creates a new user:

```
<SimpleUser>
   <UserName>gking</UserName>
   <Password>ifs</Password>
   <DistinguishedNameSuffix>.yourcompany.com</DistinguishedNameSuffix>
   <AdminEnabled>true</AdminEnabled>
   <HomeFolderRoot>/home</HomeFolderRoot>
   <EmailAddressSuffix>@yourcompany.com</EmailAddressSuffix>
</SimpleUser>
```

When you create objects using XML, you can use the Windows or Web interfaces, or FTP to load the file into Oracle *i*FS. When you load the file, it must be parsed in order to create the objects specified in the XML file. Parsing occurs automatically, as necessary.

Use the Command Line Utilities for Regular Maintenance

The Command Line Utilities allow you to access and manipulate the Oracle *i*FS objects at a very detailed level. The Command Line Utilities provide functionality and commands that are similar to UNIX commands with additional capabilities specific to Oracle *i*FS. Usually, you would use the Command Line Utility if you have Oracle *i*FS Manager installed on the client-side, but do not have it installed on the server-side.

Using the Command Line Utilities, you can browse Oracle *i*FS, list and modify an object's attributes, create new folders and files, and perform simple searches.

In addition, you can write any operating system script, such as Perl, and incorporate the Command Line Utilities commands. You can also write a batch script, comprised entirely of Command Line Utilities commands, to be executed regularly.

Use Server Manager to Manage and Monitor the Running Servers and Agents

Using Server Manager, you can start, stop, suspend, resume, and list information for the various servers and agents. You can also use Server Manager to manage the servers and agents remotely from other machines from which the agents and servers may be running. In this way, it is possible to have protocol servers running on different machines, yet managed from a single Server Manager. Server Manager itself is a server, so one Server Manager can manage any other Server Managers in the system.

See Also

- For more information on using the Web interface, see the *Oracle Internet File System User's Guide*.
- For more information on using XML configuration files to create users and groups, see Chapter 5, "Task 2: Add Users and Groups".
- For a complete listing of Command Line utilities, see Appendix A, "Command Line Utilities Reference".
- For more information on using Server Manager, see Chapter 2, "Starting Oracle iFS" and Chapter 8, "Using Server Manager to Start and Stop Servers".

2

Starting Oracle *i*FS

To use Oracle *i*FS, you must start the servers and agents used by Oracle *i*FS. This chapter shows you how to start Oracle *i*FS. Topics include:

Starting Oracle *i*FS

Before you can use Oracle *i*FS, you must first start the servers and agents used by Oracle *i*FS. These processes used by Oracle *i*FS are managed using Server Manager. This tool provides managing and monitoring capabilities for the Oracle *i*FS servers and agents.

To start the servers and agents, you must first start Server Manager. To do this, use a command line on the same machine where Oracle *i*FS is installed.

Server Manager can be run in two different ways: interactively or running in the background. Running Server Manager in the background is recommended if it will have any agents running. This avoids the possibility of mistakenly exiting out of Server Manager and shutting down all the running agents and protocol servers it was managing. An interactive Server Manager can be used to monitor any agents, protocol servers and other server managers in the system.

When Server Manager is started, the agents, if specified in the definition file, are also started. Starting all the agents and protocol servers may take a few minutes depending upon the number of agents and available hardware resources.

See Also

For an introduction to Server Manager and to learn about the various commands to use with Server Manager, see Chapter 8, "Using Server Manager to Start and Stop Servers".

Starting Server Manager

To start Server Manager, use the `ifsstart` command. This starts Server Manager running in the background. The `ifsstart` command is found in the following directory:

Platform	Directory
UNIX	`$ORACLE_HOME/ifs<version>/bin`
Windows NT	`%ORACLE_HOME%\ifs<version>\bin`

You can use the `ifsstart -h` command to access help.

Using ifstart on UNIX

On UNIX, you can use the `ifsstart` command with parameters. Use the following syntax:

```
ifsstart [-w] [-a] [p]
```

When Server Manager is started, a dialog displays where you must enter the following:

- Oracle *i*FS username—If nothing is entered, the default is `system`.
- Oracle *i*FS password—The username's password.
- Oracle *i*FS service name—The default is ServerManager.
- Oracle *i*FS schema password—The schema password.

The following parameters may be passed to the `ifsstart` command:

Parameter	**Description**
no parameters	Two Server Managers are started in the background and load all agents in the definition files and start all agents that have a TRUE setting for the "Start" parameter. By default, all agents including any External Server Agents for the protocols selected in the Oracle *i*FS configuration process are set to true. If the Java Web Server (JWS) was selected in the Oracle *i*FS configuration process, then it is started. The Sendmail process is also started.
`-w`	For UNIX only. Server Manager loads all agents in the definition file and starts all agents, including External Server Agents that have a TRUE setting for the "Start" parameter. The Sendmail process is started. JWS is not started.
`-a`	For UNIX only. Server Manager starts all agents that have a TRUE setting for the "Start" parameter, excluding all External Server Agents even if they have a TRUE setting. This parameter also starts the Sendmail process and the JWS if selected in the Oracle *i*FS configuration process.
-p	For UNIX only. Server Manager starts all External Server agents with a TRUE setting for the "Start" parameter. This flag also starts the Sendmail process and JWS if selected in the Oracle *i*FS configuration process.

If you choose to run Server Manager in the background, after using the `ifsstart` command, there is no information displayed.

Using ifsstart on Windows NT

On Windows NT, you can simply use the `ifsstart` command from a command prompt to start Server Manager. You must be in the `%ORACLE_HOME%` directory.

Using the ifssvrmgr Command

The `ifssvrmgr` command is used to start the default Server Manager to monitor the system or to start with a custom definition file. This starts Server Manager interactively on either UNIX or Windows NT. You can use the `ifssvrmgr` command with or without a definition file specified. The `ifssvrmgr` file is found in the following directory:

Platform	Directory
UNIX	`$ORACLE_HOME/ifs<version>/bin`
Windows NT	`%ORACLE_HOME%\ifs<version>\bin`

Use the following syntax:

```
ifssvrmgr [<definition_filename>]
```

The following parameters are passed to the `ifsvrmgr` command:

Parameter	Description
no parameters	Server Manager runs interactively with no agents loaded. This Server Manager can be used to monitor and manage the agents and servers in the system.
`definition_filename`	Server Manager runs interactively, using the specified custom definition file to load and start the agents as indicated by the "Start" parameter. This Server Manager can be used to monitor and manage the agents and servers in the system.

Note: The specification must be an aboluste or relative path and the definition file.

Stopping Oracle *i*FS

Use Server Manager to shut down any running agents and protocol servers it was managing. The `ifsstop` command will automatcially shut these processes down, and will stop the JWS and the Sendmail process, if they are running.

There are three different ways to shut down Oracle *i*FS servers:

- Shut down all agents and servers in the Oracle *i*FS system.
- Shut down all agents and servers managed by a specific Server Manager.
- Shut down specific an agent or server managed by a specific Server Manager.

Shutting Down All Agents and Servers in Oracle *i*FS

To shut down all agents and servers in Oracle *i*FS, use the `ifsstop` command. The `ifsstop` command will shut down all agents, protocol servers and Server Manager started with the `ifsstart` command. `ifsstop` will also stop JWS and the Sendmail process, if running. The `ifsstop` command is found in the following directory:

Platform	Directory
UNIX	`$ORACLE_HOME/ifs<version>/bin`
Windows NT	`%ORACLE_HOME%\ifs<version>\bin`

Syntax:

```
ifsstop
```

Note: You must use the `ifsstop` command from the `%ORACLE_HOME%` directory. This command returns an error if used from a Server Manager prompt.

After running the `ifsstop` command, you will be prompted to enter the following information:

- Java Web Server (JWS) administrator password—The password for the JWS administrator user (default is admin).

- Oracle *i*FS username—If nothing is entered, the default is `system`.
- Oracle *i*FS password—The username's password.
- Oracle *i*FS service name—The default is ServerManager.
- Oracle *i*FS schema password—The schema password.

Shutting Down All Agents and Servers Managed by a Specific Server Manager

To shut down all agents and servers managed by a specific Server Manager, you must run the `stop server <server_manager>` command from a Server Manager interactive session. All running agents and protocols servers managed by the Server Manager specified will be shut down, including the specified Server Manager. JWS and Sendmail cannot be stopped with this command.

Syntax:

```
stop server <server_manager>
```

Parameters:

`<server_manager>`	The name or ID of a Server Manager

Shutting Down Specific Agents or Servers Managed by a Specific Server Manager

To shut down specific agents or servers managed by a specific Server Manager, you must run the stop `server <server_name> -m <server_manager>` or the `stop agent <agent_name> -m <server_manager>` command from a Server Manager interactive session. The specified agent or protocol server managed by the specified Server Manager will be shut down. JWS and Sendmail cannot be stopped with this command.

Syntax:

```
stop server <server_name> -m <server_manager>
```

```
stop agent <agent_name> -m <server_manager>
```

Parameters:

`server_name` The name of the protocol server being managed by the specified Server Manager to be shut down. The associated External Service agent will also be shut down automatically.

`agent_name` The name of the agent being managed by the specified Server manager to be shut down. If the agent is an external service agent, the associated protocol server will also be shut down automatically.

`server_manager` The name or ID of a Server Manager.

Starting the JWS

Oracle *i*FS provides the standard implementation of the Sun Java Web Server 2.0 as part of its distribution. Before using the Web interface to administer Oracle *i*FS, you must start the Web server. To do this, use the `ifsstart` command, which starts all protocol servers. To administer the Java Web Server, log into port 1717 using HTTP.

To start the Web server independently, use the `ifsjwsstart` file located in the following directory:

Platform	**Location**
UNIX	`$ORACLE_HOME/ifs<version>/bin`
Windows NT	`%ORACLE_HOME%\ifs<version>\bin`

See Also

- For more information on using the `ifsstart` command, see Chapter 8, "Using Server Manager to Start and Stop Servers".

Configuring the System for Optimal Performance

Before running Oracle *i*FS, it is important to understand how to configure the system for optimal performance. There are three types of files to be recognized. The following table lists the file, gives a brief description, and the location:

File	Description	Location
Properties files	Used to configure Oracle *i*FS services.	`$ORACLE_HOME/ifs<version>/settings/oracle/ifs/server/properties/<service-name>.properties`
Definition files	Used to configure Server Manager and protocol servers.	`$ORACLE_HOME/ifs<version>/settings/<server-name>.def`
Log files	Used by Server Manager and protocol servers for logging activity and error conditions.	`$ORACLE_HOME/ifs<version>/log/<server-name>.log`

Property Descriptions

Use a text editor to edit the following properties in the Properties file. These properties are configured during installation.

Property	Specifies	Example
Database Properties		
`User`	The name of the Oracle *i*FS schema (the Oracle user that owns the Oracle *i*FS schema). Required; no default value.	`User=`*`ifssys`*
`Driver`	The fully-qualified class of the Java JDBC driver to use when connecting to the Oracle8*i* database. Required.	`driver=`*`oracle.jdbc.driver.OracleDriver`*
`DatabaseUrl`	Which database to connect to, depending on the JDBC driver. Required; no default value.	`DatabaseUrl=`*`jdbc:oracle:oci8:@tnsname`*

Note: The driver property has a fixed value and should not be modified.

Secondary Properties

There are several secondary properties you may modify, including Server, User Session, and Tracing properties. Oracle Corporation recommends that you retain the default settings for these secondary properties. Refer to Appendix B, "Secondary Properties" for a list of these secondary properties.

Using Analyze

To maintain peak performance, it is important for the Oracle8*i* database to know the storage distribution of the files in Oracle *i*FS. It is recommended that you run analyze to update the database statistics, especially after adding large numbers of documents, such as after a bulk load.

You will need to run analyze more often when you first install Oracle *i*FS. Adding two-hundred documents to an existing five-hundred documents has a larger impact than adding two-hundred documents to ten-thousand documents.

See Also

- For more information on configuration, see the *Oracle Internet File System Installation Guide.*

Logging In as the Administrator for the First Time

Logging in to any Oracle *i*FS tool for the first time is a simple process. Oracle *i*FS Manager does not require you to start a server. Before using the Web interface, however, you must start the Java Web Server. To use the Command Line Utilities, you must start the Command Line Protocol server (CUP).

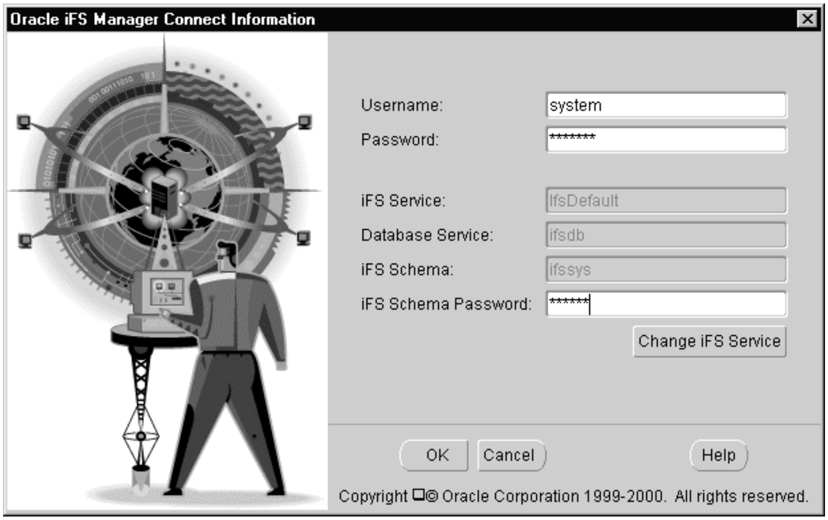
Starting and Logging in to Oracle *i*FS Manager

To start Oracle *i*FS Manager, run the `ifsmgr` script installed with Oracle *i*FS from a command line prompt. This script is located in the following directory:

Platform	**Directory**
UNIX	`$ORACLE_HOME/ifs<version>/bin`
Windows NT	`%ORACLE_HOME%\ifs<version>\bin`

This directory contains either `ifsmgr.bat` for Windows NT machines or `ifsmgr.sh` for UNIX machines.

When Oracle *i*FS Manager starts, the Login window displays.



Username: system

Password: ********

iFS Service: IfsDefault

Database Service: ifsdb

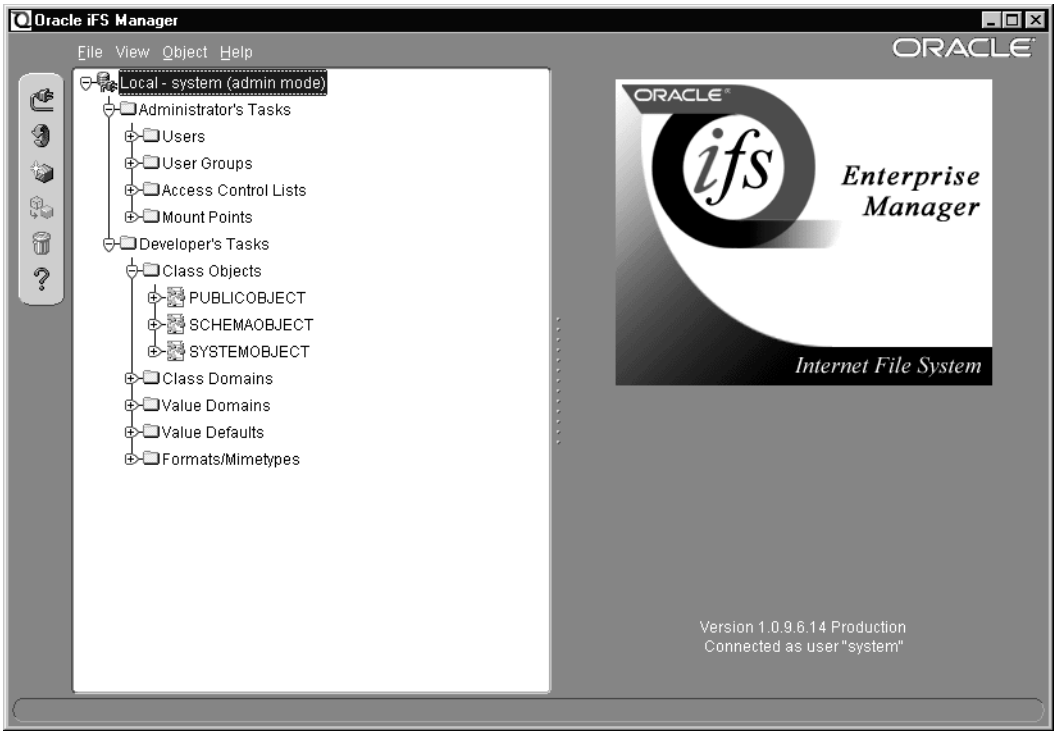
iFS Schema: ifssys

iFS Schema Password: ******

Copyright □© Oracle Corporation 1999-2000. All rights reserved.

1. Enter a username and password (the default administrator login is system / manager).
2. Notice that the following fields are grayed out. They display information for the underlying Oracle *i*FS Service:
   - Oracle *i*FS Service—The property settings for the Oracle *i*FS Service. This field is not editable, but you can change it by clicking Change Oracle *i*FS Service.
   - Database Service—The Oracle database Net8 Service Name. This field is not editable and is determined by the Oracle *i*FS Service field.
   - Oracle *i*FS Schema—The Oracle database schema name for the Oracle *i*FS repository. This field is not editable and is determined by the Oracle *i*FS Service field.
3. Enter the Oracle *i*FS Schema Password.
4. Click OK to complete the logon process.

The main window for Oracle *i*FS Manager displays:



Version 1.0.9.6.14 Production
Connected as user "system"

The following table lists the default users in a new Oracle *i*FS installation:

Username	Password	Description
system	manager	User with administrator permissions. It is recommended that the password be changed when you first login. This system user is an Oracle *i*FS user and is not necessarily the database user, system / manager.
scott	tiger	A sample user without administrator permissions.
guest	welcome	A guest user to allow anonymous access to Oracle *i*FS. The guest user has access to the /ifs/public Folder and any other folders and documents with the Public, Published, or Protected ACL.

Note: The guest user is required by Oracle *i*FS File Transfer Protocol (FTP) and the Command Line Utility Protocol (CUP) to function correctly. If the guest user is deleted, you will not be able to log into FTP or CUP.

Tips

To change the Oracle *i*FS Service, click the Change Oracle *i*FS Service button. Enter a new Oracle *i*FS Service in the field provided and click OK. This step is optional.

See Also

For more information on logging into the Web interface and using it to administer Oracle *i*FS, see the *Internet File System User's Guide*.

3

Using Oracle *i*FS Manager

Oracle *i*FS Manager is the primary tool used to administer Oracle *i*FS. This chapter introduces Oracle *i*FS Manager and describes how to use it. Topics include:

Oracle *i*FS Manager: An Overview

Oracle *i*FS Manager is an administration application that manages Oracle *i*FS-specific system objects and processes. It is an Oracle Enterprise Manager (OEM) extension, but currently can only be launched as a stand-alone application.

Oracle *i*FS Manager provides a powerful and easy-to-use graphical interface, and is divided between administrative tasks and developmental tasks. The following administrative tasks can be performed using Oracle *i*FS Manager:

- Create, review, modify, and delete file system components and Oracle *i*FS objects such as users, groups, Access Control Lists (ACLs), and mount points.
- Create, review, modify, and delete folders.
- Monitor and manage protocol servers and agents.

The following developmental tasks can be performed using Oracle *i*FS Manager:

- Create, review, modify, and delete Oracle *i*FS objects such as class objects, value domains, value defaults, and formats/mimetypes.
- Register parsers, renderers, and JSP lookups.

You Must Have Administrative Permissions

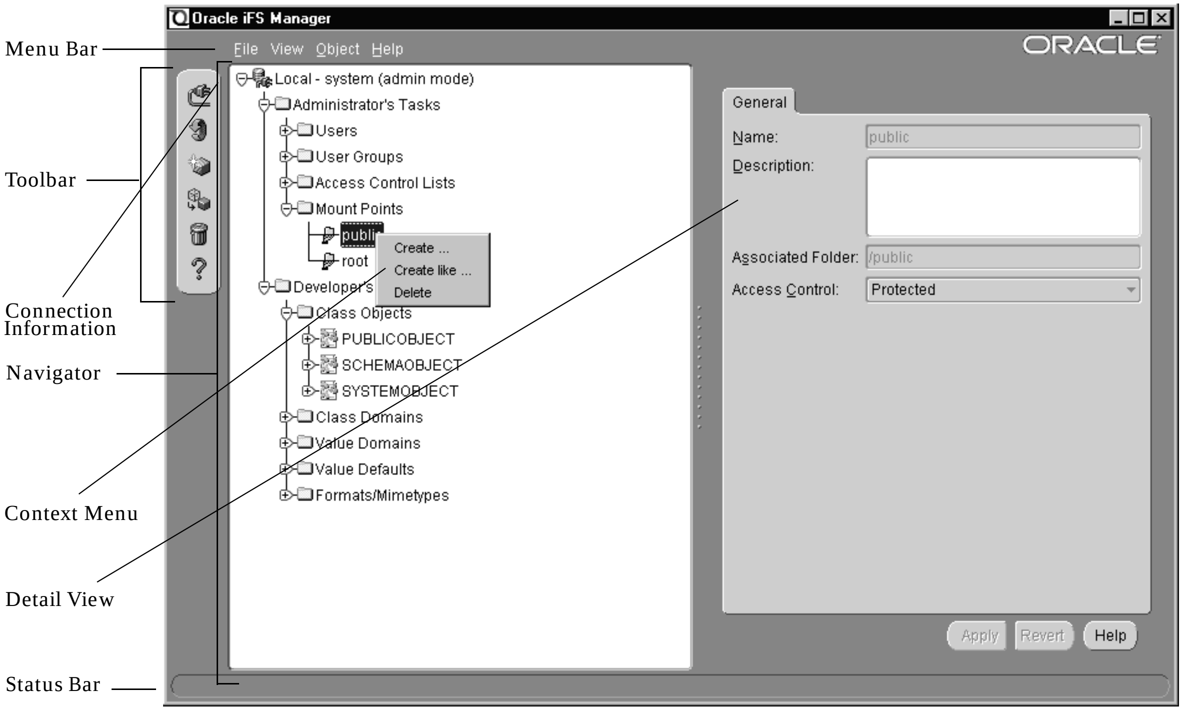
To log into Oracle *i*FS Manager, you must have administrative permissions. All tasks are performed in administrator mode; they are not restricted by ACL-based security checking.

Parts of the Oracle *i*FS Manager Window

The main Oracle *i*FS Manager window has the following parts:

- Menu Bar
- Connection Information
- Toolbar
- Navigator
- Context Menus
- Detail View
- Status Bar
- Browser

Figure 3–1 Oracle iFS Manager



Name: public

Description:

Associated Folder: /public

Access Control: Protected

Menu Bar

The menu bar provides access to all commands and to special features not available using the toolbar or context menus. Click each menu to display its commands. Click a command to execute it. There are four menus:

- File Menu
- View Menu
- Object Menu
- Help Menu

File Menu

The File menu controls the connection to an Oracle *i*FS instance and provides access to the Oracle *i*FS folder hierarchy. Commands include:

- Connect—Display the current connection to an Oracle *i*FS instance. Administrative tasks can be performed only on an Oracle *i*FS instance connected to Oracle *i*FS Manager.
- Disconnect—Break the connection of an Oracle *i*FS instance.
- Monitor—Launch a window to manager Server Manager.
- Browse—Launch a window to examine the Oracle *i*FS file hierarchy.
- Exit—Leave Oracle *i*FS Manager.

View Menu

From the View menu, you can refresh the information displayed in the Navigator and toggle on and off the display of the toolbar and status bar. Commands include:

- Refresh—Update the items in the Navigator under the selected object to include any changes to the repository during your session of Oracle *i*FS Manager. The Detail View window is also refreshed based on the object selected.
- Show Toolbar—Display the toolbar. Show is the default setting.
- Show Statusbar—Display the status bar. Show is the default setting.

Object Menu

The Object menu provides access to commands you can perform on Oracle *i*FS objects:

- Create—Define a new object. Select the type of object to be created, then complete the property sheet to describe it.
- Create Like—Define a new object based on an existing one. Select the object to be copied, and Oracle *i*FS Manager populates the values in the property sheet for the new object with the values of the object you are copying. Complete the property sheet.
- Delete—Remove the selected object. Oracle *i*FS Manager deletes the object and refreshes the Navigator.
- Register—Register a JSP lookup, parser, or renderer. Select the object type, then fill in the corresponding property sheet.

- Options—Select User Definitions to edit the default values for user creation.

Help Menu

The Help menu provides links to online help.

- Contents—Launch the Help Navigator window. Use the Contents tab to select from the help topics available. Use the Index tab to perform a keyword search. Use the Search tab to perform a full-text search.
- Search for Help On—Launch the Help Navigator window with the Search tab activated.
- About Oracle Oracle *i*FS Manager—Display version information for Oracle *i*FS Manager.

Connection Information

The highest level of the Oracle *i*FS Manager hierarchy shows the Oracle *i*FS service it is connected to. On the Detail View, you can see which account is logged in.

Toolbar

The toolbar is made up of icons that represent frequently used commands. To display a caption describing the icon, pause the cursor on the icon. The toolbar can be displayed or hidden using the View menu. The following functions are provided:

Function	Icon	Description
Connect	🔌	View or change the current connection to Oracle *i*FS.
Refresh	🔄	Refresh the data displayed in Oracle *i*FS Manager.
Create	✨	Define a new Oracle *i*FS object (user, group, ACL).

Function	Icon	Description
Create Like		Define a new Oracle *i*FS object similar to an existing object. This icon is enabled only when an object is selected in the Navigator.
Delete	🗑	Remove a selected object. This button is enabled only when an object is selected in the Navigator.
Help	?	Display the help file.

Navigator

The Navigator displays a hierarchical tree view of all objects used to administer Oracle *i*FS. The objects are grouped by whether they are administrative or developmental tasks.

Each object type in the Navigator is identified by an icon and name. If there is a '+' or '-' to the left of an object's icon and name, the object is a container that can be expanded to display the objects it contains. A container is represented by a folder icon is a logical grouping of one specific type of object, such as users or groups.

Note: The tree view in the Navigator is often populated as the result of queries executed over a network. Therefore, there may be a delay when expanding a container.

Context Menus

As in other Windows applications, you can right-click an object to pop up a context menu; that is, a shortcut menu relating to the object right-clicked.

Selected Item	Context Menu Options
Object, such as a particular user or group	Create, Create Like, Delete
Instance	Connect, Disconnect
Container object, such as user or group	Create

Detail View

To the right of the Navigator is the Detail View, composed of one or more tabbed property sheets displaying information about the object selected. Often, these property sheets may be edited.

Status Bar

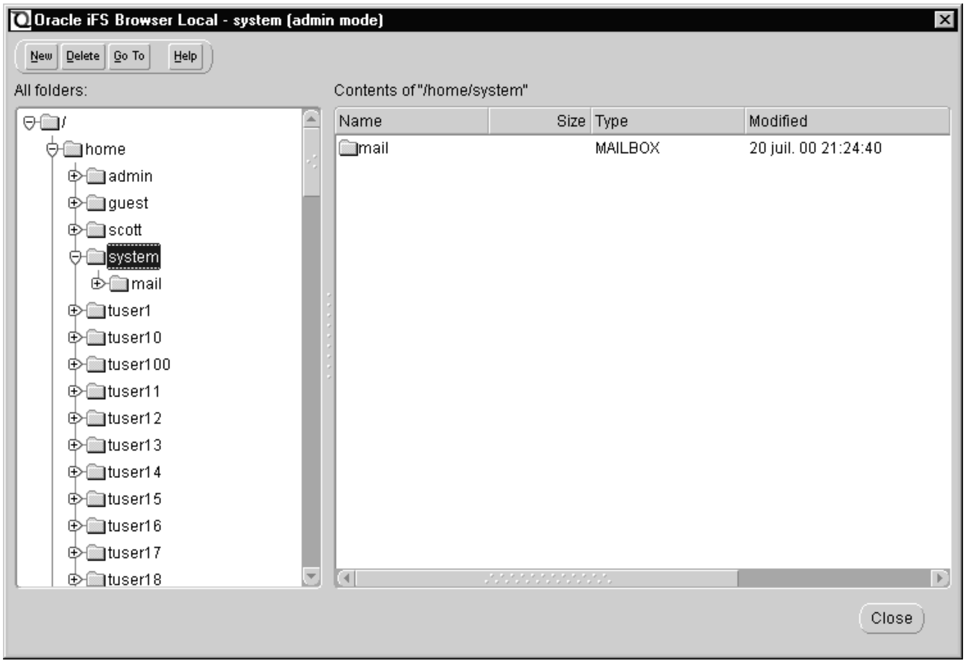
The status bar, below the Navigator, displays the current state of a process. For example, when a creation dialog is opened for creating a user, the status bar displays "Creating a user...". The status bar can be toggled on and off by selecting the Show Status Bar command on the View menu.

Browser

If you select Browse from the File menu, Oracle *i*FS Manager displays a window showing the folder hierarchy of the connected Oracle *i*FS instance. The browser consists of a Navigator and a toolbar that lets you create a new directory, delete a directory, and go to a specific directory. To the right of the Navigator, Oracle *i*FS Manager displays the attributes of folders as you select them in the Navigator. The Browser provides:

- A quick view of the contents of the folder hierarchy.
- An easy way to create new directories.
- A quick view of the attributes of the folders in the folder hierarchy.

Figure 3–2 Browser



Contents of "/home/system"

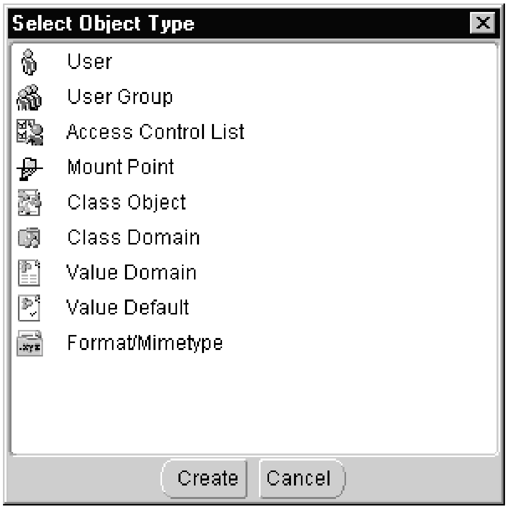
Name	Size	Type	Modified
mail		MAILBOX	20 juil. 00 21:24:40

Using Create and Create Like to Create New Objects

Oracle *i*FS Manager offers two ways to create new objects. To create a new object, you can:

- Use Create to create an object from scratch.

  1. Select Create from the toolbar, context menu, or the Object menu.
  2. Select the object type to be created from the Select Object Type dialog and click Create.



The empty Property Sheet displays.

3. Complete the property sheet and click Create.

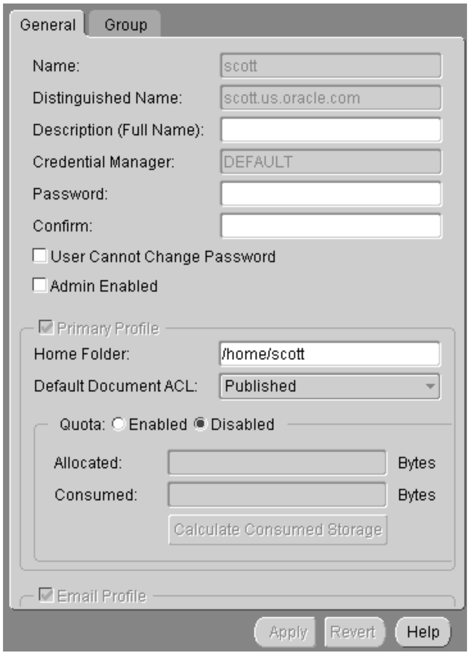
- Use Create Like to copy some attributes from the selected object. For example, if you want to create a new ACL with almost identical Access Control Entries (ACEs) as DocumentACL, select the DocumentACL, and click Create Like on the toolbar. The Create ACL dialog would be initialized with the ACEs from DocumentACL.

  1. Select an existing object.
  2. Select Create Like from the toolbar, context menu, or the Object option on the menu bar. The Property Sheet for the object selected displays.
  3. Complete the property sheet and click Create.

Working with Property Sheets

When an object in the Navigator is selected, one or more property sheets display in the Detail View. Use the property sheet to view details about objects or to modify them. See Figure 3–3 for an illustration of a property sheet.

Figure 3–3 Property Sheets



Name: scott

Distinguished Name: scott.us.oracle.com

Description (Full Name):

Credential Manager: DEFAULT

Password:

Confirm:

□ User Cannot Change Password

□ Admin Enabled

☑ Primary Profile

Home Folder: /home/scott

Default Document ACL: Published

Quota: ○ Enabled ● Disabled

Allocated: Bytes

Consumed: Bytes

☑ Email Profile

Modifying Objects Using Property Sheets

The following buttons are used when modifying objects using property sheets:

- Apply—Process the modifications made to the object.
- Revert—Cancel the modifications made to the object and reset the property sheet back to the original data.
- Help—Display context-sensitive help about the current property sheet.

Getting Help

When using Oracle *i*FS Manager, online help is available for windows and dialog boxes. The Help system is context-sensitive, meaning the topic displayed when you click Help pertains to the window or dialog box open. You can also find topics through a table of contents, use the index to perform a keyword search, or the Search tab to perform a full-text search.

There are several ways of accessing Help:

1. From the main window:
   - Press F1.
   - On the menu bar, select Contents from the Help menu.
   - On the toolbar, click the Help icon.
2. In a dialog box or on a property sheet:
   - Click the Help button or press F1.

4

Task 1: Set Up the Folder Hierarchy

This chapter covers planning and setting up your folder hierarchy. The following topics are included:

- What Is a Folder Hierarchy?
- Guidelines for Structuring the Folder Hierarchy
- Planning Your Folder Hierarchy and Access Requirements
- Setting Up the Folder Hierarchy: The Steps to Follow
- Creating and Deleting Folders Using Oracle iFS Manager
- Creating Folders Using the Command Line Utilities
- Creating Multiple Folders Using XML
- Creating and Modifying Mount Points
- Deleting Mount Points

What Is a Folder Hierarchy?

A folder hierarchy is an organizational structure of one or more folders in Oracle *i*FS. Folder hierarchies organize the repository so that users can browse through it easily. You can create multiple folder hierarchies to organize information in different ways to make browsing convenient for different types of users. For example, the sales, development, marketing, and consulting departments of a company may use different hierarchies.

What Is a Folder in Oracle *i*FS?

A folder in Oracle *i*FS is similar to a physical file folder or a computer file folder, and is used to group and organize files or other objects. The organization of the folders is conceptual, not physical. The files and objects are not physically located in the folder; rather, they are in the repository and a folder references the objects that it contains.

Because folders do not physically contain their objects:

- A single object can be referenced by more than one folder at a time.
- Through searches, a file can be accessed independently of the folder(s) in which it is referenced.
- File security is enforced independently of folder security.

What Is Installed?

When Oracle *i*FS is installed, basic objects are created to get you started. Some of these objects include a folder hierarchy, commonly used ACLs and several users. The Web interface is ready for you to add users and files. XML configuration files are provided to create users, groups, and additional ACLs.

The following diagram shows the folder hierarchy as it looks out-of-the-box. A corresponding table describes the specific folders in the hierarchy.

```
ifs/
  system/
    security/
       users/
       groups/
       acls/
  webui/
  winui/
  apps/
```

```
   outbox/
   jsp-bin/
home/
public/
```

where:

Folder	Description
/ifs	Oracle *i*FS root folder that contains application-specific data and Oracle *i*FS system administrators' folders.
/ifs/system	Administrators' folders.
/ifs/webui	Required files for the Web interface.
/ifs/winui	Required files for the Windows interface.
/ifs/apps	Recommended directory for storing application files.
/ifs/outbox	System-wide applications outbox; used by any custom agents to send e-mail.
/ifs/jsp-bin	Java Server Pages (JSPs) that have been approved by the administrator for use. The JSPs will not execute unless they reside in this folder.
/home	By default, all users' home folders are created under this folder.
/public	A folder that all users can access; used to store files that are intended to be shared throughout the system.

Guidelines for Structuring the Folder Hierarchy

Follow these guidelines when planning your folder hierarchy.

- The top-level folder is called `root`, which represents the highest level where documents and folders can be stored.
- Folders and files can also be placed in multiple folders. By using multiple folders, the same files and folders can be organized differently, but only one copy exists in the repository. This optimizes the time it takes to create a folder hierarchy and provides a more flexible way for users to navigate the folders.
- You can create mount points for your folder hierarchy. Mount points in Oracle *i*FS are named sub-directories that can be accessed in the Windows and the Web interfaces as if they were top-level folders.

Planning Your Folder Hierarchy and Access Requirements

To plan effectively, you should consult subject matter experts and managers in your organization to determine how they conceptually organize their work. Managers can provide information on how files are used and who should have access to what.

Access Control Lists (ACLs) are associated with each object in the hierarchy. An ACL contains a list of users and groups who can perform certain actions on the folder, such as browsing its contents, or adding or deleting items from that folder.

When designing the access permissions for the folder hierarchy, note the following features:

- If a user does not have access to a folder, but the folder contains a file which the user does have access to, the user will not be able to find the file via browsing, but the user will be able to find the file via searching.
- If a file does not have an ACL specified, the file is specified as PUBLIC; therefore, that file can be accessed by all users. It is **strongly recommended** that default ACLs are specified in the Primary User Profile when users are created. Creating users using Oracle *i*FS Manager or through the Web interface sets the default ACLs to the ACLs you specify.

Default Folder ACLs

Folder security is defined by the ACL applied to it. The ACL applied to any newly created folder on the default ACL specified in the Primary User Profile of the user who created the folder. The ACL for a folder can be changed at any time by its owner once the folder has been created.

See Also

For more information on security, see Chapter 7, "Task 4: Set Up Security".

For more information on specifying defaults for creating users, see Chapter 5, "Task 2: Add Users and Groups".

Setting Up the Folder Hierarchy: The Steps to Follow

Follow these five steps to set up the folder hierarchy.

1. Define the Structure

The first step in setting up the folder hierarchy is deciding the hierarchy for the folder or directory tree. A folder can reside in multiple folders (have multiple parents), although this type of folder tree can be confusing to users as they browse through the tree. It is recommended to start with each folder descending from a single parent folder.

2. Name and Describe the Folders

Examine your folder hierarchy and decide on a name and description for each folder. It is important to use meaningful names and enter complete descriptions as users will use these keywords for searching.

3. Define Folder Security

The initial ACL for folders is the default ACL in the Primary User Profile of the user creating the folder tree. If security needs to be more finely grained, create new ACLs specific to users' requirements. Users can change a folder's ACL after creating it. The Web and Windows interfaces do not support files without an associated ACL.

4. Create the Hierarchy

Create the hierarchy by starting at the root of the tree, adding the folders, then creating each folder's child folders, working downward. You can create folders using Oracle *i*FS Manager, the Web or Windows interfaces, or through the Command Line Utilities.

5. Set the Mount Points

Mount points in Oracle *i*FS are named directories that can be accessed as if they were top-level folders through the Windows and Web interfaces. The important attributes of mount points, especially for your users, are:

- Their names, which will show up in Windows Explorer and the Directory Tree in the Web interface.
- The description, which is an optional free-form description of the mount point.

- The folder path, which is the location in the folder hierarchy where users will end up if they connect to that mount point.

As the administrator, you should limit the number of mount points you create because having more than a few dozen makes it very difficult for users to find what they need.

The following mount points are set up by default:

- Root
- Home
- Public

Creating and Deleting Folders Using Oracle *i*FS Manager

Using Oracle *i*FS Manager, you can only create folders one at a time and only from a single level.

The owner of the folders created is the user who is logged into Oracle *i*FS Manager.

To create folders:

1. Access the Browser by selecting the Browse option from the File menu.
2. Navigate to the directory where you want to place the new folder and select that directory.
3. Click the New Directory button on the toolbar and type a name for the folder in the dialog.
4. Click OK. The new folder displays in the Navigator.

Deleting a Folder Using Oracle *i*FS Manager

To delete a folder using the Browser:

1. Access the Browser by selecting the Browse option from the File menu.
2. Navigate to and select the folder you want to delete.
3. Click the Delete button on the toolbar.

Creating Folders Using the Command Line Utilities

To create several folders at a time, it is recommended that you use the Command Line Utilities and a batch file. Using the batch file, you create the folders and assign different ACLs.

To create folders:

1. Write a batch file using a text editor. For example:

   ```
   login system/<password>
   cd /public
   mkdir specifications
   mkdir specifications/development
   mkdir specifications/qa
   setattr /public/specifications ACL -avsystemacl public
   setattr /public/specifications/development ACL -avsystemacl public
   setattr /public/specifications/qa ACL -avsystemacl public
   ```

2. Run the batch file using the following command:

   ```
   $ORACLE_HOME/ifs<version>/bin/ifsshell -i batchfile.txt
   ```

3. To acknowledge that the folders were created, use any interface and browse to the /public directory. Drill down to see all folders.

Tips

To see if any errors occurred during the execution of your batch file, check the CupServer.log found in the following directory:

Platform	Directory
UNIX	`$ORACLE_HOME/ifs<version>/log`
Windows NT	`%ORACLE_HOME%\ifs<version>\log`

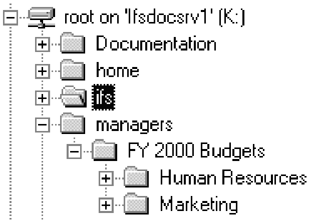
Creating Multiple Folders Using XML

Using XML, you can create multiple folders. When creating multiple objects, you must use the `<ObjectList>` tag. The following code creates multiple folders. This can be used when creating the folder hierarchy.

```
<ObjectList>
   <Folder>
       <Name> FY 2000 Budgets </Name>
       <FolderPath> /managers </FolderPath>
   </Folder>
   <Folder>
       <Name> Marketing </Name>
       <FolderPath> /managers/FY 2000 Budgets </FolderPath>
   </Folder>
   <Folder>
       <Name> Human Resources </Name>
       <FolderPath> /managers/FY 2000 Budgets </FolderPath>
   </Folder>
</ObjectList>
```

This XML example specifies that the folders are placed under the "managers" folder, therefore, this folder must already exist in Oracle *i*FS for the other folders to be created.

When using the Windows interface to load this XML file, you must drag and drop the XML file to any directory. In this example, the XML file must be placed in `/managers`. After this XML file has been loaded into Oracle *i*FS, the directory looks like:



All folders in this example are created in a single transaction, meaning, if there is an error, the complete transaction is rolled back, the folders are not created, and a log file is generated in the top directory. You can use this log file to troubleshoot errors.

Certain objects must be created in their own transaction. The following table lists the tasks that are required to be in their own XML file:

Task	Tag
Creating users	SimpleUser
Creating new types (subclassing)	ClassObject

See Also

For more information on loading files into Oracle *i*FS, see Chapter 6, "Task 3: Load Files into Oracle iFS".

Creating and Modifying Mount Points

You use mount points for two reasons:

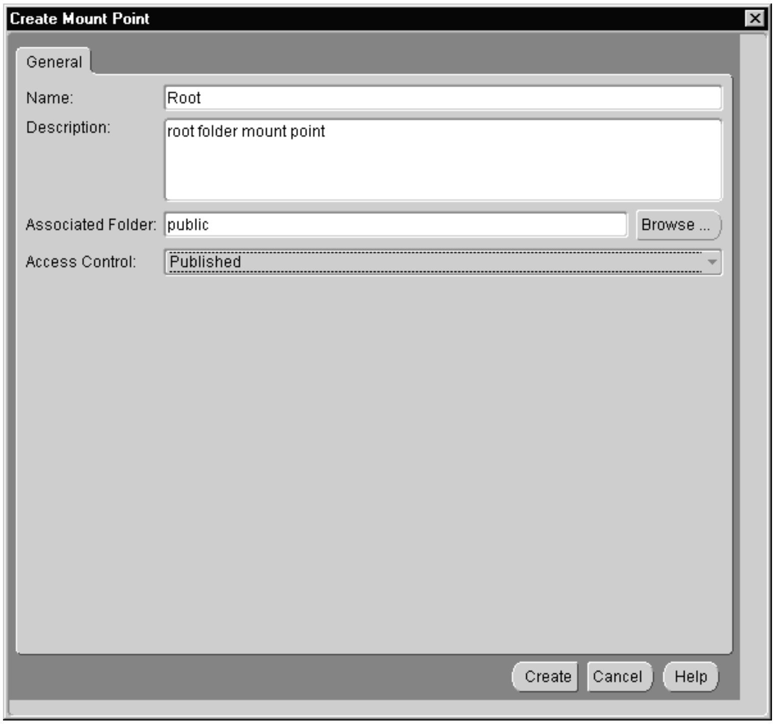
- Convenience—If a user always accesses the `Public/Docs/WIP` folder, he or she can use a mount point to start there instead of browsing through the hierarchy.
- Security—You might instruct users to log onto a specific point in the hierarchy.

Creating Mount Points

To create mount points:

1. Click Create on the toolbar.
2. Select Mount Point from the Select Object Type dialog. You can also use the Create Like option.

3. Click Create. The Create Mount Point dialog displays:



Name: Root

Description: root folder mount point

Associated Folder: public

Access Control: Published

4. Enter information in the following fields:
   - Name (required)—A meaningful name for the mount point.
   - Description—A description of the mount point.
   - Associated Folder (required)—The folder referred to by the mount point. For example, for the mount point "public," the associated folder is "/public."
5. Select the appropriate ACL for the mount point from the drop-down list.
6. Click Create.

   The new mount point displays in the Navigator.

Modifying Mount Points

The description and the ACL of the mount point may be modified.

1. In the Navigator pane, select the mount point by clicking its icon. The properties of the mount point display in the Detail View.
2. Enter a new description in the Description field. The mount point description is optional.
3. If you select a new ACL for this mount point, the ACL of the associated folder is also changed.
4. Click Apply.

Deleting Mount Points

Usually, you would delete a mount point if you do not want people to mount from Windows or navigate from the Web interface to a specific folder. If a mount point is deleted, a user will receive an error if they have the deleted mount point mapped which they access after it is deleted. To delete a mount point:

1. In the Navigator pane, select the mount point by clicking its icon.
2. Click Delete on the toolbar or select the Delete command from the Object menu.
3. Select OK to confirm the delete operation. Oracle *i*FS Manager refreshes the Navigator. The mount point no longer displays.

Tips

If there is no longer a need for a particular mount point because a folder in the associated folder path needs to be deleted, first delete the mount point.

Setting Up Folders for Users

If you create a user with Oracle *i*FS Manager, the Web interface, or XML, Oracle *i*FS automatically creates the user, home folder, e-mail folder, Primary User Profile, and e-mail profile.

Folder security is distinct from file security. Folder security defines who can browse the contents of the folder and add or delete folder items. File security defines who can access, modify, and delete attributes and contents of a file. If an ACL is not specified at the time the folder or file is created, then the ACL is determined by the user's default ACL. If a file is placed in a user's home folder, it does not mean the file is associated with the PRIVATE ACL, and if a file is placed in a public folder, it does not mean the file is associated with the PUBLIC ACL. The ACL associated with a folder or file can be modified at any time once it has been created to reflect appropriate access.

5

Task 2: Add Users and Groups

The second task involved in setting up Oracle *i*FS is to add users and define groups of users for ease of administration. This chapter steps you through determining which Oracle *i*FS administration tool to use and the basic steps to follow with each tool. The following topics are included:

- Understanding Users in Oracle iFS
- Set the Default User Definitions Using Oracle iFS Manager
- Create the Oracle iFS Users Using Oracle iFS Manager
- Using an XML Configuration File to Create Users
- Creating Groups Using Oracle iFS Manager
- Creating Groups with XML

Understanding Users in Oracle *i*FS

Out-of-the-box, most people will use Oracle *i*FS as an enhanced file system. To customize Oracle *i*FS for your organization, some of these users should have administrative permissions. To manage users, you need to manage information in both the Oracle *i*FS server and the Credential Manager.

User Profiles

Oracle *i*FS maintains Oracle *i*FS-specific user information in user profiles. Each active Oracle *i*FS user has a Primary User Profile, an e-mail user profile, and, optionally, extended user profiles:

- Primary user profile—Contains user information, such as the user’s home folder, default Access Control List (security), and quota control information.
- E-mail user profile—(optional) Contains e-mail management information.
- Extended user profile—(optional) Custom applications can define extended user profiles to contain application-specific profile information for a user. An e-mail user profile is an extended user profile.

Credential Manager Users

Oracle *i*FS uses a credential manager to authenticate users. A credential manager is an extensible authentication mechanism that determines the validity of a credential, such as a user name and password.

Each Oracle *i*FS user specifies the name of the credential manager used to authenticate that user, as well as the distinguished name that identifies that user to the credential manager.

A single default credential manager is created when Oracle *i*FS is installed. For each Oracle *i*FS user created, a corresponding entry, consisting of the user’s distinguished name and password, is stored in this default credential manager.

The default credential manager optionally requires users to be RDBMS users. If this feature is enabled, the Oracle *i*FS user named "jsmith" could only log into Oracle *i*FS if there was an RDBMS user named "jsmith." The credential manager only checks the user name. The RDBMS password does not have to match the Oracle *i*FS password.

To enable this feature, set the CredentialManagerIfsRdbmsUserMustExist property to true in the secondary properties file. This property is located in the following directory:

Platform	Directory
UNIX	`$ORACLE_HOME/ifs<version>/settings/oracle/ifs<version>/server/properties`
Windows NT	`%ORACLE_HOME%\ifs<version>\settings\oracle\ifs<version>\server\properties`

Note: Currently, Oracle *i*FS currently supports the default credential manager.

HTTP Authentication

HTTP authentication is a common mechanism for adding security to static Web pages. Oracle *i*FS uses its own security features as the basis for HTTP authentication. Files and folders need to have the PUBLISHED ACL applied for users to use a browser to access these objects without being forced to log in. If they are forced to log in, they must supply their Oracle *i*FS username and password.

Creating Users

Users can be created with all the Oracle *i*FS administration tools. Oracle recommends that you use Oracle *i*FS Manager, XML, or the Web interface. These tools create all objects associated with a new users with standardized settings. Creating users involves two steps:

1. Set the default user definitions in Oracle *i*FS Manager.
2. Create the user using Oracle *i*FS Manager, XML, or the Web interface.

Note: In the sections that follow, the Oracle *i*FS Manager is used to illustrate the procedures for setting up Oracle *i*FS.

See Also

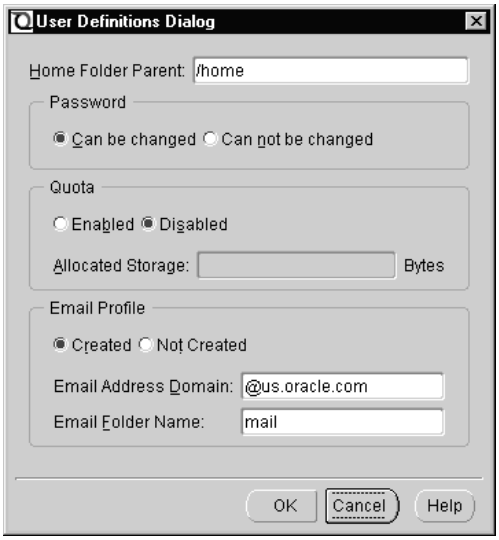
- For information on permission bundles, see Chapter 7, "Task 4: Set Up Security".
- For more information on the secondary properties, see Appendix B, "Secondary Properties".
- For more information on HTTP authentication, see the *Oracle Internet File System Developer's Guide*.

Set the Default User Definitions Using Oracle *i*FS Manager

Before creating a user, you can set the default user definitions which will be applied to all users you create. Using the user definitions provide a uniform way of creating users that is consistent with the various clients. If you do not set the user definitions, the defaults apply.

To set the default user definitions:

1. From the Menu bar, select Options from the Object menu.
2. Select User Definitions from the Options menu. The User Definition Dialog displays:



Home Folder Parent: /home

Password

● Can be changed ○ Can not be changed

Quota

○ Enabled ● Disabled

Allocated Storage: Bytes

Email Profile

● Created ○ Not Created

Email Address Domain: @us.oracle.com

Email Folder Name: mail

3. Specify the Home Folder Parent, the directory in which to store the users' home folders. The default is `/home`.

4. Specify whether or not users have the ability to change their passwords. You can override this setting on the Create User dialog.
5. Enable or disable quota control. If you enable quota control, type the storage size, in bytes, in the Quota Storage field. If you disable quota, the user has an unlimited quota. Users with administrative permissions usually have an unlimited quota amount.

   Enabling quota control limits storage content. The default allocated quota storage is 25 megabytes per user. If a user’s quota is reached, they will be unable to save documents in Oracle *i*FS.
6. Specify whether or not to create an e-mail profile. If Yes, type the e-mail address domain and the e-mail folder name in the fields provided. If you select No, these fields are disabled.

   The Email Address field is automatically populated when the users’ name is entered and consists of the `<username>@<domain>`. By default, the suffix for the users’ e-mail address is what you specify when setting the user definitions.

   The Email Folder field is a specified value that is the parent folder for all mailboxes, such as Inbox.
7. Click OK.

Tips

Quota checking is only completed when the Quota Agent is running and quota is enabled. Enabling a user’s quota without a running agent will allow users to go over their quota.

See Also

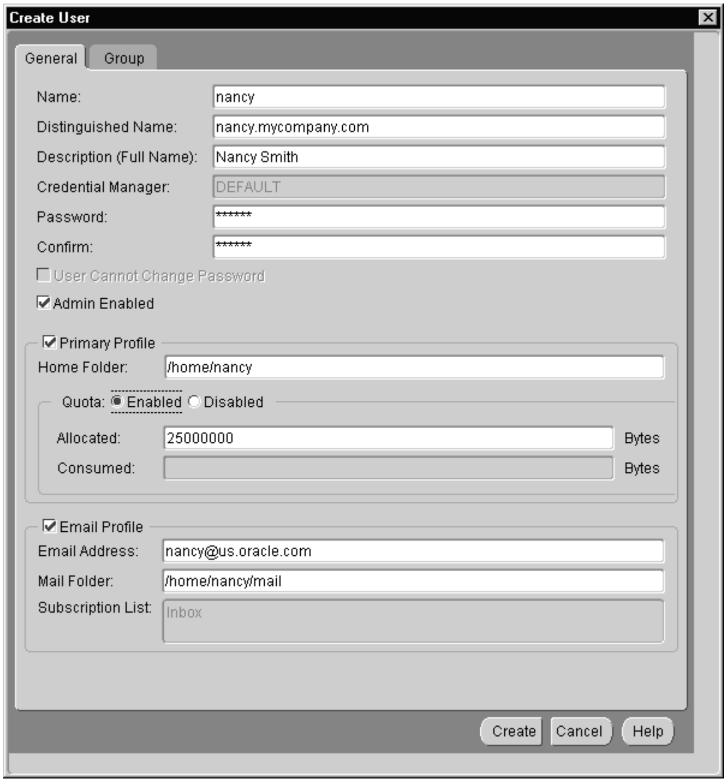
For more information on the Quota Agent, see Chapter 8, "Using Server Manager to Start and Stop Servers".

Create the Oracle *i*FS Users Using Oracle *i*FS Manager

To create an Oracle *i*FS user:

1. From the toolbar, select Create from the Object menu.
2. Select User from the Select Object Type window.

   The Create User dialog displays:



Name: nancy

Distinguished Name: nancy.mycompany.com

Description (Full Name): Nancy Smith

Credential Manager: DEFAULT

Password: ******

Confirm: ******

☐ User Cannot Change Password

☑ Admin Enabled

☑ Primary Profile

Home Folder: /home/nancy

Quota: ● Enabled ○ Disabled

Allocated: 25000000 Bytes

Consumed: Bytes

☑ Email Profile

Email Address: nancy@us.oracle.com

Mail Folder: /home/nancy/mail

Subscription List: Inbox

3. Enter information in the following required fields:

   - Name—The user name for the user you are creating. When you enter a user name, the Home Folder and Mail Folder fields are automatically populated based on the name you entered.
   - Distinguished Name—The fully qualified name in a domain used by Credential Manager for authentication. You usually enter user’s full name.

- Description (Full Name)—(optional) You can specify the full name of the user.
- Password and Confirm—The user's password. This can be changed by the user you are creating. Or, you can keep users from changing passwords by checking the box next to User Cannot Change Password.

4. If this user is to have administrative permissions, select the Admin Enabled checkbox.
5. The Primary Profile checkbox is checked as the default. If you do not want to create a Primary User Profile, unselect this checkbox.

   Each user has a Primary User Profile. This profile points to the user's home folder location and the user's default ACLs. The default ACLs determine which ACLs should be associated when the user creates different objects, such as folders and documents. By default, Oracle *i*FS sets the non-administrator user's default ACL to PUBLISHED (except for e-mail-related objects, such as messages. These have the PRIVATE ACL as the default), and sets the administrator's default ACL to PRIVATE (except ACL, PropertyBundle, DirectoryObject, VersionSeries, and VerionsDescription, which are PUBLISHED).
6. In the Home Folder field, enter a folder name if it is different from the user name.

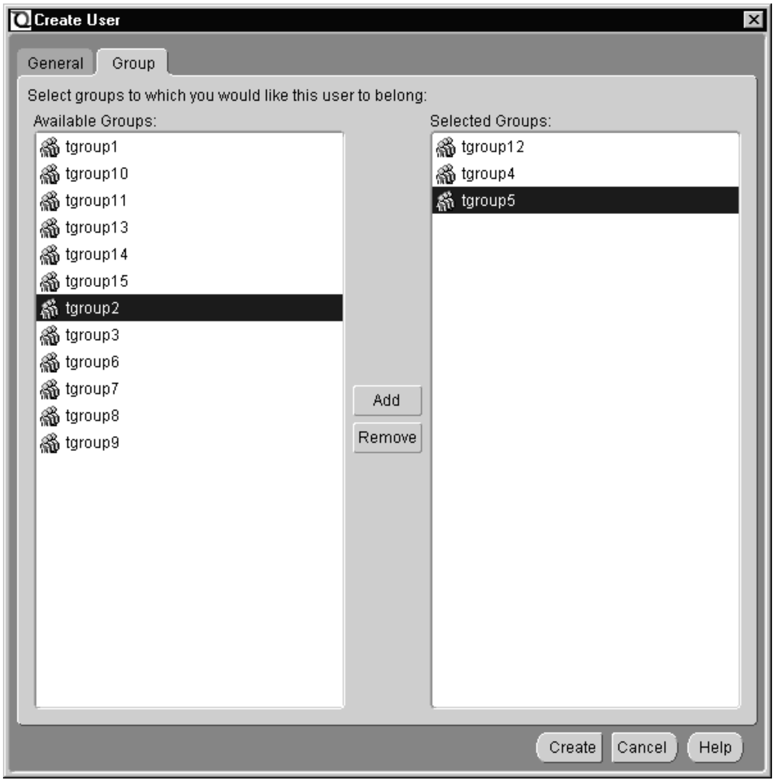
   A user's home folder is the default directory where a user starts when logging into Oracle *i*FS. Users can use their home folder as their personal workspace and to store their private files. A user's home folder can reside anywhere, so Oracle *i*FS Manager provides a dialog to change the default home folder location. The Web interface creates a user's home folder as the user's login name, under the `/home` directory. For example, `/home/jsmith`. The user's Primary User Profile points to the user's home folder location.
7. In the Quota section, select Enable or Disable to enable or disable quota for this user and override the quota established on the User Definitions Dialog. If you select Enable, enter the quota number of bytes in the Allocated field. If you select Disable, the user has an unlimited quota. Users with administrative permissions usually have an unlimited quota amount.

   If a user's quota is enabled, content storage is limited, by default, to 25 megabytes. Users cannot store content in Oracle *i*FS when this limitation is reached.

8. By default, the E-mail Profile checkbox is enabled. If you do not want to create an e-mail profile, unselect this checkbox. In the Mail Folder field, enter a folder name if it is different from the user name.

   The user's e-mail profile points to the user's e-mail folder location and specifies the user's e-mail address. Oracle *i*FS Manager and the Web interface create the user's e-mail folders under the home folder. For example, `/home/jsmith/mail/inbox`.

9. To add this user to existing groups, click the Group tab. The Select Group window displays:



Select groups to which you would like this user to belong:

Available Groups:

- tgroup1
- tgroup10
- tgroup11
- tgroup13
- tgroup14
- tgroup15
- tgroup2
- tgroup3
- tgroup6
- tgroup7
- tgroup8
- tgroup9

Selected Groups:

- tgroup12
- tgroup4
- tgroup5

10. From the Available Groups list, select the groups to which to add the user. Click the Add button to add the user to the selected group. The groups display in the Selected Groups list.

11. Click Create.

Using an XML Configuration File to Create Users

Creating users can be complicated since many other user-related objects, such as home folders, user profiles, and mail boxes are created at the same time. Using the `<SimpleUser>` tag within the XML file applies all the default definitions found in Table 5–1, "Additional User Definitions". You can change the user default settings in Oracle *i*FS Manager to suit your requirements. You do not need to specify every single value unless you want to override any of the default values. You override the defaults by explicitly setting them in the XML file. The values are case-sensitive.

Creating a single user and all its user-related objects are created in a single transaction. This includes a Directory User, Primary User Profile, Email Profile, Inbox, and a Home Folder are created in one step. Therefore, do not include creating any other objects besides users in your XML file. Also if you are creating 10 users in a single file, and an error occurs on the 7th user, the previous 6 users and their user related objects are not rolled back.

XML Example

The following XML file lists the definitions used to create users with an XML file. When using XML, you really only need to include the username and password, but this example displays other definitions you can include to override the defaults you set using Oracle *i*FS Manager. This example also shows you how to create users by parsing the XML file and checking that the users were created using the Command Line Utilities.

1. Create your user XML file:

```
<SimpleUser>
   <UserName>gking</UserName>
   <Password>ifs</Password>
   <DistinguishedNameSuffix>.yourcompany.com</DistinguishedNameSuffix>
   <AdminEnabled>true</AdminEnabled>
   <HomeFolderRoot>/home</HomeFolderRoot>
   <EmailAddressSuffix>@yourcompany.com</EmailAddressSuffix>
</SimpleUser>
```

2. Create the users by loading the XML file into Oracle *i*FS. Use the following command to load the XML file into Oracle *i*FS using the Command Line Utilities. You must be an administrator.

```
$ORACLE_HOME/ifs<version>/bin/ifslogin system/<password>
$ORACLE_HOME/ifs<version>/bin/ifsput users.xml
```

3. Confirm that the users were created:

```
$ORACLE_HOME/ifs<version>/bin/ifsls -class DirectoryUser
```

You can also drag and drop the user file into Oracle *i*FS through the Windows or Web interface and FTP. Oracle *i*FS invokes the XML parser and the users are created.

Additional User Definitions

The following table lists the complete set of user definitions you can set and a brief description. The definitions in this table can be set using an XML configuration file for creating users.

Table 5–1 Additional User Definitions

Definition Option	Default Value	Data Type	Description/Purpose
`UserName`	None	String	Required. A name for the user being created.
`Password`	None	String	Required. A password for the user being created.
`ShouldCreateCredentialManagerUser`	TRUE	Boolean	If the user exists.
`ReplaceCredentialManagerPassword`	FALSE	Boolean	If using an existing credential manager user, set true to override old password.
`CredentialManager`	DEFAULT	String	Which credential manager to use for access to the ’authentication engine.’
`AdminEnabled`	FALSE	Boolean	If user will have administration privileges.
`CanChangePassword`	TRUE	Boolean	If creating a guest user, this can be set to false.
`HasPrimaryUserProfile`	TRUE	Boolean	Whether to create a primary user profile for this user.
`HasHomeFolder`	TRUE	Boolean	Whether to create a home folder for this user.
`HomeFolderRoot`	"/home"	String	Root folder of the user’s home folder.
`HomeFolderHasPolicyBundle`	TRUE	Boolean	If administration privileges required to free home folder.
`HasContentQuota`	TRUE	Boolean	Whether to implement a content quota for this user.
`ContentQuotaAllocatedStorage`	25000000	Long	Storage space allocated to user in bytes.

Table 5–1 Additional User Definitions

Definition Option	Default Value	Data Type	Description/Purpose
`ContentQuotaEnabled`	FALSE	Boolean	Whether to enable or disable the implemented content quota.
`HasEmail`	TRUE	Boolean	Whether to create an email profile for this user.
`DistinguishedName`	None	String	If none specified, the DistinguishedName is constructed from username + suffix.
`PrimaryUserProfileName`		String	If none specified, the DistinguishedName is constructed from username + suffix.
`EmailUserProfileName`		String	If none specified, the DistinguishedName is constructed from username + suffix.
`ContentQuotaName`		String	If none specified, the DistinguishedName is constructed from username + suffix.
`HomeFolderName`		String	If none specified, the DistinguishedName is constructed from username + suffix.
`HomeFolderDescription`		String	If none specified, the DistinguishedName is constructed from username + suffix.
`DefaultAclsBundleName`		String	If none specified, the DistinguishedName is constructed from username + suffix.
`HomeFolderPolicyBundleName`		String	If none specified, the DistinguishedName is constructed from username + suffix.
`EmailAddress`		String	If none specified, the DistinguishedName is constructed from username + suffix.
`DistinguishedNameSuffix`	The value you enter as the distinguished name during installation is the default value.	String	Suffix of the distinguished name.
`PrimaryUserProfileNameSuffix`	"Primary Profile"	String	Suffix of the primary user profile name.
`EmailUserProfileNameSuffix`	"Email Profile"	String	Suffix of the email profile name.

Table 5–1 Additional User Definitions

Definition Option	Default Value	Data Type	Description/Purpose
`EmailAddressSuffix`	The value you enter as the distinguished name during installation is the default value with an "@" attached.	String	Suffix of the email address.
`ContentQuotaNameSuffix`	"Content Quota"	String	Suffix of the content quota name.
`HomeFolderNameSuffix`	""	String	Suffix of the home folder name.
`HomeFolderDescriptionSuffix`	"'s home folder"	String	Suffix of the home folder description.
`DefaultAclsBundleNameSuffix`	"DefaultACLs"	String	Suffix of the default ACL's name.
`HomeFolderPolicyBundleNameSuffix`	"Policy Bundle for Home folder and Inbox"	String	Suffix of the home folder policy bundle name.
`EmailSubfolderName`	"mail"	String	Name of the email subfolder.
`InboxName`	"inbox"	String	Name of the email inbox.
`DirectoryUserAcl`	Published	SystemAcl	ACL of a directory user.
`HomeFolderPolicyBundleAcl`	Published	SystemAcl	ACL of the home folder policy bundle.
`HomeFolderAcl`	Private	SystemAcl	ACL of home folder.
`DefaultAclsBundleAcl`	Published	SystemAcl	ACL of default ACLs bundle.
`ContentQuotaAcl`	Private	SystemAcl	ACL of the content quota.
`PrimaryUserProfileAcl`	Private	SystemAcl	ACL of the primary user profile.
`EmailUserProfileAcl`	Private	SystemAcl	ACL of the email profile.
`EmailSubfolderAcl`	Private	SystemAcl	ACL of the email subfolder.
`InboxAcl`	Private	SystemAcl	ACL of the email inbox.
`AclBundleAllPublished`	AclBundleAllPublished	Property Bundle	An out-of-an-the-box property bundle for the default ACLs of a non-administration user.
`AclBundleForAdmin`	AclBundleForAdmin	Property Bundle	An out-of-the-box property bundle for the default ACLs of an administration user.
`DefaultAcls`	AclBundleAllPublished	String	The property bundle to use for the defaults.
`DefaultAclsTable`	None	Hashtable	Customizing a property bundle for the default ACLs

Viewing Users

Using Oracle *i*FS Manager, you can view the properties of existing Oracle *i*FS users.

To display all existing users:

- Click the user icon in the Navigator. The tree expands to display all users.

To display user properties:

- Navigate to the user whose properties you want to display and select that user by clicking it. The properties display in the Detail View.

Changing a User's Password

It may be necessary to change a user's password as you administer Oracle *i*FS. For example, if a user forgets his or her password, this is how to reset it.

1. Navigate to the user whose password you want to change and click that user. The user's properties display in the Detail View.
2. In the Password field, type a new password.
3. Confirm the password by typing it again in the Confirm field.
4. Click Apply.

See Also

For information on changing the `ifssys` password, or if you have lost the password, see Chapter 13, "Oracle iFS Log Files and Troubleshooting Information".

Changing a User's Home Folder

It may be necessary from time to time to change a user's home folder. For example, if the home directory is changed or moved, you must change their home folder directory.

1. Navigate to the user whose home folder you want to change and click that user.
2. In the Home field, type a new location for the user's home folder.
3. Click Apply.

Changing a User's Default ACL for Documents

It may be necessary to change a user's default ACL. To do this:

1. Navigate to the user whose default ACL you want to change and click that user.
2. Click the Default ACL drop-down list and select a new default ACL.
3. Click Apply.

Changing a User's E-mail Address

It may be necessary to change a user's e-mail address, for example, if their name changes.

1. Navigate to the user whose e-mail address you want to change and click that user.
2. In the Email Address field, type a new e-mail address for this user.
3. Click Apply.

Changing a User's Quota Control

It may be necessary to change a user's quota control. For example, some users need more space than others.

1. Navigate to the user whose quota control you want to change and click that user.
2. Select to either disable or enable the user's quota control. If you decide to enable quota control, enter the number of bytes allotted for this user.
3. Click Apply.

If a user's quota is enabled, content storage is limited, by default, to 25 megabytes. Users cannot store content in Oracle *i*FS when this limitation is reached.

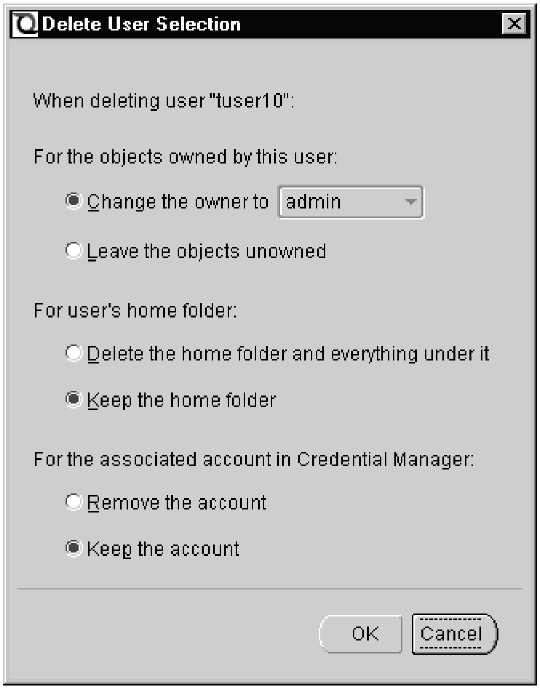
Deleting Users

You can delete users using Oracle *i*FS Manager. When you delete a user, you can change the ownership of the objects owned by that user and specify if the user's home folder is to be deleted.

To delete a user:

1. Navigate and select the user you want to delete. The user's properties display in the Detail View.
2. Click the Delete button on the toolbar or select Delete from the Object menu. You can also click the Delete button on your keyboard.

   The Delete User Selection dialog displays:



Delete User Selection

When deleting user "tuser10":

For the objects owned by this user:

● Change the owner to admin

○ Leave the objects unowned

For user's home folder:

○ Delete the home folder and everything under it

● Keep the home folder

For the associated account in Credential Manager:

○ Remove the account

● Keep the account

3. If you want to change the ownership of the objects owned by the user you are deleting, select Change the owner to and select the new owner from the drop-down list. If you do not select to change the ownership of the objects owned by the users, those files become unowned; their owner attribute is null. Any user with administrative permissions can own these files.
4. If you want to delete the user's home folder, select Delete the home folder and everything under it. Note that everything under the user's home folder is deleted as well.
5. If you want to delete the associated Credential Manager account, select Remove the account. It is recommended that you do this only if you created it.

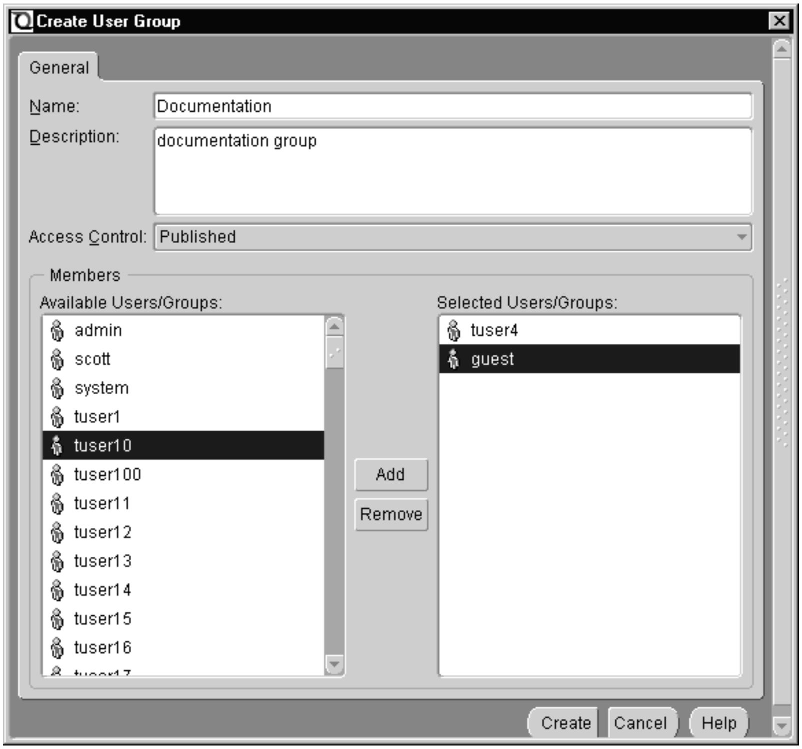
6. Click OK to delete the user. Confirm the delete operation by selecting Yes.

Creating Groups Using Oracle *i*FS Manager

By assigning users to groups, you make administration and maintenance easier. Instead of adding each user to an Access Control List for a file or folder—a time consuming task—you can add a group of users all at the same time. Oracle *i*FS is shipped with world, a default group. When users are created, they are automatically added to this group. To create a group, you define the group itself, then populate it.

To create a group:

1. On the toolbar, click Create.
2. Select User Group and click Create. The Create User Group window displays:



Name: Documentation

Description: documentation group

Access Control: Published

Members

Available Users/Groups: admin, scott, system, tuser1, tuser10, tuser100, tuser11, tuser12, tuser13, tuser14, tuser15, tuser16, tuser17

Selected Users/Groups: tuser4, guest

3. Type a name for the group.

4. Type an optional description. The description displays only when you modify the group.
5. Select the ACL for the group from the drop-down list.

   The default ACL of the user that created this group is assigned to any new group you create. Therefore, a group with the PUBLISHED ACL cannot be edited by any user. When creating groups, it is important to discuss with department managers and other users if they want all Oracle *i*FS users to have read-only access to the groups you are creating for their respective departments. If you need to change the default ACL, you can do so by modifying the ACL for a specific group.
6. From the Available Users / Groups list, select the users and groups to add to the group. Click the Add button to add the user or group to the group.
7. Click Create. The new group displays in the Navigator.

Viewing Groups

Using Oracle *i*FS Manager, you can view existing Oracle *i*FS groups and their properties.

To display all existing groups:

- Click the group icon in the Navigator. The tree expands to display all groups.

To display group properties:

- Navigate to the group whose properties you want to display. The properties display in the Detail View.

Adding Users and Groups to the Target Group

The group you want to add users and groups to is called the target group.

1. Navigate to the target group for which you want to add users and groups and click the group to select it. The group properties display in the Detail View.
2. Select the users and groups from the Available Users / Groups list to be added and click the right arrow button. The selected users and groups display in the Selected Users / Groups list.
3. Click Apply.

Removing Groups and Users from the Target Group

1. Navigate to the target group for which you want to remove groups or users and click the group to select it. The group properties display in the Detail View.
2. Select the users and groups from the Selected Users/Groups list to be removed and click the left arrow button. The selected users and groups display in the Available Users/Groups list.
3. Click Apply.

Renaming Groups

To rename an existing group:

1. Navigate to the group you want to rename and select it by clicking its icon. The group properties display in the Detail View.
2. Enter a new name in the Name field. You can also enter a new description in the Description field. The group description is optional.
3. Click Apply.

Deleting Groups

You can delete groups using Oracle *i*FS Manager. Any user with the correct permissions can delete groups as long as they have the correct permissions.

To delete a group:

1. Navigate to the group you want to delete and select it by clicking its icon.
2. Click Delete on the toolbar or select the Delete option from the Object menu.
3. Select OK to confirm the delete operation. Oracle *i*FS Manager refreshes to display the changes.

Creating Groups with XML

This example XML file creates groups. You can use this file, although you must modify the name of the group to match the name of the group you are creating. When adding users to the group, the user names must exist. If you use FTP or CUP to load the XML file into Oracle *i*FS, it is parsed and not saved anywhere in Oracle *i*FS.

```
<DIRECTORYGROUP>
  <Name>DemonstrationUsers</Name>
  <Members>
    <REF reftype='name'>msmith</REF>
    <REF reftype='name'>mallen</REF>
    <REF reftype='name'>sward</REF>
    <REF reftype='name'>rjones</REF>
    <REF reftype='name'>tmartin</REF>
    <REF reftype='name'>dblake</REF>
    <REF reftype='name'>eclark</REF>
    <REF reftype='name'>sscott</REF>
    <REF reftype='name'>gking</REF>
    <REF reftype='name'>tturner</REF>
    <REF reftype='name'>jadams</REF>
    <REF reftype='name'>pjames</REF>
    <REF reftype='name'>gford</REF>
    <REF reftype='name'>amiller</REF>
  </Members>
</DIRECTORYGROUP>
```

6

Task 3: Load Files into Oracle *i*FS

This chapter provides information on the different protocols and methods you can use to load files into Oracle *i*FS. The following topics are included:

- The Four Ways to Load Files into Oracle iFS
- Drag and Drop Files into Oracle iFS
- FTP Your Files into Oracle iFS
- Upload Files Using the Web
- Upload Files Using the Command Line Utilities
- Versioned Vs. Non-Versioned Files
- Deleting Older File Versions
- Migrating Your Data

The Four Ways to Load Files into Oracle *i*FS

This section discusses the four possible ways of loading documents into Oracle *i*FS. Remember to set the default ACL on your user ID to ensure that loaded files have the correct security.

Drag and Drop in the Windows and Web Interfaces

In both the Windows Explorer and the Web interface, you can drag and drop files into Oracle *i*FS. When you load files using either SMB or drag and drop in the Web interface, the XML files are parsed by default. If you do not want the XML files to be parsed, you must use another protocol to load your files.

Use FTP to Load Files

While running the FTP server, you can use any FTP client to load files into Oracle *i*FS. Using the FTP protocol is the most efficient way to perform bulk file loading.

When you load files using FTP, XML files are parsed by default. You can turn off the parsing mode by using a quote command.

Browse and Upload Using the Web Interface

Using the Web, choose Upload by Browse and select the file to upload. You can select to parse the file and apply an ACL. This is an efficient tool because it can be used from any machine and does not require any client installation except for a browser.

Use the Command Line Utilities

You can use the Command Line Utilities to load files into Oracle *i*FS by using the `ifsmv` command or the `ifsput` command. When you load files into Oracle *i*FS using the Command Line Utilities, XML files are parsed by default. You can turn off the parsing mode by using the `ifsmode parse off` command.

See Also

- For more information on using the Command Line Utilities to load files into Oracle *i*FS, see Appendix A, "Command Line Utilities Reference".

Drag and Drop Files into Oracle *i*FS

In Windows:

1. Select the files from your local directory.
2. Drag and drop the files into a folder on the Oracle *i*FS drive.

In the Web interface:

1. Navigate to the folder where you want to store the file.
2. Click the Upload icon and choose Via drag and drop. An Upload window is displayed with the current folder information.
3. Select one or more files, then drag and drop them into the Upload window.
4. Click OK to confirm that you want to upload the files. If you are uploading multiple files, a progress window appears and lists each file as it is transferred to Oracle *i*FS.
5. You can continue to drag files into the Current Directory window to transfer additional files. When you are finished uploading files, close the Current Directory window.

Note: You cannot upload folders using this method. You must explicitly create the folders by navigating to where you want the new folder, then clicking the New icon and choosing Folder. In the dialog box, type the name for the new folder.

FTP Your Files into Oracle *i*FS

To FTP your files into Oracle *i*FS, the following requirements must be met:

- An FTP client must be installed on your local machine.
- You must know the port number for FTP, which is chosen during configuration.
- An account on the Oracle *i*FS server must be already set up.

To FTP your files into Oracle *i*FS:

1. Using an FTP client, log into the Oracle *i*FS server. You can use either the hostname of the Oracle *i*FS server or its IP address.

2. Navigate to the files on your local machine that you want to load into Oracle *i*FS and use your FTP client's utilities to load the files.

> **Note:** If you have trouble logging into Oracle *i*FS using your FTP client, make sure to enter the correct port, which is chosen during configuration of Oracle *i*FS.

Upload Files Using the Web

Using the Web interface, you can only load one file at a time:

1. Navigate to the directory where you want to store the file.
2. Click the Upload icon and select Via Browse.
3. In the Upload window, click the Browse button.
4. Locate the file you want to upload and click Open.
5. In the Options section, the following options are available:
   - Apply ACL—Select the ACL you want to apply to the file from the drop-down list. You must use an existing ACL. The default document ACL is PRIVATE.
   - Language—Select the language for this file. This option cannot be used to convert from one language to another.
   - Character Set—Select the character set for this file.
6. In the Upload window, click the Upload button.

Tips

When you want to view a file using the Web interface that is not recognized, a new navigator window is opened and a dialog prompts for you to pick the application, save the file, or cancel this operation.

Upload Files Using the Command Line Utilities

To browse and upload files using the Command Line Utilities:

1. Login into Oracle *i*FS using the following command:

   ```
   ifslogin system/<password>
   ```

2. Browse the repository to select the file you want to upload. Use the following command:

   ```
   ifscd /public
   ```

3. Load the selected files. Use the following command:

   ```
   ifsput myfile.doc
   ```

Versioned Vs. Non-Versioned Files

Versioning is a way of keeping previous copies of files available in case you need a history of the changes. Whenever you check in a versioned file, Oracle *i*FS automatically creates a new version and keeps a copy of the older file. For example, if you make a change to a memo and save it, then decide that you made a mistake, you can retrieve the old version instead of trying to reconstruct it from memory.

Once a file has been versioned, users can check out the file, replace it with an updated version, then check in the file. Each time the file is checked in, a new version is created and stored.

Using versioned files takes up more disk space because a copy of each version is saved on your system. Older versions also count against a user’s quota.

The benefits of versioning, such as maintaining a history of a file, need to be weighed against the disk space and quota requirements.

As an administrator, you have the option to purge older versions, assuming they are no longer needed to help manage your disk space.

See Also

For more information on versioning a file, see the *Oracle Internet File System User's Guide*.

Deleting Older File Versions

Often, you will need to delete out-dated versions of files to free up space. You can perform this task using either the Web or Windows interface. You do not need to be an administrator to perform this task.

Using the Web Interface to Delete Old Versions of Versioned Files

To delete old versions of versioned files:

1. Navigate to the versioned file you want to delete and select it by checking its Select box.
2. Click Edit on the toolbar and select History.
3. Select one or more old versions and click Delete.

Using the Windows Interface to Delete Old Versions of Versioned Files

To delete old versions of versioned files:

1. Navigate to the versioned file you want to delete and select it.
2. Right-click file, and select Properties.
3. On the Properties dialog box, select the Oracle *i*FS Versions tab.

   Select the version you want to delete and click Remove.

Migrating Your Data

You may want to migrate data from an old file system to Oracle *i*FS.

1. Log into Oracle *i*FS through Microsoft Windows as an administrator.
2. Click and drag the entire directory structure from the original file system into Oracle *i*FS,

   -or-

   Drag different parts of the directory tree separately, confirming that each part of the tree has been copied before copying the next one.

3. Log into Oracle *i*FS through Windows as a test user and verify that all the files and folders are accessible.
4. From the Start menu, select Oracle *i*FS from the Find option and perform a series of searches—on both filename and contents—to ensure that users can find files easily.

7

Task 4: Set Up Security

This chapter provides information on setting up user, folder, and document security. Topics include:

- Overview of Security in Oracle iFS
- Users and Groups as Grantees
- Primary User Profiles and Default ACLs
- Creating a Custom ACL
- Modifying ACLs
- Deleting ACLs
- Using the Web Interface to Apply ACLs to a Folder or File
- Using the Windows Interface to Apply an ACL
- Using XML to Create an ACL

Overview of Security in Oracle *i*FS

Security for all objects, including folders and files, is maintained through Access Control Lists (ACLs) and Access Control Entries (ACEs). An ACL has a name and a set (one or more) of ACEs.

The System ACLs

There are four system ACLs for you to apply to your documents and folders. These ACLs cover the most common security needs in most shared environments. The system ACLs are listed and described in the table below.

ACL	Description
Private	Grants no permissions to any other user. Other users cannot view, modify, or delete a user's document in any way, unless changed by the owner.
Protected	Enables other users to see the files in the folder, add documents and folders to the folder, and remove documents and folders they have created from the folder, but are not allowed to delete the folder itself.
Public	Allows full access to the item. All users can make any changes that the owner can make.
Published	Allows other users to view the contents, but they are not allowed to modify or delete the document.

Note: You should not modify these system ACLs.

All objects may have an associated ACL. If no ACL is assigned, the object is PRIVATE (only the owner has full access to it). If a default ACL is specified in a user's Primary User Profile, that ACL is assigned to the new public object created by that user.

Similar to files, ACLs have an ACL associated with them to manage the users who can use them or modify them. System ACLs can be created and modified by system administrators only.

Working with ACEs

Each ACE is comprised of a:

- Grantee—A grantee is a specific Oracle *i*FS user or group.
- Set of Permissions—The permissions granted to the user or group, or revoked from the user or group.
- Action—Specifies the action to be taken, either Grant or Revoke.

For each user or group, you need to decide which of the permission bundles will be granted to them. Permission bundles are sets of permissions that are applied to users. The set of permission bundles assigned to a specific user or group is called an Access Control Entry (ACE).

ACLs give you the flexibility to grant the permission bundles your co-workers require while helping to protect your files against accidental modification or unauthorized access.

By setting the permission bundles for an individual or group, you create an Access Control Entry (ACE) in your Access Control List (ACL). It’s possible to create a different ACE for every user in your Oracle *i*FS repository for every file you control, but, in most cases, the standard ACLs that come with Oracle *i*FS should meet your needs, and the remaining cases can be handled by creating an ACL.

When you define an ACE, the first choice you must make is whether the ACE you’re creating is going to grant the permission bundles you select, or revoke them. If you are going to give only a few permission bundles to a selected user or group, choose Grant, and select only the permission bundles that they will be given. If you are going to give the selected user or group general access with only a few restrictions, choose Revoke. Then explicitly grant those permission bundles to the grantee.

Working with Permission Bundles

A permission bundle is a collection of permissions. Permission bundles are convenient because you do not have to specify the individual permissions. For example, the permission bundle, "All," is a collection of permissions which grants all possible permissions. A permission bundle can be granted to a user or group, or revoked from a user or group. Oracle *i*FS Manager, the Web interface, and the Windows interface support ACLs and ACEs with permission bundles.

The following table lists the permission bundles used in Oracle *i*FS and the permissions they contain. The ClassCreate and ClassRestrictedCreation permission bundles are used exclusively for ACLs associated with class objects and are also known as ClassAccessControlLists.

Permission Bundle	Description	Permissions
All	All permissions are included.	Discover, SetAttribute, GetContent, SetContent, Delete, Lock, Grant, AddMember, RemoveMember, AddItem, RemoveItem, AddRelationship, RemoveRelationship, AddVersionSeries, RemoveVersionSeries, AddVersion, RemoveVersion, SetDefaultVersion, SetPolicy
Read	User can find and read the content of the object.	Discover, GetContent
Delete	Users can find and delete the object.	Discover, Delete
Modify	Users can find and modify the object.	Discover, SetAttribute, GetContent, SetContent, Lock, AddMember, RemoveMember, AddItem, RemoveItem, AddRelationship, RemoveRelationship, AddVersionSeries, RemoveVersionSeries, AddVersion, RemoveVersion, SetDefaultVersion, SetPolicy
Protected	To be used for folders; users can find the folder and add or delete folder items, but they cannot modify the folder.	Discover, AddItem, RemoveItem
ClassPublic	To be used with class object; users can create instances of this class and search them.	Create, SelectorAccess
ClassRestrictCreation	To be used with class object; users can access instances of this class.	SelectorAccess

The permissions bundled in the above permission bundles are listed and defined in the table below. "Item" can be either a file or directory, unless stated otherwise.

Permission	Definition
Add Item	Grantee can add an item to a folder.
Add Member	Grantee can add a member to a group.
Add Relationship	Grantee can create a link to the item in another folder.
Add Version	Grantee can update a versioned file.
Add Version Series	Grantee can change a file to a versioned file.
Create	Grantee can create instances of a class object. Applies only to ClassAccessControlLists.
Delete	Grantee can delete the object.
Discover	Grantee can browse to the item through a Directory Tree, and can find the item using Find operations.
Get Content	Grantee can view and copy the content of the file.
Grant	Grantee can modify the ACL assigned to the item to grant additional permissions.
Lock	Grantee can lock and unlock the object.
Remove Item	Grantee can delete an item from the folder.
Remove Member	Grantee can remove a member from a group.
Remove Relationship	Grantee can delete an existing link to the item in another folder.
Remove Version	Grantee can remove a version of a versioned file.
Remove Version Series	Grantee can remove a version series from a family. Permission applies to the family.
Selector Access	Grantee can search a class in either a search or selector. Applies only to ClassAccessControlLists. Selector access is governed by the class’s ClassACL.
Set Attribute	Grantee can change any of the item’s attributes.
Set Content	Grantee can change the contents of the file.
Set Default Version	Grantee can change the default version of a version series of a family or version series.

Permission	Definition
Set Policy	Grantee can change a policy property bundle associated with a public object. Permission applies to the policy property bundle.

Oracle *i*FS only allows the creation of new permission bundles using XML.

Users and Groups as Grantees

Groups and ACLs provide a powerful mechanism for managing access to objects. Assume a group is defined as a grantee in an ACE. If the group membership is changed, either by adding or deleting a member, those changes are reflected automatically in access to the object. If the group is granted permissions on an object and a new member is added to the group, that new member automatically has access to the object the on which the group has permission. The order of the ACEs is significant. The ACL is resolved in the order of the ACEs.

Users can also be specified as a grantee in an ACE. For example, if the user "jsmith" is part of the group "ifsdev." The ACL results in everyone in "ifsdev" having full access, except "jsmith," who cannot delete the object.

- ACL Name—IFSDEV ACL
- ACL Description—Grants read access to "ifsdev" and, in addition, allows "jsmith" the delete permission.
- ACEs (or ACL):

Grantee	Permission Bundle	Grant or Revoke
IFSDEV	Read	Grant
JSMITH	Delete	Grant

Note: If the order of the grantees is reversed, the ACL in the example above would resolve to "jsmith" having only DELETE permissions.

Primary User Profiles and Default ACLs

Each user has a Primary User Profile. One of the attributes of this profile is a set of default ACLs. These ACLs determine the default ACL associated with all objects the user creates. Default ACLs can be:

- Explicit—An ACL for each class.
- General—One ACL for a public object and its subclass.

Default ACLs are used at the time an object is created, unless the creator specifies a different ACL. The ACL can be changed at any time after the object has been created.

When an object has no ACL, the owner and administrator have access to it. Therefore, it is strongly recommended that a Primary User Profile with the default ACLs specified is created for each user. Creating a user with Oracle *i*FS Manager, the Web interface, or XML creates a user profile with the following default ACLs:

- Non-administrator user—PUBLISHED, except Mailbox, Mail Document, Mail Folder, and Message, which are PRIVATE.
- Administrator user—PRIVATE, except ACL, Property Bundle, Directory Object, Version Series, and Version Description, which are PUBLISHED.

Creating a Custom ACL

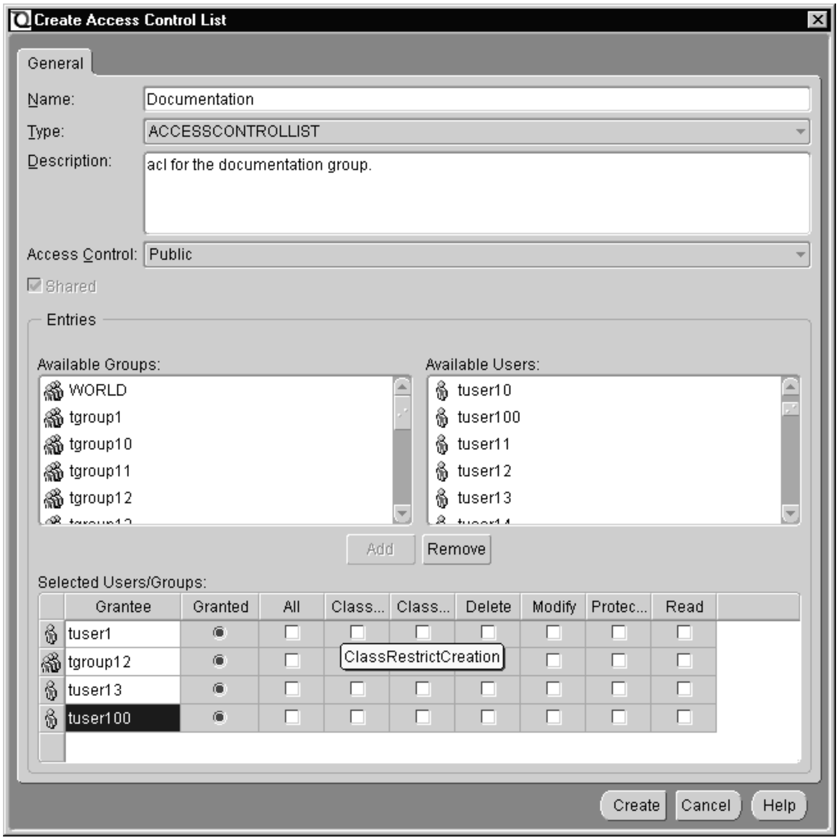
If you want to create a batch of ACLs, use the Create Like option. This creates an ACL similar to an existing ACL, which you can then customize. To apply an ACL, to a file or folder, use the Web or the Windows interface.

Note: You do not need administrative permissions to define ACLs, therefore, users can also define ACLs.

To create a custom ACL:

1. On the Oracle *i*FS Manager toolbar, click Create.

2. Select Access Control List from the Select Object Type window and click Create. The Create Access Control List window displays:



Name: Documentation

Type: ACCESSCONTROLLIST

Description: acl for the documentation group.

Access Control: Public

☑ Shared

Available Groups:

- WORLD
- tgroup1
- tgroup10
- tgroup11
- tgroup12

Available Users:

- tuser10
- tuser100
- tuser11
- tuser12
- tuser13

Selected Users/Groups:

Grantee	Granted	All	Class...	Class...	Delete	Modify	Protec...	Read
tuser1	●	□	□	□	□	□	□	□
tgroup12	●	□	ClassRestrictCreation			□	□	□
tuser13	●	□	□	□	□	□	□	□
tuser100	●	□	□	□	□	□	□	□

3. In the Name field, type a name for the ACL.

4. Select the ACL type from the drop-down list. The choices are:

   - AccessControlList—Any custom ACL created.
   - SystemAccessControlList—Generally for system-wide use.
   - ClassAccessControlList—Used only for class objects.

5. In the Description field, type an optional description for the ACL. The description is used to describe the type of ACL.

6. Select an ACL for this ACL from the drop-down list. Because the ACL itself is a file, it needs access security.
7. Select the groups or users for whom you want to create ACEs. Hold down the Control key to select multiple items. Click the Add button to add the groups or users to the Selected Users / Groups list.
8. Associate a permission bundle with each grantee by selecting the corresponding checkbox.

   Each of the users or groups you selected can have a separate set of ACEs associated with it. For each, select Grant to grant the permission bundles you select. If Grant is not selected, the permission bundles you select are revoked from that user or group. Scroll right to see all of the available permission bundles.
9. Click Create. The ACL displays in the Navigator.

See Also

For more information on creating and applying ACLs, see "Using the Windows Interface to Apply an ACL".

Modifying ACLs

To modify ACLs, use the Detail View on the Oracle *i*FS Manager. To display the ACL you want to modify:

1. In the Navigator, under Administrator's tasks, click Access Control Lists.
2. Click the ACL you want to modify. The properties of the ACL display in the Detail View.
3. You can only perform the following tasks:
   - Modify the ACL description and the ACL applied to this ACL.
   - Add or remove entries (ACEs) and modify their permissions.
4. Click Apply to apply changes. Click Revert to restore the original ACL properties.

Note: Modifying system or class ACLs is not recommended.

Deleting ACLs

To delete an ACL, select the ACL to be deleted and do one of the following:

- Click the Delete button on the toolbar.
- On the menu bar, select Delete from the Object option.

Confirm the delete operation by selecting Yes in the confirmation dialog box.

Using the Web Interface to Apply ACLs to a Folder or File

To apply an ACL to a file or folder:

1. Navigate to the folder or file to which you want to apply the ACL.
2. Click the Select checkbox to the left of the file or folder. You can select more than one item at a time.
3. Click Edit and select Apply ACL. A dialog displays with a list of all system ACLs and custom ACLs you have added.
4. Select the ACL you want to apply to the selected file(s) or folder(s).
5. Click OK.

   A success message displays to tell you that the ACL has been applied to the items.
6. Click OK.

Using the Windows Interface to Apply an ACL

To apply an ACL to a file or folder:

1. Navigate to the file or folder for which you want to apply an ACL.
2. Right-click the file, then select Properties.
3. On the Properties dialog box, click the Oracle *i*FS Security tab.
4. To apply an existing ACL, click the Use Existing ACL and select the ACL from the window.
5. Click Use ACL. The ACL is ready to be associated with the file or folder.

   Click Apply or OK to associate the ACL with the file or folder.

Using XML to Create an ACL

Through XML, you can create an ACL using the existing permission bundles. In addition, you can also create your own custom permission bundle. Once you have created your own permission bundle, you can use Oracle *i*FS Manager, the Web interface, or XML to create ACLs using the custom permission bundle.

Example

To create your own custom permission bundle:

```
<PermissionBundle>
      <Name> MyPermissionBundle </Name>
      <AccessLevel>
            <Discover> true </Discover>
            <GetContent> true </GetContent>
            <SetContent> false </SetContent>
      </AccessLevel>
</PermissionBundle>
```

To create an ACL using the custom permission bundle:

```
<AccessControlList>
  <Name> MyAcl </NAME>
  <Description> Custom ACL using custom permission bundle </Description>
  <ACEs>
    <AccessControlEntry>
      <Grantee classname='directorygroup' Reftype="name"> ifsdev </Grantee>
      <Active> true </Active>
      <Granted> true </Granted>
      <PermissionBundles>
         <PermissionBundle Reftype='name'> MyPermissionBundle </PermissionBundle>
      </PermissionBundles>
    </AccessControlEntry>
  </ACEs>
</AccessControlList>
```

8

Using Server Manager to Start and Stop Servers

Server Manager is a tool that provides managing and monitoring capabilities for the Oracle *i*FS servers and agents. Custom agents can be written to perform specific tasks on behalf of a user or the system. Topics include:

- Server Manager: An Overview
- Managing Server Manager with Oracle iFS Manager
- Using the Command Line to Manage Server Manager
- Oracle iFS Agents
- Definition Files for Protocol Servers

Server Manager: An Overview

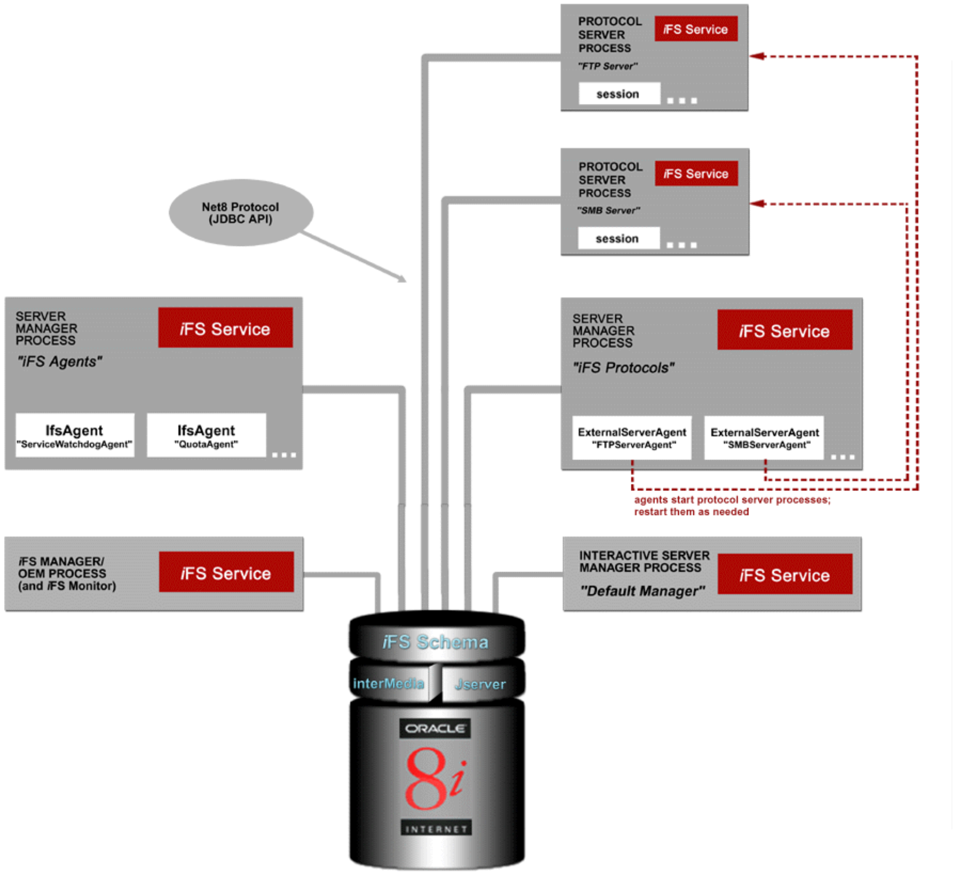
Using Server Manager, you can start, stop, suspend, resume, and list information for the various servers and agents. You can also use Server Manager to manage the servers and agents remotely from other machines from which the agents and servers may be running. In this way, it is possible to have protocol servers running on different machines, yet managed from a single Server Manager. Server Manager itself is a server, so one Server Manager can manage any other Server Managers in the system.

The default installation is configured for three pre-defined servers to run with Server Managers:

- `IfsAgents`—Loads and manages all agents; runs in the background.
- `IfsProtocols`—Loads and manages External Server agents that start up protocol servers; runs in the background.
- `DefaultManager`—Used to monitor the agents and servers in the system; runs in interactive mode.

In the following diagram, three Server Managers are displayed. The first Server Manager is managing the protocol servers; the second is managing the agents; and the third Server Manager is running interactively, monitoring the servers. In addition, Oracle *i*FS Manager is running. These Server Managers can be running on either the same or on different machines. All three Server Managers are able to monitor any registered server or agent running in the system.

Figure 8–1 Oracle iFS Server Manager Diagram



Another important tool are the log files. The log files detail the activity on the protocol server. Some servers have the option to output verbose logging. As the administrator, you will need to manage these files because over time, these files can become quite large depending upon the activity and type of logging selected. By default logging will be appended to the current log.

Managing Server Manager with Oracle *i*FS Manager

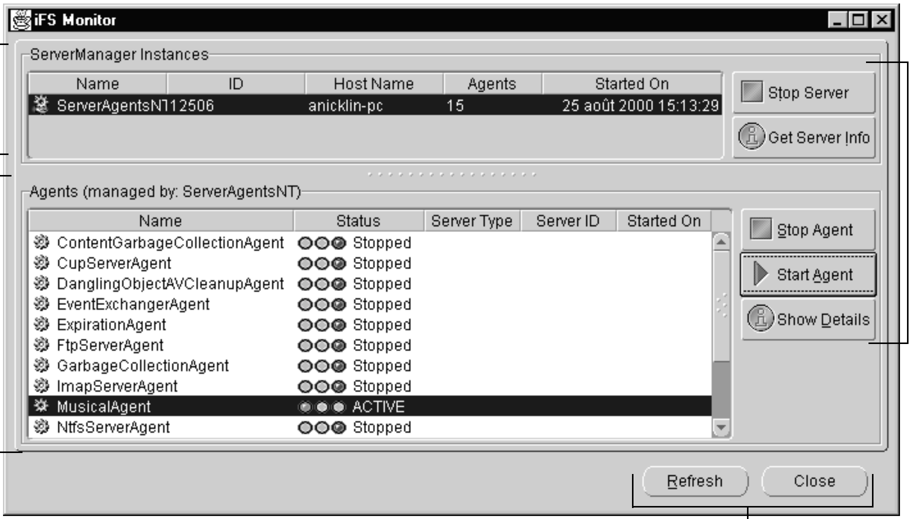
To use Oracle *i*FS Manager to manage the Server Manager instances, you use Oracle *i*FS Monitor. Oracle *i*FS Monitor uses the same Server Manager commands you use on a command line, but through a graphical interface. You still have the option of issuing Server Manager commands on the command line, but now there is an easier way of managing the servers and agents monitored by Server Manager through Oracle *i*FS Manager.

The Oracle *i*FS Monitor is only used to monitor the servers and agents started by Server Manager. You cannot use Oracle *i*FS Monitor to start Server Manager; you must start Server Manager from the machine where Oracle *i*FS was installed. However, you can use Oracle *i*FS Monitor to stop the agents and the Server Manager instance that is monitoring the agents.

Oracle *i*FS Monitor displays the servers registered with the database and the agents managed by Server Manager.

Parts of the Oracle *i*FS Monitor Window

The Oracle *i*FS Monitor window has the following parts:



ServerManager Instances

Name	ID	Host Name	Agents	Started On
ServerAgentsNT	12506	anicklin-pc	15	25 août 2000 15:13:29

Agents (managed by: ServerAgentsNT)

Name	Status	Server Type	Server ID	Started On
ContentGarbageCollectionAgent	○○● Stopped			
CupServerAgent	○○● Stopped			
DanglingObjectAVCleanupAgent	○○● Stopped			
EventExchangerAgent	○○● Stopped			
ExpirationAgent	○○● Stopped			
FtpServerAgent	○○● Stopped			
GarbageCollectionAgent	○○● Stopped			
ImapServerAgent	○○● Stopped			
MusicalAgent	●●● ACTIVE			
NtfsServerAgent	○○● Stopped			

Server Manager Instance(s)

Agents managed by the Server Manager

Task Buttons

Oracle *i*FS Monitor Buttons

Where:

- Server Manager Instance(s)—Displays the Server Manager instances currently running. Each Server Manager instance includes:
  - Name—The name of the Server Manager.
  - ID—The numeric identifier of the Server Manager.
  - Host Name—The name of the machine where the Server Manager is running.
  - Agents—The number of agents currently managed by the particular Server Manager.
  - Started On—The date and time the Server Manager instance was started.
- Agents (managed by: <Server Manager Name>)—Displays the agents managed by the Server Manager selected in the Server Manager Instances box. Each Agent includes:
  - Name—The name of each agent.
  - Status—The status of each agent. When an agent is stopped, a red light displays and the status is Stopped. When an agent is started, a green light displays and the status is changed to ACTIVE. A yellow light indicates that the agents is still in the process of starting or stopping.
  - Server Type—The type of the running server, such as FTP or CUP.
  - Server ID—The numeric identifier for each server running the agent.
  - Started On—The date and time the agent was started.

Note: Information for server type and server ID is only displayed for an external server agent whose server is currently running.

Task Buttons

The buttons on the right side of the Oracle *i*FS Monitor window are used to perform Server Manager tasks. The following table describes the buttons and their purpose.

Button	Purpose
Stop Server	Stops the selected Server Manager instance and all protocols.
Get Server Info	Describes the selected Server Manager instance in a separate window.
Stop Agent	Stops a selected agent.
Start Agent	Starts a selected agent.
Show Details	Shows the details of a selected agent in a separate window.

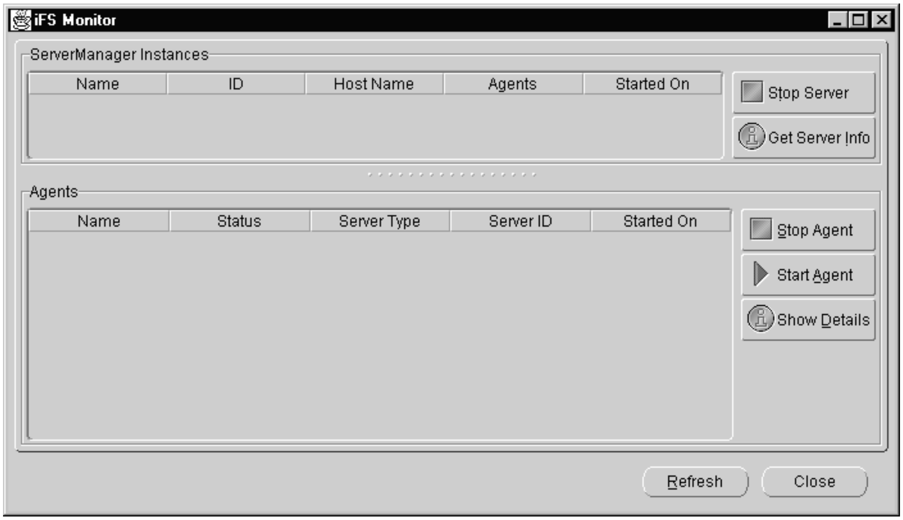
Monitor Buttons

The buttons at the bottom of the Oracle *i*FS Monitor window are used to perform Oracle *i*FS Monitor window tasks. The following table describes the buttons and their purpose.

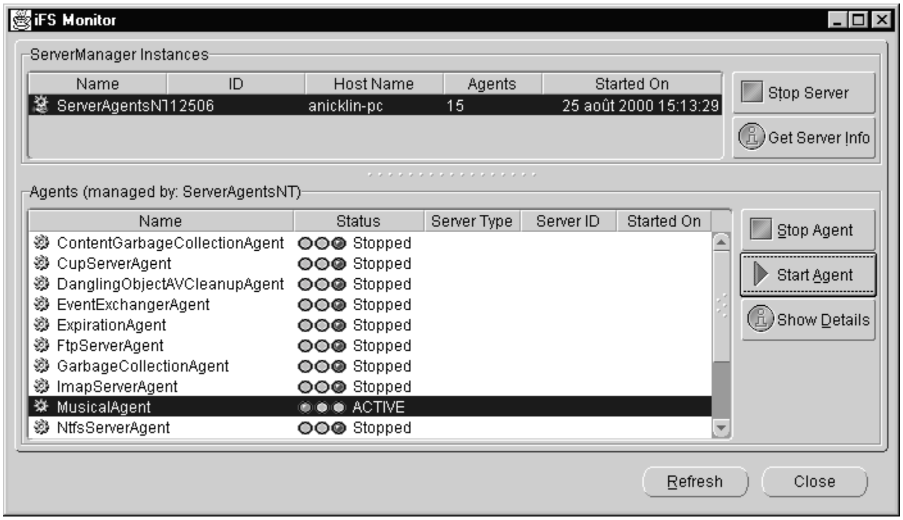
Button	Purpose
Refresh	Refresh the data displayed in the Oracle *i*FS Monitor.
Close	Closes the Oracle *i*FS Monitor window.

Starting Oracle *i*FS Monitor

Oracle *i*FS Monitor is accessed through Oracle *i*FS Manager by selecting Monitor from the File menu. A window displays containing the Server Manager instances and agents. If you are starting Oracle *i*FS Monitor before starting an instance of Server Manager, no information is displayed as shown here:



When a Server Manager instance is started, select the Server Manager for which you want to view agents. The agents display in the Agents box as shown here:



iFS Monitor

ServerManager Instances

Name	ID	Host Name	Agents	Started On
ServerAgentsN112506		anicklin-pc	15	25 août 2000 15:13:29

Agents (managed by: ServerAgentsNT)

Name	Status	Server Type	Server ID	Started On
ContentGarbageCollectionAgent	○○● Stopped			
CupServerAgent	○○● Stopped			
DanglingObjectAVCleanupAgent	○○● Stopped			
EventExchangerAgent	○○● Stopped			
ExpirationAgent	○○● Stopped			
FtpServerAgent	○○● Stopped			
GarbageCollectionAgent	○○● Stopped			
ImapServerAgent	○○● Stopped			
MusicalAgent	●●● ACTIVE			
NtfsServerAgent	○○● Stopped			

All agents are displayed, even if they are have a status of Stopped.

See Also

- Before you can use Oracle *i*FS Monitor, you must start at least one instance of Server Manager. For more information on starting a Server Manager instance, see Chapter 2, "Starting Oracle iFS".
- For a list of Server Manager commands, including the command for starting a Server Manager instance on the machine where Oracle *i*FS was installed, see "Using the Command Line to Manage Server Manager".
- Oracle *i*FS Monitor is accessed through Oracle *i*FS Manager. For more information on starting Oracle *i*FS Manager, see Chapter 2, "Starting Oracle iFS".
- For more information on the agents that you can manage using Oracle *i*FS Monitor, see "Oracle iFS Agents".

Starting An Agent with Oracle *i*FS Monitor

If an agent has been specified in the definition file to start automatically with Server Manager, then the status of that agent displays as ACTIVE and a green light is displayed when the Oracle *i*FS Monitor is started. To start an agent manually, do the following:

1. Make sure at least one instance of Server Manager is started.
2. Click the stopped agent you want to start to select it.
3. Click the Start Agent button.

   This action may take several minutes. A yellow light and a status of Starting indicates the in the process of starting. When the agent is started, the status changes to ACTIVE and a green light is displayed.

Stopping An Agent with Oracle *i*FS Monitor

To stop an agent, do the following:

1. Make sure at least one instance of Server Manager is started and click the instance where the agent you want to stop is managed.
2. Click the agent you want to stop.
3. Click the Stop Agent button.

   This process may take several minutes. A yellow light and a status of Stopping indicates the agent is in the process of starting. When the agent has been stopped, the status changes to Stopped and a red light is displayed.

Stopping a Server Manager Instance with Oracle *i*FS Monitor

To stop a Server Manager instance, do the following:

1. Click the Server Manager instance you want to stop.
2. Click the Quit Server button.

   This process may take several minutes. Each agent managed by the selected Server Manager is stopped and finally the Server Manager instance is stopped.

See Also

You can also perform these same tasks on the machine where Oracle *i*FS is installed through a command line. For more information on these commands, see "Using the Command Line to Manage Server Manager".

Using the Command Line to Manage Server Manager

The Server Manager provides the ability to manage agents and protocol servers with the following commands, invoked at the command line:

- Start—Start running the agent.
- Stop—Stop a running agent or manager. When a Server Manager instance is stopped, all protocol servers and all agents that are managed by that instance are stopped, then exits.
- Suspend—Temporarily stop running an agent without actually stopping it. This is useful if an agent collects cumulative data and stopping it would cause the agent to lose this information.
- Resume—Resume running a suspended agent.
- Restart—To stop and restart the agent, generally used for the external server agent. This is useful for restarting a protocol server to pick up new parameter settings in the definition file.
- Modify—Change the thread priority of an agent.

Note: When a Server Manager instance is stopped, all protocol servers and all agents managed by that instance are stopped. Then this Server Manager instance is exited. When a protocol server that is being managed by a Server Manager instance is stopped, the Server Manager instance still manages the other agents, and does not exit. Therefore, the protocols server can then be easily re-started using the `restart` command. When all servers are stopped using the `stop servers` command in the Server Manager instance, all instances of Server Manager are stopped, except for the instance from which the `stop servers` command was issued.

Monitoring Server Manager Servers and Agents

Server Manager also provides good monitoring functionality. If you issue the `LIST AGENTS` command, the following would be returned:

- Name—The name of the agent.
- Status—The current status of the agent; possible status values are:
  - Active—The agent is running.
  - Resumed—The agent was resumed and in the process of running.
  - Starting—The agent is in the process of starting.
  - Stopped—The agent is not running.
  - Suspended—The agent has been suspended.
- Priority—The thread priority; default value is 5.

The following list describes the sort of information that is provided when issuing the `LIST SERVER` command:

- Server Name—The name of the running server and, optionally, its unique ID.
- Managing Server—The name of the Server Manager that is managing it, if any; if a protocol server was started directly from the operating system, there would be no managing server.
- Hostname—The machine on which the server is running.
- Server Type—The type of the running server, such as FTP or SMB.

When the `DESCRIBE` command is issued, more detailed information is provided on an agent or server, although the information returned depends upon the agent or the server.

Server Manager Commands—The Complete List

The complete list of commands and their syntax are below. The parameters:

- `<agent>`—The name of the agent.
- `<server>`—The name of the server.
- `<manager>`—Either the name of the Server Manager or its ID.
- `<priority>`—An integer value, ranging from 1 to 10, representing the thread priority.

Command	Description
`LIST AGENTS`	Lists all local agents.
`LIST AGENTS -m <manager>`	Lists all agents managed by the specified manager.
`LIST SERVERS [-i]`	Lists all servers; the optional -i flag specifies to include the server's ID.
`DESCRIBE AGENT <agent>`	Describes the specified local agent.
`DESCRIBE AGENT <agent> -m <manager>`	Describes the specified agent managed by the specified manager.
`DESCRIBE SERVER`	Describes the local server
`DESCRIBE SERVER <server>`	Describes the specified server.
`DESCRIBE SERVER <server> -m <manager>`	Describes the specified server managed by the specified manager.
`START AGENT <agent>`	Starts the specified local agent.
`START AGENT <agent> -m <manager>`	Starts the specified agent managed by the specified manager.
`SUSPEND <agent>`	Suspends the specified local agent.
`SUSPEND <agent> -m <manager>`	Suspends the specified agent managed by the specified manager.
`RESUME <agent>`	Resumes the specified local agent.
`RESUME <agent> -m <manager>`	Resumes the specified agent managed by the specified manager.
`MODIFY AGENT <agent> -p <priority>`	Modify the specified local agent with the specified thread priority.
`MODIFY <agent> -p <priority> -m <manager>`	Modify the specified agent managed by the specified manager with the specified thread priority.
`STOP AGENT <agent>`	Stops the specified local agent
`STOP AGENT <agent> -m <manager>`	Stops the specified agent managed by the specified manager.
`STOP AGENTS`	Stops all local agents.
`STOP AGENTS -m <manager>`	Stops agents managed by the specified manager.
`STOP SERVER <server>`	Stops the specified server.

Command	Description
`STOP SERVER <server> -m <manager>`	Stops the servers managed by the specified manager.
`STOP SERVERS`	Stops all servers, except this Server Manager.
`RESTART AGENT <agent>`	Restarts the specified local agent.
`RESTART AGENT <agent> -m <manager>`	Restarts the specified agent managed by the specified manager.
`HELP`	Displays summary help information.
`HELP KEYWORDS`	Displays a list of command keywords.
`HELP <command>`	Displays help on the command specified.
`EXIT`	Shuts down this server manager.

Oracle *i*FS Agents

Oracle *i*FS provides agents that perform required tasks to support the Oracle *i*FS system. Agents run in the same process as the Server Manager that started it. Custom agents can also be written to perform specific tasks. There are four different types of agents:

- Time-based Agents—These agents are triggered by time intervals. The time interval can be configured.
- Event-based Agents—These agents are triggered by events. The agent registers for specific Oracle *i*FS events and is triggered when they occur.
- Time-based and Event-based Agents—These agents are triggered by time events, Oracle *i*FS events, or both.
- Agents to Manage Protocol Servers (External Server Agents)—A special agent to start up protocol servers and to respond to unexpected protocol server failures.

Server Manager can be configured to automatically start up any agent. Any agent that is not started automatically can be started manually. Each Server Manager should have its own definition file. The definition file contains information specific to the Server Manager and the list of agents to load and manage. Once Server Manager has started and loaded the agents, any change to the definition file, including modifying the list of agents, will not be picked up until the Server Manager is shut down and restarted.

The following information is found in the definition file:

- Username / Password (optional)—The Oracle *i*FS user information to log in to the system. The user must be an Oracle *i*FS administrator. Rather than storing the password in a file, you can enter the username and password interactively by setting `InteractiveCredentials` to true.
- Output File—The location of the log file.
- Manager Name—The name of the Server Manager. If you are running multiple Server Managers, each name must be unique.
- Interactive—Specifies if the Server Manager should run in the background (false) or interactively (true). Running interactively allows the administrator to enter in commands for managing the agents and servers.
- Interactive Credentials—If this option is set to true, a prompt for the username, password, service name, and schema password is displayed when Server Manager is run.
- List of Agents—The agents to be loaded and managed by this Server Manager.
- Agent Specific Information—Information required for that agent to run.

The following agents are provided with Oracle *i*FS:

- Expiration Agent
- Garbage Collection Agent
- Content Garbage Collection Agent
- Dangling Object AV Cleanup Agent
- Event Exchanger Agent
- External Server Agent
- Service Watchdog Agent
- Quota Agent
- Outbox Agent

Expiration Agent

All public objects have an attribute called ExpirationDate. Once this date passes, the public objects are automatically deleted. This is handled by the Expiration agent, which periodically deletes expired objects. If the expiration date of a public object passes, the agent deletes the public object. This agent runs at specific times of the day as specified in the definition (.def) file by the parameter `activationtimes`.

The following information must be included in the definition file to run the Expiration agent:

Parameter	Default Entry	Description	Comments
`Name`	`ExpirationAgent`	The name of the agent.	Can run only one in system; do not modify.
`Class`	`oracle.ifs.agents.server.ExpirationAgent`	The agent class path.	Do not modify.
`Start`	`true`	Specifies if the agent automatically starts up when the Server Manager starts up.	true=automatic start, false=manual
`InitialTimeOfDay`	`02:00:00`	First time event.	Set time based on a 24 hour clock.
`ActivationPeriod`	`24h`	Defines the time interval between when the agent will be triggered.	h = hours, m = minutes, s = seconds
`TwoPhaseDeleteSupport`	false	For future implementation.	Do not modify.

Garbage Collection Agent

The Garbage Collection agent runs at specific times of the day as specified in the definition (.def) file. Currently, this agent fixes invalid public object owners, creators, deletors, and modifiers. It uses the following four parameters for this purpose, which are specified in the definition file:

- ReplacementOwner
- ReplacementCreator
- ReplacementModifier

- ReplacementDeletor

For example, a document is created and modified by "ssmith." The creator, owner, and last modifier attribute of document are set to ssmith's object id. If "ssmith" is deleted, the attribute value becomes invalid. The agent replaces these invalid attribute values with the ID of the replacement owner, creator, or deletor specified in the definition file.

The following information must be included in the definition file to run the Garbage Collection agent:

Parameter	Default Entry	Description	Comments
`Name`	`GarbageCollectionAgent`	The name of the agent.	Can run only one in system; do not modify.
`Class`	`oracle.ifs.agents.server.GarbageCollectionAgent`	The agent class path.	Do not modify.
`Start`	`true`	Specifies if the agent automatically starts up when the Server Manager starts up.	true=automatic start, false=manual
`InitialDelay`	`30m`	The initial delay when the agent will first run; this allows the agents' first trigger to be staggered.	h = hours, m = minutes, s = seconds
`ActivationPeriod`	`1h`	The time interval between the agent's subsequent runs.	h = hours, m = minutes, s = seconds
`ReplacementOwner`	`system`	User to be replaced as owner.	Modify as needed.
`ReplacementDeletor`	`system`	Users to be replaced as deletor.	Modify as needed.
`ReplacementCreator`	`system`	User to be replaced as creator.	Modify as needed.
`ReplacementModifier`	`system`	User to be replaced as modifier.	Modify as needed.

Content Garbage Collection Agent

Content is stored separately from file attributes. The Content Garbage Collection agent identifies unreferenced content and deletes it. For performance reasons, the content of a document is not deleted when the document is deleted. The Content Garbage Collection agent deletes the unreferenced content. This agent runs at specific times of the day as specified in the definition (.def) file.

The following information must be included in the definition file to run the Content Garbage Collection agent:

Parameter	Default Entry	Description	Comments
`Name`	`ContentGarbageCollection Agent`	The name of the agent.	Can run only one in system; do not modify.
`Class`	`oracle.ifs.agents.server. ContentGarbageCollection Agent`	The agent class path.	Do not modify.
`Start`	`true`	Specifies if the agent automatically starts up when the Server Manager starts up.	true=automatic start, false=manual
`InitialTimeOfDay`	`30m`	The initial delay when the agent will first run; this allows the agents' first trigger to be staggered.	h = hours, m = minutes, s = seconds
`ActivationPeriod`	`1h`	The time interval between the agent's subsequent runs.	h = hours, m = minutes, s = seconds

Dangling Object AV Cleanup Agent

This agent is similar to the Garbage Collection agent. This agent cleans up dangling object type references and identifies all invalid object references, such as references to objects that no longer exist, and sets these references to null for array type attributes and zero for scalar attributes. For example, this agent cleans up the owner attribute of a document pointing to directory object which was deleted and is now invalid.

The following information must be included in the definition file to run the Dangling Object AV Cleanup agent:

Parameter	Default Entry	Description	Comments
`Name`	`DanglingObjectAV CleanupAgent`	The name of the agent.	Can run only one in system; do not modify.
`Class`	`oracle.ifs.agents.server. DanglingObjectAVCleanup Agent`	The agent class path.	Do not modify.
`Start`	`true`	Specifies if the agent automatically starts up when the Server Manager starts up.	true=automatic start, false=manual
`InitialTimeOfDay`	`00:45:00`	The first time event.	Set time based on a 24 hour clock.
`ActivationPeriod`	`24h`	Time interval to when the agent runs again.	h=hours, m=minutes, s=seconds

`ExcludedAttributeList`	`AUDITENTRY`
	`PUBLICOBJECT:OWNER`
	`PUBLICOBJECT:DELETOR`
	`PUBLICOBJECT:LASTMODIFIER`
	`VERSIONSERIES:RESERVOR`
		A list of attributes for which invalid references to Library Objects are not cleaned up.	Modify as needed.

Event Exchanger Agent

This agent periodically purges expired events from the event queue. The following information must be included in the definition file to run the Event Exchanger agent:

Parameter	Default Entry	Description	Comments
`Name`	`EventExchangerAgent`	The name of the agent.	Can run only one in system; do not modify.
`Class`	`oracle.ifs.agents.server.Event ExchangerAgent`	The agent class path.	Do not modify.

Parameter	Default Entry	Description	Comments
`Start`	`true`	Specifies if the agent automatically starts up when the Server Manager starts up.	true=automatic start, false=manual
`InitialDelay`	`30m`	The initial delay when the agent will first run; this allows the agents' first trigger to be staggered.	h = hours, m = minutes, s = seconds
`ActivationPeriod`	`20s`	The time interval between the agent's subsequent runs.	h = hours, m = minutes, s = seconds
`EventLifespan`	`1800`	The number of seconds after which an event is assumed to have been delivered and become eligible for purging.	Unit of measure is seconds.

External Server Agent

This is a special kind of agent used to start protocol servers. The External Server agent reads the command for starting up a server, and executes that command in a separate process. It also re-starts the protocol servers if they go down unexpectedly.

The default entries in the definition file are specific to the protocol server this agent is managing. This example is the External Server agent for the FTP server:

Parameter	Default Entry	Description	Comments
`Name`	`FtpServerAgent`	The name of the agent.	Modify to be unique.
`ServerName`	`FtpServer`	The name the Server Manager uses to identify the protocol server.	Modify to be unique.
`Class`	`oracle.ifs.agents.server.ExternalServerAgent`	The agent class path.	Do not modify.

Parameter	Default Entry	Description	Comments
`Start`	`true`	Specifies if the agent automatically starts up when the Server Manager starts up.	true = automatic start, false = manual
`Activation Period`	`20s`	Period of time the agent waits for a state transition.	Do not modify.
`ReStartOnError`	`true`	Specifies if an the agent should restart the server when an unexpected server failure is detected.	true = automatic start up, false = manual
`PassCredentials`	`true`	Passes the Oracle *i*FS credentials to the protocol server.	true = pass credentials, false = do not pass
`ConfigurationFilename`	`FtpServer-Sun.def`	The definition file for the protocol server.	Modify as needed.
`LogFilename`	`FtpServer.log`	The log file name.	Modify to be unique.
`Command`	`java -DFtpServer -mx64M oracle.ifs.protocols.ftp.server.FtpServer`	The command to start the server.	Modify as needed; in particular, the heap size depending upon usage and HW resources available.
`JavaProcess`	`false`	If true, a java class is running. Uses other parameters (JavaProcessClass, JavaVmArgs, JavaProcessArgs) to construct the command to be run and ignored COMMAND parameter.	Use this if a java class is running.
`JavaProcessClass`	`none`	If JavaProcess parameter is set to true, provide the class that needs to be run here.	Not used if JavaProcess=false.

Parameter	Default Entry	Description	Comments
`JavaVmArgs`	`none`	If a java class is running, you can provide arguments to java VM here. (not the arguments for the class itself). Not used if JavaProcess=false.	VM arguments like -mx, -ms and others need to be provided here.
`JavaProcessArgs`	`none`	If a java class is running, these are the arguments to the java class itself (not the arguments to VM).	Not used if JavaProcess=false.
`PropagateClassPath`	`true`	If a java class is running (by setting JavaProcess to true), then setting this to true makes the CLASSPATH of ESA to be passed down to the environment in which the Java class runs. Not used if JavaProcess=false.	Use this if you are running a java class.
`PassIfsArgs`	`true`	If this is set (or not specified), the External Server Agent passes internal Oracle *i*FS arguments to the command or java class being run.	Do not modify this parameter.

Service Watchdog Agent

The Service Watchdog agent works together with the External Server agents. This agent cleans up after server failures. Immediately after a server failure, the failed server will show up in the list of servers. After a few minutes of server failure, the Service Watchdog agent detects that the server is no longer running and removes it from the list of servers. The following information must be included in the definition file to run the Service Watchdog agent:

Parameter	Default Entry	Description	Comments
`Name`	`ServiceWatchdogAgent`	The name of the agent.	Can run only one in system; do not modify.
`Class`	`oracle.ifs.agents.server.ServiceWatchdogAgent`	The agent class path.	Do not modify.
`Start`	`true`	Specifies if the agent automatically starts up when the Server Manager starts up.	true=automatic start, false=manual
`ActivationPeriod`	`20s`	The time interval to when the agent will run again.	h = hours, m = minutes, s = seconds
`DeadServiceCheckFrequency`	`6`	The frequency of activation periods in which a dead service will be checked. The default is to check every 6 x the ActivationPeriod, which defaults to 20 seconds.	
`UnlockedServerCheckFrequency`	`1`	The frequency of activation periods in which an unlocked server will be checked. The default is to check for dead service every 1 x the ActivationPeriod.	An unlocked server is one that is in the process of shutting down the server.
`ServiceTimeoutPeriod`	`120`	The time before the Server Manager will kill an unresponsive protocol server.	Unit of measure is seconds.

Parameter	Default Entry	Description	Comments
`UnlockedServerTimeout Period`	`10`	The time before Server Manager will kill an unlocked protocol server.	Unit of measure is seconds.

Quota Agent

The Quota agent is triggered by an event to check for any users who may have used up their quota. This agent also periodically checks all users who are active, according to a specified timer period. The agent updates the user's consumed storage. When the consumed storage is over the allocated quota, then the user will not be able to add any more documents to Oracle *i*FS.

A user's quota is calculated upon the content already consumed. Therefore, a user will go over quota by adding the final file that pushes the consumed storage over the allocated storage. When setting users' allocated quota, remember that a users' last file will put them over quota before being denied.

Quotas will not be enforced if:

- The Quota agent has not been started or running.
- The user's quota has not been enabled.

The following information must be included in the definition file to run the Quota agent:

Parameter	Default Entry	Description	Comments
`Name`	`QuotaAgent`	The name of the agent.	Can run only one in system; do not modify.
`Class`	`oracle.ifs.agents.server.QuotaAgent`	The agent class path.	Do not modify.
`Start`	`true`	Specifies if the agent automatically starts up when the Server Manager starts up.	true=automatic start, false=manual
`ActivationPeriod`	`15m`	The time interval to when the agent will run again.	h=hours, m=minutes, s=seconds

Outbox Agent

The Outbox Agent provides a mechanism to deliver Oracle *i*FS e-mail created programmatically. Oracle *i*FS applications creating e-mails will place the messages in the system outbox (`/ifs/outbox`). The Outbox agent is notified every time an event is received indicating the insertion of an e-mail into the outbox folder. The agent sends the e-mail to the mail transfer agent (Sendmail) via SMTP.

Note: This agent is only required for custom e-mail applications.

The following information must be included in the definition file to run the Outbox agent:

Parameter	Default Entry	Description	Comments
`Name`	`OutboxAgent`	The name of the agent.	Modify to be unique.
`Class`	`oracle.ifs.protocols.email.outbox.OutboxAgent`	The agent class path.	Do not modify.
`Start`	`false`	Specifies if the agent automatically starts up when the Server Manager starts up.	true=automatic start, false=manual
`ActivationPeriod`	`15m`	The time interval to when the agent will run again.	h=hours, m=minutes, s=seconds

Definition Files for Protocol Servers

After setting up the Server Manager's definition file, you may want to modify the definition files for each of the protocol servers. Each definition file contains information specific to the protocol server. If you will be running multiple copies of the same protocol server, it is strongly recommended that you have a uniquely named definition file for each instance of that protocol.

The following definitions files are discussed:

- Windows Client Protocol (WCP) Server
- Windows NT File System Protocol Server (for Windows NT)
- Simple Mail Transfer Protocol (SMTP) Server

- Internet Mail Access Protocol (IMAP) Server
- Server Message Block (SMB) Protocol Server
- File Transfer Protocol (FTP) Server
- Command Line Utilities Protocol (CUP) Server

Windows Client Protocol (WCP) Server

The Windows Client Protocol (WCP) is the main protocol used by the Oracle *i*FS Utilities on Windows 95, Windows 98, Windows NT 4.0, and Windows 2000. Only one WCP server is required for each Oracle *i*FS schema. If you start more than one instance of the WCP server, only the last instance started will take into effect. Make sure the correct instance of the WCP server is available.

The WCP port assignment is different from other protocols used by Oracle *i*FS because it does not run on any designated port. Each time it is started, the WCP finds an open port on the machine on which it is running. Exactly one WCP server should run in a system.

The following parameters must be included in the definition file to run the WCP server. To include these parameters, edit the `WcpServer.def` file located in the `$ORACLE_HOME/ifs<version>/settings` directory.

Parameter	Default Entry	Description	Comments
`server`	WcpServer	The name of the server properties file.	The properties file defines settings for the Oracle *i*FS repository.
`AppendLog`	true	Specifies if the log should be appended to or overwritten each time the server is started.	true = append, false = overwrite
`LogCommands`	true	Logs every command.	true = log every command, false = do not log every command
`ShowConnectLocations`	true	Include the location where a connection originates.	true = include location, false = do not include location
`WcpServerSocket Timeout`	0	Amount of time between activities before the WCP server socket times out. Default is 0; no timeout.	Unit of measure is milliseconds.

Parameter	Default Entry	Description	Comments
`AuthenticatedPoolSize`	2	The number of concurrent library sessions servings per mounted Oracle *i*FS drive. Specify the optional maximum number of concurrent Windows utilities you want to run on each mounted drive.	
`AcceptQueueSize`	50	The number of server requests back-logged before denying requests.	Do not modify.
`WcpClientSize`	10	Used to track the number of pre-allocated WCP sessions.	Increase to a larger number for better performance if a larger number of clients connect simultaneously.
`OptimizeStartup`	true	Specifies if the server should "warm up" for better performance.	true = warm up server, initial cost starting up server, need to indicate objects (warm up settings to load, false = do not warm up server
`WarmupAcls`	`true`	List of the objects to be loaded during start up for better user performance.	
`WarmupFormats`	`true`		
`WarmupMedias`	`true`		
`WarmupUsers`	`true`		

Windows NT File System Protocol Server (for Windows NT)

The NT File System Protocol Server allows the Oracle *i*FS repository to appear as a local hard drive on the server machine. When using the NT File System Protocol Server, the Oracle *i*FS repository will be mapped to a drive letter on the server machine and can be accessed using the file system as if it were a local hard drive.

The following parameters must be included in the definition file to run the NTFS server. To include these parameters, edit the `NtfsServer.def` file located in the `$ORACLE_HOME/ifs<version>/settings` directory.

Parameter	Default Entry	Description	Comments
`server`	`NtfsServer`	The name of the server properties file.	The properties file defines settings for the Oracle *i*FS settings.

Parameter	Default Entry	Description	Comments
`AppendLog`	`true`	Specifies if the log should be appended to or overwritten.	true = append, false = overwrite
`DriveLetter`	`o`	The drive letter used to map the Oracle *i*FS drive.	An available and valid drive letter (a-z).
`LogLevel`	`1`	Specifies the type of information listed in the log file.	A number from 1 to 4. 1 is the minimum and 4 is the maximum information in the log file.
`LogStackTrace`	`false`	Specifies if the Java stack trace is listed in the log file.	true = list Java stack trace, false = do not list Java stack trace

Simple Mail Transfer Protocol (SMTP) Server

This is a base implementation of the Simple Mail Transfer Protocol. Oracle *i*FS uses Sendmail as the mail transfer agent. The SMTP server is used as the mail delivery agent for Oracle *i*FS. Sendmail connects to the SMTP server via the SMTP protocol and uses it to check if inboxes for e-mail recipients exist in Oracle *i*FS and to deliver mail to those inboxes.

The following parameters must be included in the definition file to run the SMTP server. To include these parameters, edit the `SmtpServer.def` file located in the `$ORACLE_HOME/ifs<version>/settings` directory.

Parameter	Default Entry	Description	Comments
`server`	`SmtpServer`	The name of the server properties file.	The properties file defines settings for the Oracle *i*FS repository.
`LogCommands`	`true`	Logs every command.	true = log every command, false = do not log every command
`AppendLog`	`true`	Specifies if the log file should be appended to or overwritten.	true = append, false = overwrite
`port`	`2500`	The port on which the server is running.	This port is also in the `Sendmail.cf` file. If changed, it must be changed in both places.
`SmtpSessionCount`	`5`	The number of SMTP sessions created at startup time.	

Parameter	Default Entry	Description	Comments
`OptimizeStartup`	`true`	Specifies if the server should "warm up" for better performance.	true = warm up server; initial cost starting up server; need to indicate objects (warm up settings), false = do not warm up server
`WarmupAcls`	`true`	List of the objects to be loaded during start up for better user performance.	
`WarmupFormats`	`true`		
`WarmupMedias`	`true`		
`WarmupUsers`	`true`		

Internet Mail Access Protocol (IMAP) Server

This is an implementation of Internet Mail Access Protocol. This provides a mechanism for mail clients to access mail from Oracle *i*FS accounts.

The following parameters must be included in the definition file to run the IMAP server. To include these parameters, edit the `ImapServer.def` file located in the `$ORACLE_HOME/ifs<version>/settings` directory.

Parameter	Default Entry	Description	Comments
`server`	`ImapServer`	The name of the server properties file.	The properties file defines settings for the Oracle *i*FS repository.
`LogCommands`	`false`	Logs every command.	true = log every command, false = do not log every command
`AppendLog`	`true`	Specifies if the log file should be appended to or overwritten.	true = append, false = overwrite
`ImapTimeout`	`600000`	The amount of time before the server waits before timing out; default is 600 seconds or 10 minutes.	Unit of measure is milliseconds.
`ImapPortNum`	`143`	The port on which the server is running.	
`MaxLiteral`	`512`	Required by the IMAP command.	Do not modify.

Parameter	Default Entry	Description	Comments
`OptimizeStartup`	`true`	Specifies if the server should "warm up" for better performance.	true = warm up server; initial cost starting up server; need to indicate objects (warm up settings), false = do not warm up server
`WarmupAcls`	`true`	List of the objects to be loaded during start up for better user performance.	
`WarmupFormats`	`true`		
`WarmupMedias`	`true`		
`WarmupUsers`	`true`		

Server Message Block (SMB) Protocol Server

The SMB server allows Windows users to easily copy files between their local drives and the Oracle *i*FS repository. It also allows users to browse and edit files and even execute programs directly from the Oracle *i*FS repository. SMB is best suited for interactive browsing and modifying documents.

The following parameters must be included in the definition file to run the SMB server. To include these parameters, edit the `SmbServer.def` file located in the `$ORACLE_HOME/ifs<version>/settings` directory.

Parameter	Default Entry	Description	Comments
`server`	`SmbServer`	The name of the server properties file.	The properties file defines settings for the Oracle *i*FS repository.
`TraceLevel`	`Err`	Specifies the level of output for the log file.	Off = all, ERR = trace errors and significant events only, Low = minimal tracing, High = verbose tracing
`TraceFile`	none	The trace file name.	The default directory is `$ORACLE_ HOME/ifs<version>/log/`
`VerboseIfsExceptions`	`true`	Turn on verbose mode for Oracle *i*FS exceptions.	true = verbose mode, false = succinct mode
`ShowTimeLevel`	`off`	Add millisecond timestamp for this level of tracing.	Err = trace errors and significant events only, off = all

Parameter	Default Entry	Description	Comments
ShareMyHome	true	Specifies if the virtual MyHome is shared as a mount point.	true = MyHome the list of mount points, false = only administer defined mount points
Workgroup	WORKGROUP	Windows workgroup.	This determines the workgroup where the SMB server will appear in the Network Neighborhood.
HostName	machine name	The name of the server when browsing the Network Neighborhood when mounting an Oracle *i*FS drive.	Default is the name of the machine the SMB server is running on.
SmbPort	139	Port for the SMB server to listen on.	Need to run as root since it is the default port. All Windows clients require that this port number is not changed.
BufferLog	true	Specifies if the log file should be buffered.	true = buffer, false = do not buffer
AppendLog	true	Specifies if the log file is appended or overwritten.	true = append, false = overwrite
AllowGuestIfNot User	true	Allows users who do not have valid accounts on this server to browse as a guest.	true = allows non-guest users as guest, false = forces a valid username
MaxOpenFiles	30	The maximum number of files a user can have open on a single mountpoint.	
MaxAllOpenFiles	100	The total number of open files the SMB server allows across all users.	The total is also limited by the number of connections configured for the SMB server and database.
EnableParsing	true	Enables the parsing feature when saving Oracle *i*FS files.	true = enables parsing, false = disables parsing
EnableRendering	true	Enables the rendering feature when reading Oracle *i*FS files.	true = enables rendering, false = disables rendering
SessionTimeout	43200000 (12 hours)	The server will disconnect idle sessions after this timeout.	Unit of measure is in milliseconds.
NbTraceLevel	ERR	Specifies the level of output for the log file.	Off = all, Err = trace errors and significant events only, Low = minimal tracing, High = verbose tracing

Parameter	Default Entry	Description	Comments
`NbTraceFile`	`none`	The file name for the trace file.	The default directory is `$ORACLE_HOME/ifs<version>/log`.
`IpAddr`	`server IP address`	The IP address that is advertised as the location for the SMB server.	This value is defaulted from the server IP address and should not be modified except in unusual cases, for example, multi-homed server.
`Netmask`	`255.255.252.0`	The mask used to determine the address for broadcast packets.	This limits or broadens the scope of broadcast messages. Most networks should use the default.
`Broadcast`	none	Overrides the default broadcast address from the IP and Netmask packets.	Change only if you need to explicitly specify a broadcast address.
`HostAnnouncer`	`true`	Enables the SMB server to be listed in the Network Neighborhood.	true = server will be listed, false = server will not be listed.

File Transfer Protocol (FTP) Server

The FTP server allows users to easily transfer files between one file system and the Oracle *i*FS repository. FTP is particularly useful when performing bulk transfers; in regards to both ease and performance. In addition to the common FTP commands, the Oracle *i*FS server also has built-in parsing and versioning support.

The following parameters must be included in the definition file to run the FTP server. To include these parameters, edit the `FtpServer.def` file located in the `$ORACLE_HOME/ifs<version>/settings` directory.

Parameter	Default Entry	Description	Comments
`server`	`FtpServer`	The name of the server properties file.	The properties file defines settings for the Oracle *i*FS repository.
`LogCommands`	`false`	Logs every command.	true = log every command, false = do not log every command
`AppendLog`	`true`	Specifies if the log file should be appended to or overwritten.	true = append, false = overwrite

Parameter	Default Entry	Description	Comments
VerboseLogging	false	Logs detailed information, including exception stacks.	true = log detailed information, false = log summary information
ShowConnect Locations	true	Logs the location where a connection originates.	true = include location, false = do not include location
DateFormat	MMM dd HH:mm	Specifies the default date format.	
Timeout	900000	Amount of time between activity before the connection times out; default is 900 seconds or 15 minutes.	Unit of measure is milliseconds.
Delimiter	/	Specifies the Folder Path Delimiter.	
port	21 (Windows NT), 2100 (Unix)	The port on which the server is running.	
AcceptQueueSize	50	The number of server requests back-logged before denying requests.	Do not modify.
OptimizeStartup	true	Specifies if the server should "warm up" for better performance.	true = warm up server; initial cost starting up server; need to indicate objects (warm up settings), false = do not warm up server
WarmupAcls	true	List of the objects to be loaded during start up for better user performance.	
WarmupFormats	true		
WarmupMedias	true		
WarmupUsers	true		
AnonymousPool Size	2	Initial pool size for anonymous connections.	
WebuiTimeout	60000	Time the FTP session is valid when called from the Web interface for drag and drop upload. Default is 60 seconds.	Unit of measure is milliseconds.

Command Line Utilities Protocol (CUP) Server

The CUP server responds to the requests from the Oracle *i*FS command line utilities, a set of low-level development tools. Since the command line utilities allow direct access to the objects and their attributes, the user must be familiar with the Oracle *i*FS class hierarchy to manipulate the objects. The utilities target users with deeper knowledge and understanding of the Oracle *i*FS objects and their relationships with one another. It is not meant to be an end-user tool.

Note: Logging into Oracle *i*FS through the Command Line Utilities allows any other users on the same machine to access Oracle *i*FS through that login without logging on themselves. To avoid this problem, use the `ifsshell` command on machines shared by multiple users.

The following parameters must be included in the definition file to run the CUP server. To include these parameters, edit the `CupServer.def` file located in the `$ORACLE_HOME/ifs<version>/settings` directory.

Parameter	Default Entry	Description	Comments
`server`	`CupServer`	The name of the server properties file.	The properties file defines settings for the Oracle *i*FS repository.
`LogCommands`	`false`	Logs every command.	true = log every command, false = do not log every command
`AppendLog`	`true`	Specifies if the log file should be appended to or overwritten.	true = append, false = overwrite
`ShowConnect Locations`	`true`	Logs the location where a connection originates.	true = include location, false = do not include location
`DateFormat`	`MMM dd HH:mm`	Specifies the default date format.	
`AnonymousPool Size`	`2`	Initial pool size for anonymous connections.	
`Timeout`	`900000`	Amount of time between activity before socket times out; default is 900 seconds or 15 minutes.	Unit of measure is milliseconds.

Parameter	Default Entry	Description	Comments
SessionStateTimeoutPeriod	900000	Amount of time between activity before the session times out; default is 900 seconds or 15 minutes.	Unit of measure is milliseconds.
Delimiter	/	Specifies the Folder Path Delimiter.	
port	4180	The port on which the server is running.	
AcceptQueueSize	50	The number of server requests back-logged before denying requests.	Do not modify.
OptimizeStartup	true	Specifies if the server should "warm up" for better performance.	true = the initial cost of starting up the server; need to inidicateobjects (warm up settings) to load, false = do not warm up server
WarmupAcls	true	List of the objects to be loaded during start up for better user performance.	
WarmupFormats	true		
WarmupMedias	true		
WarmupUsers	true		

9

Configuration and Back Up

A major feature of Oracle *i*FS is its support of multiple protocols, such as FTP, SMB, and IMAP. This chapter provides information on setting up, configuring, and administering these protocols and e-mail. Topics include:

- Configuration and Port Issues
- Configuring E-mail
- Integrating Oracle iFS Mail and Sendmail Pro 8.9.3
- Using the Windows NT File System Protocol Server
- Backup and Recovery of Data
- Configuring Oracle Apache Web Server
- Configuring Apache As Your Web Server on Windows NT

Configuration and Port Issues

Each protocol used by Oracle *i*FS listens on a port. These ports are assigned numbers by default although they can be reconfigured. Some protocols also need to be configured before they can be used. If you configure a protocol, remember to start and stop that protocol's server.

Protocols and their Ports

The following table displays the Oracle *i*FS protocols and the default ports they run on, whether or not the parts can be configured, and what definition file to use to configure them. All definition files are located in the following directory:

```
$ORACLE_HOME/ifs<version>/settings
```

Protocol	Default Port	Port Configurable?	Definition File	Variable to Configure
File Transfer Protocol (FTP)	21	Yes	FtpServer.def	Change in the definition file.
Command Line Utility Protocol (CUP)[1]	4180	Yes	CupServer.def	Change the `IFS_CUP_PORT` variable and the port number.
Simple Mail Transfer Protocol (SMTP)[2]	2500	Yes	SmtpServer.def	Change the port number in SmtpServer.def, Sendmail.cf and ifs.m4
Internet Messaging Access Protocol (IMAP)	143	Yes	ImapServer.def	Change in the definition file.
HyperText Transfer Protocol (HTTP)	80	Yes	Not Applicable	Use the JWS administration utility and change using port 1717.
Windows Client Protocol (WCP)[3]	Generated	Not Applicable	WcpServer.def	Not applicable.
Server Message Block (SMB)	139	No	SmbServer.def	Do not configure SMB.

[1] This is a proprietary protocol used for Oracle *i*FS operations.

[2] This is a simple implementation to support verification and delivery.

[3] This is a proprietary protocol used for Oracle *i*FS operations.

Note: If you decide to configure these protocols manually instead of using the Configuration Utility provided during installation, the steps may generate multiple errors.

Configuring E-mail

When you create users, among the folders automatically created is a mail folder within which there is an inbox folder. The user's mail messages are delivered into the inbox folder. You can use either Outlook Express or Netscape 4.7 Messenger as your e-mail clients. You must set up e-mail accounts appropriately.

E-mail Setup for UNIX

Oracle *i*FS uses an open-source implementation of the Sendmail 8.9.3 product as the mail transfer agent. Oracle *i*FS has an SMTP server to support verification and delivery. In addition, Oracle *i*FS also provides an IMAP server.

Basic e-mail configuration is completed as part of installation and configuration of Oracle *i*FS. When e-mail is configured during installation, the following is done:

- A new sendmail executable is placed in `/usr/lib`.
- A `sendmail.cf` file is generated and placed in `/etc`.

Sendmail Configuration for UNIX

The `sendmail.cf` file generated during installation is created from a template. This template is used by the Oracle *i*FS configuration program only; Oracle *i*FS also provides .m4 and .mc files to support your own configuration of Sendmail. Configuring the `sendmail.cf` file is not recommended unless completed by qualified Sendmail specialists; it will not be supported by Oracle.

> **Note:** Oracle will not provide support in configuring Sendmail. The `sendmail.cf` file provided with Oracle *i*FS allows you to use Oracle *i*FS e-mail. Any further extensions and enhancements to the `sendmail.cf` file is the responsibility of the user. In accordance with open source licensing requirements, a full copy of the sendmail source is also provided with Oracle *i*FS. To view the full sourcecode for Sendmail, Version 8.9.3, use the `sendmail.8.9.3.tar.gz` in the `$ORACLE_HOME/ifs<version>/admin/email/<platforms>/opensrc` directory.

The .mc and .m4 files that can be used for configuration are found in the following location:

- `$ORACLE_HOME/ifs<version>/admin/email/<OS>/ifs-<OSNAME>.mc`

  where <OS> and <OSNAME> can be the following:

Platform	<OS>	<OSNAME>
AIX	`aix`	`aix4`
HP-UX	`hp-ux`	`hpux10`
Linux	`linux`	`linux`
Tru64	`osf1`	`osf1`
Solaris	`solaris`	`solaris2`

- `$ORACLE_HOME/ifs<version>/admin/email/common/cf`—For the base of the .m4 directory tree.

For the default configuration:

- Added files—`$ORACLE_HOME/ifs<version>/admin/email/cf/mailer/ifs.m4`

Note for Solaris Users: For configuration, use the GNU M4. The Solaris version of the M4 macro-processor is not compatible with the open source Sendmail m4 files. For reference, go to www.gnu.org or see the *Help for UNIX System Administrators, Sendmail* guide (ISBN: 1-56592-222-0, publisher: O'Reilly). GNU M4 is not shipped with Oracle *i*FS 1.0.

Integrating Oracle *i*FS Mail and Sendmail Pro 8.9.3

The Oracle *i*FS e-mail system provides an IMAP server and an SMTP protocol server that provides verification and local delivery support. Oracle *i*FS e-mail does not have its own mail transfer agent, but relies on sendmail to act as the mail transfer agent. This section pertains to the Sendmail Pro product from Sendmail, Inc., and the following steps guide you through set up and configuration of Sendmail Pro version 8.9.3.

Installing Sendmail Pro 8.9.3

To install Sendmail Pro 8.9.3, refer to the Sendmail Pro 8.9.3 documentation. The installation is completed using the standard UNIX pkgadd utility. For further installation instructions, refer to the INSTALL document located for your platform of the Sendmail Pro distribution.

Oracle *i*FS Installation Dependencies

When the Oracle *i*FS Configuration Assistant is run during installation of Oracle *i*FS, the required pre-configuration (.m4 and .mc) files are prepared for e-mail configuration.

Configuring Sendmail

The configuration process involves running the `ifsemailsetup_pro893` script located in the `$ORACLE_HOME/ifs<version>/bin` directory. This script sets up .m4 and .mc files and generates a sendmail configuration file that is used to integrate Oracle *i*FS and Sendmail Pro 8.9.3. You must run this script as the `root` user.

Quick Setup Steps

Steps 1 and 2 can be executed in any order.

1. Configure Oracle *i*FS using the Configuration Assistant. This step can be completed independently of the Sendmail install.
2. Install Sendmail Pro 8.9.3.
3. Run the script `ifsemailsetup_pro893` for configuring Oracle *i*FS e-mail.

> **Note:** If you want to migrate to use open source Sendmail instead of Sendmail Pro 8.9.3, you must first de-install Sendmail Pro 8.9.3.

Sendmail Pro 8.9.3 Graphical Administration Tool

Sendmail Pro 8.9.3 includes a graphical administration and configuration tool. This is a Web-based tool that allows access to the configuration files from a Web server. However, due to the fact that the Sendmail Pro administration tool does not know about Oracle *i*FS, the tool cannot be used to configure Oracle *i*FS. If the tool is used to configure Sendmail pro, Oracle *i*FS support will not be included in the new configuration. Sendmail administrators will need to use the .m4 and .mc files provided in the distribution to reconfigure Sendmail.

Using E-mail with Oracle *i*FS

To use e-mail:

1. Make sure that Sendmail is running.
2. Make sure the SMTP and IMAP servers are running.
3. Make sure the Outbox Agent is running.

See Also

For more information on using Server Manager to check that these servers and agents are running, see Chapter 8, "Using Server Manager to Start and Stop Servers".

Note: Using Server Manager and the `ifsstart` and `ifsstop` commands will also start and stop sendmail. Use the command `/usr/lib/sendmail -bd -q15m` to start Sendmail manually.

Using the Windows NT File System Protocol Server

To provide Explorer-based access to an Oracle *i*FS instance running on Windows NT, use the Windows NT file system protocol server (NTFS). This file system protocol server allows the Oracle *i*FS repository to appear as a local drive on the Windows NT system running the Oracle *i*FS server. Once the Oracle *i*FS repository is mounted as a local drive, the Oracle *i*FS repository can be shared using the standard Windows networking functions.

Accessing Oracle *i*FS through the Windows NT File System Protocol Server

When the Oracle *i*FS repository is mounted as a local hard drive using the file system protocol server, access to the Oracle *i*FS repository is controlled by Windows NT. The standard Windows networking functions can be used to control shared access to the local Oracle *i*FS drive from Windows client machines.

To enable network access from Windows client machines, the Oracle *i*FS NT file system protocol server must be started and the Oracle *i*FS repository mounted as a local drive on the server machine. The local Oracle *i*FS drive can then be shared by the server machine using the Windows Explorer or the NET SHARE command from the command line. A Windows client machine can then map to the shared drive using the Windows Explorer or the NET USE command from the command line.

When a Windows client maps to a shared drive, the Windows client supplies a username and password to the Windows server machine. The Windows server machine checks that the user is a valid Windows user, either on the local machine or in the domain, and grants access to the shared drive. If the shared drive is an Oracle *i*FS drive, the Windows username used to map the drive is also passed to the Oracle *i*FS repository as the Oracle *i*FS user to determine the user's Oracle *i*FS access permissions. In order to be able to access a shared Oracle *i*FS drive, the Windows user used to map the Oracle *i*FS drive must also be an Oracle *i*FS user and an Oracle *i*FS user must also be a Windows user.

Starting the Windows NT File System Protocol Server

The Oracle *i*FS administrator username, by default is system. Any Oracle *i*FS user can be granted administrator privileges. Any user who is a member of the Windows NT administrator group can start the Oracle *i*FS Windows NT File System Driver. However, when connecting from the local machine that user will always appear as system.

Users connecting from a remote machine must have a valid Windows NT account on the machine hosting the NTFS driver. When they connect to the local machine, Windows NT will first validate that they have permission to see the mount points created by the NTFS driver (Root, Home and Public). Once they mount one of these mount points (which is managed by Windows File Sharing, not Oracle *i*FS), their access permission to files and folders inside the mount point is governed by the rights granted to the user they connected as.

Note: Using Windows NT, access to the Oracle *i*FS mount points is controlled by the native Microsoft Windows NT file sharing capabilities, and the NTFS driver simply exposes three more mount points for Windows NT file sharing to manage. This is different from the UNIX implementation where the Oracle *i*FS SMB Server actually manages the File Sharing directly.

See Also

- For more information on starting the Windows NT file system protocol server and using Server Manager, see Chapter 8, "Using Server Manager to Start and Stop Servers".

Backup and Recovery of Data

There are two important issues regarding backup and recovery of data:

- Saving the software installation as a backup
- Saving the data

Saving the Software Installation

You should back up the software installation so that you can easily restore the files in case of a disk crash or other failure. To save the software, you should make a backup on a type of media, such as tape or CD-Rom. This only needs to be done once after completing Oracle *i*FS installation.

Saving the Data

All Oracle *i*FS data is currently stored inside the database, which allows you to rely on the usual backup and recovery mechanism used for all other Oracle data. This is an important step when migrating data; if you use FTP to migrate data, all versioning, ACLs, ownership of files, and extended attributes will be lost. It is highly recommended that you backup your data before you migrate your data.

For more information on using Oracle's back up and recovery mechanism, see the *Oracle 8.1.6 Server Backup and Recovery Guide*. This guide explains what to do for the different possible scenarios, what software is required, and what to set up.

When saving files, there is no way of restoring a single document from a backup. The database is saved in a way which does not allow you to restore complete data files or to restore until a defined point in time, to save the document with all repository data describing it, or to restore this information back into the system.

Configuring Oracle Apache Web Server

To use the Oracle Apache Web Server with Oracle *i*FS, run the `ifsapachesetup` script located in the following directory:

Platform	Location
UNIX	`$ORACLE_HOME/ifs<version>/bin`
Windows NT	`%ORACLE_HOME%\ifs<version>\bin`

This script configures Apache and Jserv so that Oracle *i*FS can be accessed through port 7777 and allows users to access the Oracle *i*FS site through the `http://<server-name>:7777/ifs/ifsservlet` URL.

The sections relevant to Oracle *i*FS configuration are identified in the server configuration files. The following files are changed during configuration:

- jserv.conf
- jserv.properties
- zone.properties

These files can be found in the following directory:

Platform	Location
UNIX	`$ORACLE_HOME/Apache/Jserv`
Windows NT	`%ORACLE_HOME%\Apache\Jserv`

All other configuration is similar to the standard Apache and Jserv configuration.

Configuring Apache As Your Web Server on Windows NT

If you want to use Apache instead of JWS, this section provides configuration steps for Apache to be used with Oracle *i*FS. These are not comprehensive instructions and are more of a general set of rules you need to follow to configure Apache. It is expected that you are familiar with Apache and standard servlet engines.

These steps provide a mechanism to set Apache 1.3.12 with JRun 2.3.3 as the servlet engine. On the Apache Web site (www.apache.org), you can download the appropriate version. JRun can be purchased from allaire.com.

To install Apache and JRun:

1. Install Apache with Dynamic Shared Object (DSO) support.
2. Uncomment the following line in `httpd.conf` file to enable the mod_rewrite:

   ```
   LoadModule rewrite_module modules/ApacheModuleRewrite.dll
   ```

3. Install JRun and run the JRunConnector.
4. While Running the Jrun Connecter, select the Use 1.3.6 DSO option from the DSO Support options.

To setup JRun to run Oracle *i*FS:

1. Place the correct class path, library path and Java Virtual Machine (JVM) arguments into the following directory:

   ```
   <JRUN_HOME>/jsm-default/properties/jsm.properties
   ```

   An example set of entries are:

   ```
   jrun.rootdir=..
   jrun.api21=false
   redirect.stderr=true
   redirect.stdout=true
   admin.port=57860
   admin.bindaddress=*
   java.path=..\\examples\\jni;D:\\Oracle\\Ora81\\lib;D:\\Oracle\\Ora81\\Ct
   x\\lib;C:\\JRun\\jre\\lib;D:\\Oracle\\Ora81\\jdbc\\lib;
   java.args=-classpath
   "C:/JRun/classes;C:/JRun/jsm-default/classes;C:/JRun/lib/jrun.jar;C:/JRu
   n/lib/servlet.jar;C:/JRun/webl/WebL.jar;C:/JRun/lib/jsp.jar;C:/JRun/lib/
   xml4j.jar;C:/JRun/lib/xt.jar;C:/JRun/lib/fesi.jar;C:/JRun/lib/cfanywhere
   .jar;C:/JRun/lib/NetComponents.jar;C:/JRun/lib/OROMatcher.jar;C:/JRun/li
   b/tools.jar;C:/JRun/instantdb/classes;C:/JRun/lib/jrunadmin/jrunadmin.ja
   r;D:/Oracle/Ora81/ifs/jre/lib/classes.zip;D:/Oracle/Ora81/ifs/jre/lib/rt
   .jar;D:/Oracle/Ora81/ifs/jre/lib/i18n.jar;D:/Oracle/Ora81/ifs/jre/jre/li
   b/rt.jar;D:/Oracle/Ora81/ifs/jre/jre/lib/i18n.jar;D:/Oracle/Ora81/ifs/jr
   e/lib/tools.jar;C:/JRun/lib/jrunadmin/swing.jar;D:/Oracle/Ora81/ifs/sett
   ings;D:/Oracle/Ora81/ifs/jre/lib/rt.jar;D:/Oracle/Ora81/ifs/jre/lib/i18n
   .jar;D:/Oracle/Ora81/jdbc/lib/classes111.zip;D:/Oracle/Ora81/lib/vbjorb.
   jar;D:/Oracle/Ora81/jlib/xmlparserv2.jar;D:/Oracle/Ora81/ifs/lib/repos.j
   ar;D:/Oracle/Ora81/ifs/lib/adk.jar;D:/Oracle/Ora81/ifs/lib/email.jar;D:/
   Oracle/Ora81/ifs/lib/tools.jar;D:/Oracle/Ora81/ifs/lib/utils.jar;D:/Orac
   le/Ora81/ifs/lib/release.jar;D:/Oracle/Ora81/assistants/jlib/jnls.jar;D:
   /Oracle/Ora81/ifs/custom_
   classes;D:/Oracle/Ora81/ifs/lib/installer.jar;D:/Oracle/Ora81/ifs/lib/je
   wt-opt-4_0_15.zip;D:/Oracle/Ora81/ifs/lib/share-opt-1_1_
   4.zip;D:/Oracle/Ora81/ifs/lib/swingall.jar;D:/Oracle/Ora81/ifs/lib/clien
   tlib.jar;D:/Oracle/Ora81/ifs/webui_
   classes;D:/Oracle/Ora81/ifs/lib/http.jar;D:/Oracle/Ora81/ifs/lib/webui.j
   ar;D:/Oracle/Ora81/ifs/jws/lib/servlet.jar;D:/Oracle/Ora81/ifs/jws/lib/j
   st.jar;D:/Oracle/Ora81/ifs/jre/lib/javac.jar;"
   java.exe=D:\\Oracle\\Ora81\\ifs\\jre\\bin\\jrew.exe
   java.securitymanager=
   java.classpath=
   ```

2. Map the DocumentServlet and IfsJspServlet to aliases such as docservlet and jspservlet, respectively, in the following directory:

   ```
   <JRUN_HOME>/jsm-default/services/jse/properties/servlets.properties
   ```

   Assuming the Oracle *i*FS schema password as "ifsuser", an example set of entries are:

   ```
   servlet.docservlet.code=oracle.ifs.protocols.http.servlet.DocumentServle
   t
   servlet.docservlet.args=ifs.http.servicepassword=ifsuser
   servlet.docservlet.preload=true
   servlet.jspservlet.code=oracle.ifs.protocols.http.jsp.IfsJspServlet
   servlet.jspservlet.args=ifs.http.servicepassword=ifsuser
   servlet.jspservlet.preload=true
   ```

3. Set up Apache to forward all requests to JRun in the following directory:

   ```
   <APACHE_HOME>/conf/httpd.conf
   ```

   An example set of entries are:

   ```
   RewriteEngine on
   RewriteRule ^/(.*)\.jsp(.*) /ifsjsps/$1.jsp$2 [PT]
   RewriteRule ^/(.*)$ /ifsdocs/$1 [PT]
   ```

4. Set up JRun to map all requests to the DocumentServlet and IfsJspServlet in the following directory:

   ```
   <JRUN_HOME>/jsm-default/services/jse/properties/rules.properties
   ```

   For example, the `rules.properties` looks like:

   ```
   *.jsp=jspservlet
   /ifsjsps/=jspservlet
   /ifsdocs/=docservlet
   ```

5. Start Apache and JRun using the appropriate instructions for each.

10

Using Oracle *i*FS Manager for Development Tasks

The second section of the Oracle *i*FS Manager Navigator contains a setting that controls objects within Oracle *i*FS. As a system administrator, developers may ask you to perform simple development tasks involving these objects. Many of these tasks can be performed using XML. For more information, see the *Oracle Internet File System Developer's Guide*.

Topics include:

Class Objects: An Overview

Using Oracle *i*FS Manager, you can create, modify, and delete class objects. A class object represents a class whose instances are persistently stored and managed by the Oracle *i*FS repository. During installation, Oracle *i*FS creates a number of class objects. Some of these are used internally; others are for commonly encountered data types in content management applications. An application can use the class objects provided, or create its own by subclassing them. Each class object may define one or more attributes for its data. In addition to these, a class object inherits the attributes of its superclass.

You can subclass an existing class by creating a new class object and specifying the parent class for the new class object. The new class object inherits all the attributes of the parent class and allows you to define additional attributes. These activities will be commonplace as they are the easiest way to customize Oracle *i*FS without writing Java or XML code.

For example, to create a custom document type, you would create a new class object, such as purchase order, with a parent class of document and specify any additional attributes you want associated with the purchase order (Due Date, Customer Number, etc.). You can subclass object types using Oracle *i*FS Manager by creating new class objects.

The Complete Class Hierarchy

The Oracle *i*FS class hierarchy is presented below. Note that the classes are subdivided into three groups under each abstract superclass: public object, system object, and schema object. The class hierarchy shows which child classes inherit from which parent classes. For example, Primary User Profile is a child class of User Profile and inherits all the attributes of User Profile. You can add or create new classes through subclassing.

> **Note:** The JavaDoc does not expose all classes listed in this class hierarchy, although those not exposed are still listed.

```
PublicObject
    AccessControlList
        SystemAccessControlList
        ClassAccessControlList
    ApplicationObject
```

```
        ContentQuota
        PropertyBundle
            PolicyPropertyBundle
            ValueDefaultPropertyBundle
            ValueDomainPropertyBundle
            ServerDetail
            ServerRequest
        ServerSubClass
    Category
        MountPoint
    DirectoryObject
        DirectoryGroup
            AdministrationGroup
        DirectoryUser
    Document
        MailDocument
    Family
    Folder
        Mailbox
        Message
        MailFolder
    SearchObject
    SelectorObject
    Template
    UserProfile
        PrimaryUserProfile
        ExtendedUserProfile
            EmailExtendedUserProfile
    VersionSeries
    VersionDescription
```

SystemObject

```
    AccessControlEntry
    AuditEntry
    AuditRule
    ContentObject
    ExtendedPermission
    Format
    Media
        MediaFile
        MediaLob
            MediaBlob
        MediaReference
    PermissionBundle
    Policy
    Property
    Relationship
        BranchRelationship
        FolderRelationship
            FolderPathRelationship
                BodyPartPathRelationship
        GroupMemberRelationship
        NamedRelationship
```

SchemaObject

```
    Attribute
    ClassDomain
    ClassObject
    ValueDomain
    ValueDefault
```

Creating Class Objects

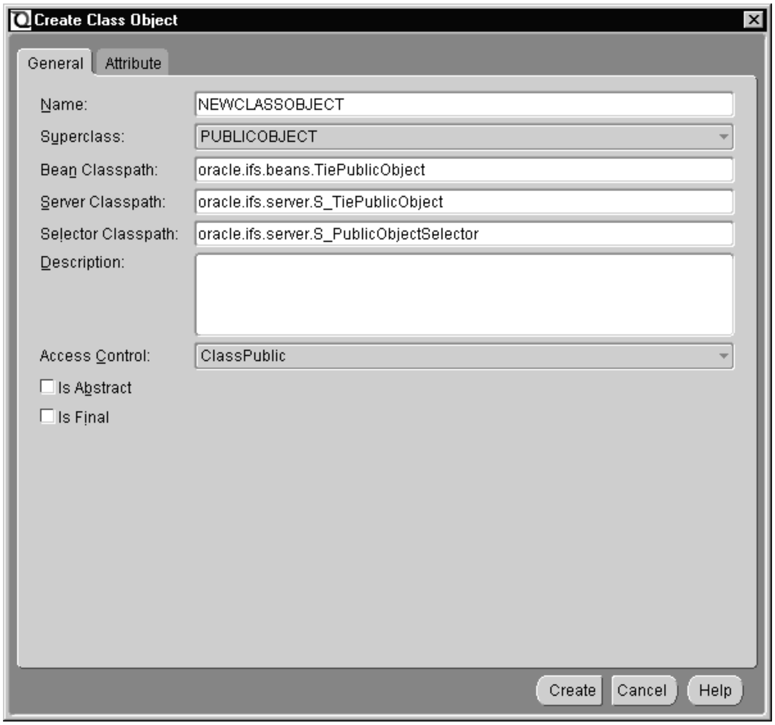
Creating a new class object involves the following tasks:

1. Define the class object information, including the class object name, superclass, and server classpath.
2. Add extended attributes to the class object. Extended attributes are attributes particularly defined for this class object. Inherited attributes are attributes of the parent class object or superclass and cannot be modified.

To create a new class object:

1. Click Create on the toolbar.
2. Select class object from the Select Object Type dialog. You can also use the Create Like option.

3. Click Create. The Create Class Object window displays.



Create Class Object

Name: NEWCLASSOBJECT

Superclass: PUBLICOBJECT

Bean Classpath: oracle.ifs.beans.TiePublicObject

Server Classpath: oracle.ifs.server.S_TiePublicObject

Selector Classpath: oracle.ifs.server.S_PublicObjectSelector

Description:

Access Control: ClassPublic

☐ Is Abstract

☐ Is Final

4. Enter information in the following fields:

   - Name (required)—The name of the new class object. Each class object must have a unique name
   - Superclass—The superclass of the new class object. Every class object must have a superclass. The superclass can be any class object that is not "final".
   - Bean Classpath—The fully-qualified name of the Java class that is instantiated to represent the instances of the new class object in the development ("beans") API. Defaults to the bean classpath of the superclass class object. It is recommended that you accept this default. If the default is accepted, you can access the extended attributes of an instance of the new class object by calling the `getAttribute` and `setAttribute` methods inherited `oracle.ifs.beans.LibraryObject`. If you specify a value other than the default, you must manually create a new Java class whose

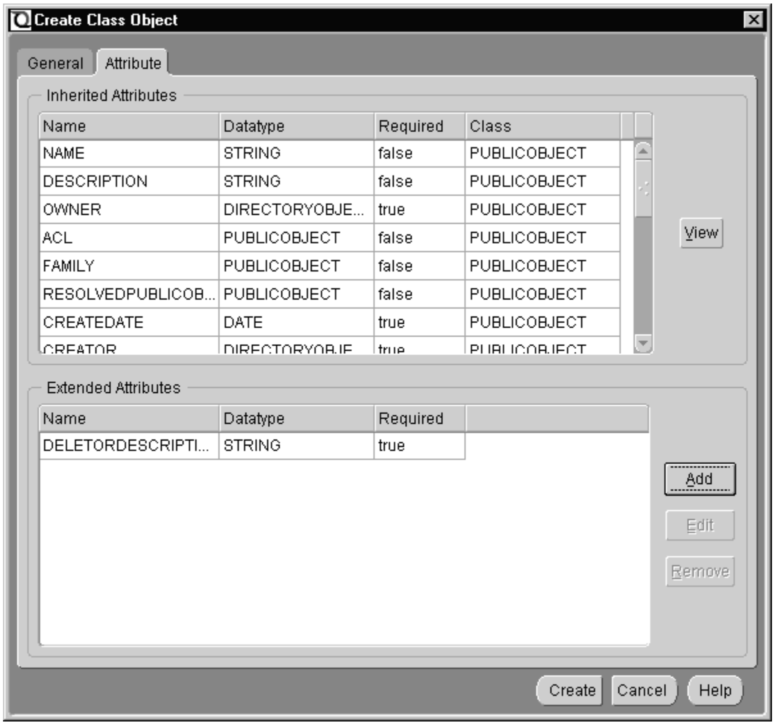
name is the value specified. You must compile this new class and ensure it is in the Java classpath before instances of the new class object can be instantiated. This is an advanced topic.

- Server Classpath—The fully-qualified name of the Java class that is instantiated to represent the instances of the new class object in the extensibility ("server") API. Defaults to the server classpath of the superclass class object. It is recommended that you accept this default. If you accept the default, you can access the extended attributes of an instance of the new class object by calling the `getAttribute` and `setAttribute` methods inherited from `oracle.ifs.server.S_LibraryObject`. If you specify a value other than the default, you must manually create a new Java class whose name is the value specified. You must compile this new class and ensure it is in the Java classpath before instances of the new class object can be instantiated. This is an advanced topic.
- Selector Classpath—Defaults to the superclass selector classpath. Always accept this default. Changing this value is not supported for this release
- Description (optional)—A description of the new class object.
- Access Control—The class ACL for the new class object. Class ACLs control who can perform certain operations on instances of the class object, such as creating new instances of a class object or selecting existing instances of a class object.
- Is Abstract—Check this box to make the new class object "abstract". Abstract class objects cannot be instantiated. Making a class object abstract is only useful if you intend to further subclass it (with those subclasses being non-abstract).
- Is Final—Check this box to make the new class object "final". Final class objects cannot be subclassed.

5. If extended attributes are not needed for this class object, click Create. If you intend to create extended objects, see "Add Extended Attributes".

Add Extended Attributes

1. Click the Attribute tab to add extended attributes. The Attribute tab property sheet displays:



Inherited Attributes

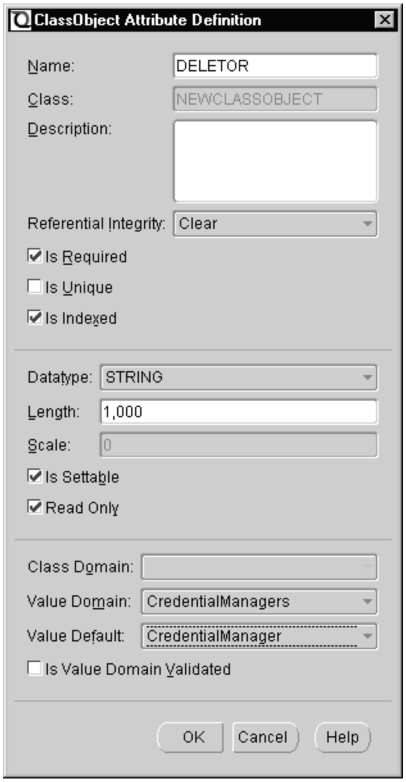
Name	Datatype	Required	Class
NAME	STRING	false	PUBLICOBJECT
DESCRIPTION	STRING	false	PUBLICOBJECT
OWNER	DIRECTORYOBJE...	true	PUBLICOBJECT
ACL	PUBLICOBJECT	false	PUBLICOBJECT
FAMILY	PUBLICOBJECT	false	PUBLICOBJECT
RESOLVEDPUBLICOB...	PUBLICOBJECT	false	PUBLICOBJECT
CREATEDATE	DATE	true	PUBLICOBJECT
CREATOR	DIRECTORYOBJE	true	PUBLICOBJECT

Extended Attributes

Name	Datatype	Required
DELETORDESCRIPTI...	STRING	true

2. To view an existing attribute, select the attribute and click View. A new window displays with the attribute information.

3. To add an extended attribute, click Add. The Class Object Attribute Definition window displays:



ClassObject Attribute Definition

Name: DELETOR

Class: NEWCLASSOBJECT

Description:

Referential Integrity: Clear

☑ Is Required

☐ Is Unique

☑ Is Indexed

Datatype: STRING

Length: 1,000

Scale: 0

☑ Is Settable

☑ Read Only

Class Domain:

Value Domain: CredentialManagers

Value Default: CredentialManager

☐ Is Value Domain Validated

4. Enter information in the following fields:

   - Name (required)—The name of the attribute. Each attribute must have a name that is unique across both the other attributes of the new class object as well as all the attributes it inherits from its superclasses.
   - Description—A description for the new attribute.
   - Referential Integrity—Allows the referential integrity behavior of the new attribute to be defined. Referential integrity rules apply only to attributes of the following datatypes: PUBLICOBJECT, DIRECTORYOBJECT, PUBLICOBJECT_ARRAY, and DIRECTORYOBJECT_ARRAY. The referential integrity rule value is ignored for all other datatypes. For these four datatypes, a referential integrity rule value of RESTRICT prevents a

PUBLICOBJECT or DIRECTORYOBJECT (or subclass) referenced by instances of this attribute from being freed. This is analogous to a "DELETE RESTRICT" database constraint. A referential integrity rule value of CLEAR allows a PUBLICOBJECT or DIRECTORYOBJECT (or subclass) referenced by instances of this attribute to be freed, causing the value of the referencing attribute (or attribute element, for array-type attributes) becomes null.

- Is Required—If checked, instances of the attribute must have a non-null value. Oracle *i*FS treats zero-length, i.e. "empty", STRING attributes as null.
- Is Unique—If checked, each non-null instance of the attribute must have a unique value.
- Is Indexed—If checked, the attribute is indexed. This allows searches and selectors that are qualified by the attribute to perform more quickly.
- Datatype—Select a data type from the drop-down list.
- Length—For a data type of string, the maximum length of instances of the attribute, in bytes. The number of characters to which this corresponds depends upon both the database character set and the value stored.
- Scale—Unused for this release.
- Is Settable—In creating a new instance of this class object, whether a value for this attribute can be explicitly set in the definition class for this class object. If unchecked, the attribute is "system-set", meaning that its value can be only be set in the extensibility ("server") API.
- Read Only—Whether the value of this attribute in an existing instance of this class object can be updated. If checked, the attribute is "non-updateable," and can only be changed in the extensibility API. Making an attribute read only is useful for enforcing business rules that prevent data from changing over time.
- Class Domain—Applies only to attribute of the following data types: PUBLICOBJECT, DIRECTORYOBJECT, SYSTEMOBJECT, SCHEMAOBJECT, PUBLICOBJECT_ARRAY, DIRECTORYOBJECT_ ARRAY, and SYSTEMOBJECT_ARRAY, SCHEMAOBJECT_ARRAY. Specifies an optional class domain on attributes of these datatypes. A class domain performs basic data validation upon creating an instance of a class object or updating the attributes of an existing instance of a class object. For example, without a class domain, a PUBLICOBJECT attribute could be set to any PUBLICOBJECT -- a DOCUMENT, a FOLDER, etc. You could use a class domain to restrict this to, for example, only DOCUMENTs, or only

DOCUMENTs and any subclasses of DOCUMENTs. See "Creating Class Domains" for more information on class domains.

- Value Domain—Specifies an optional value domain for the new attribute. A value domain performs basic data validation upon creating an instance of a class object or updating the attributes of an existing class object. For example, a value domain could ensure a STRING attribute is a valid color—"red", "green", "blue"—or a INTEGER attribute lies within a certain range—0 to 10 inclusive. See "Creating Class Domains" for more information on value domains.
- Value Default—Specifies an optional default value the new attribute. In creating a new instance of a class object, Oracle *i*FS sets this value for this attribute if no value is otherwise provided. See the "Creating Value Domains" for more information on value domains.
- Is Value Domain Validated—Specifies whether the value domain applied to this attribute is enforced; in other words, in creating a new instance of this class object or updating an existing instance of this class object, whether Oracle *i*FS ensures the value of this attribute (if set or changed) lies within the specified value domain.

5. When all the information has been entered, click OK.
6. On the Create Class Object window, click Create.

Modifying Class Objects

The only modification allowed for class objects is adding or editing an extended attribute. To add an extended attribute, see "Add Extended Attributes".

To edit an extended attribute:

1. Select the class object to edit in the Navigator.
2. On the property sheet, click the Attribute tab.
3. Select the extended attribute to edit and click Edit.
4. On the Class Object Attribute Definition window, edit the appropriate fields and click OK.
5. Click Apply on the property sheet.

Deleting Class Objects

To delete a class object:

1. Select the class object you want to delete in the Navigator and click the Delete button on the toolbar. System class objects cannot be deleted.
2. Click Yes at the prompt.

Class Domains: An Overview

Using Oracle *i*FS Manager, you can create and modify class domains. A class domain is used in conjunction with class objects. For attributes that point to objects, you can limit values to specific classes by using class domains. For example, the AccessControlList attribute for public objects has a class domain which restricts the value to objects of the AccessControlList class.

Why Use a Class Domain?

Each attribute has a datatype, such as int or String. As you would expect, Oracle *i*FS supports:

- The standard Java datatypes, such as int, long, double, boolean, String, and Date.
- Their array variants.

In addition, Oracle *i*FS supports four top-level object datatypes unique to Oracle *i*FS:

- PublicObject
- SystemObject
- SchemaObject
- DirectoryObject

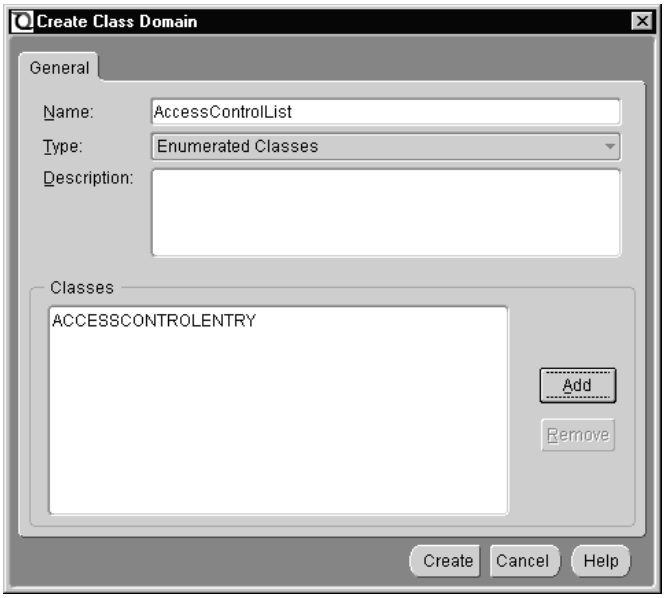
You could use a value domain to validate any of these four datatypes. But, if you want to be more specific and limit an attribute's value to a single class, you need a validation technique to ensure that the value entered is not, for example, just any Public Object, but specifically, a Document object. In this case, you would use a class domain to specify the single class that the attribute's datatype must match.

If an attribute's datatype is a subclass of one of the four unique Oracle *i*FS top-level datatypes listed above, you can specify a class domain on that attribute. Oracle *i*FS will throw an exception if the attribute is of any other class.

Creating Class Domains

To create and maintain class domain using Oracle *i*FS Manager:

1. Click Create on the toolbar.
2. Select Class Domain from the Select Object Type dialog. You can also use the Create Like option.
3. Click Create. The Create Class Domain window displays.



Create Class Domain

General

Name: AccessControlList

Type: Enumerated Classes

Description:

Classes

ACCESSCONTROLENTRY

4. Enter information on the following fields:
   - Name (required)—The name of the new class domain. Each class domain must have a unique name.
   - Type—Choose Enumerated Classes to include only the classes listed in the Classes field in the new class domain. Choose Enumerated Classes and

Subclasses to include the classes listed in the Classes field and any of their subclasses.

- Description (optional)—A description of the class domain.
- Classes—The classes in the class domain. In creating a new instance of a class object or updating an existing instance of a class object, the values of any attributes to which this class domain has been applied, if set or changed, must be instances of one of these classes (or their subclasses, if Type is Enumerated Classes and Subclasses).

5. Click Create.

Modifying Class Domains

When modifying a class domain, the Description field is the only field that can be updated. To modify a selected class domain:

1. Select the class domain to be modified in the Navigator.
2. Enter a new description on the property sheet.
3. Click Apply.

Deleting Class Domains

If a class domain is deleted, the data validation performed by that class domain no longer occurs. To delete a class domain:

1. Select the class domain to be deleted in the Navigator.
2. Click Delete on the toolbar.
3. Click Yes at the prompt.

Value Domains: An Overview

A value domain is a separate class which can contain a list of values used by developers for custom applications. You can limit attribute values to specific values of a class domain by using value domains. Limiting the possible values helps with data validation. User input is checked against the list, and if the input value is not on the list, Oracle *i*FS throws an exception.

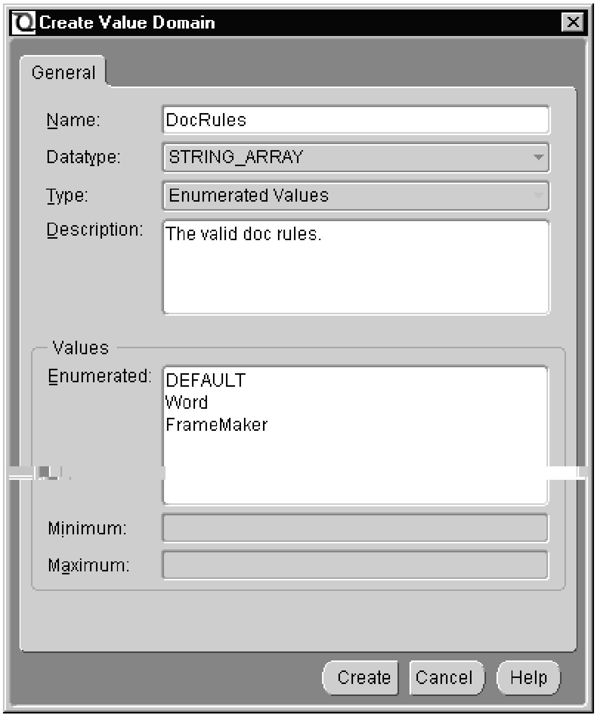
Why Use a Value Domain?

In many cases, the number of possible valid values for an attribute is limited. A frequently used form of data validation takes the value entered and checks it against a list of valid values, returning an error message if the value entered is not valid.

Creating Value Domains

To create a new value domain with Oracle *i*FS Manager:

1. Click the Create button on the toolbar.
2. Select Value Domain from the Select Object Type dialog. You can also use the Create Like option.
3. Click Create. The Create Value Domain dialog displays.



Create Value Domain

General

Name: DocRules

Datatype: STRING_ARRAY

Type: Enumerated Values

Description: The valid doc rules.

Values

Enumerated: DEFAULT
Word
FrameMaker

Minimum:

Maximum:

4. Enter information in the following fields:

   - Name (required)—The name of the new value domain. Each value domain must have a unique name.
   - Datatype—The datatype of the value domain. This determines to which attributes the new value domain may be applied. For example, an INTEGER_ARRAY value domain may be applied to attributes whose datatypes are either INTEGER or INTEGER_ARRAY.
   - Type—The type of the value domain.
     - Enumerated values: the acceptable values in the value domain are defined by a set. For example, the integer values 1, 2, 3, 5, 7.
     - An inclusive range, defined by a minimum and maximum. For example, the integer values 0 to 10, inclusive.
     - An inclusive minimum. For example, the integers greater-than-or-equal-to 3.
     - An inclusive maximum. For example, the integer values less-than-or-equal-to 14.
     - An exclusive range, defined by a minimum and maximum. For example, the integer values 14 to 69, not including either 14 or 69.
     - An exclusive minimum. For example, the integers greater-than 3 (but not including 3 itself).
     - An exclusive maximum. For example the integers less-than 14 (but not including 14 itself).
   - Enumerated values—Applies only if Type is Enumerated Values. The values in the set, one per line.
   - Enumerated Values—Applies only for certain Types. The minimum value.
   - Minimum / Maximum—Applies only for certain Types. The maximum value.

5. Click Create.

Modifying Value Domains

The Description field is the only field that can be updated.

1. Select the value domain to be modified in the Navigator.
2. Enter a new description using the property sheet.
3. Click Apply.

Deleting Value Domains

To delete a value domain:

1. Select the value domain to be deleted in the Navigator.
2. Click Delete on the toolbar.
3. Click Yes at the prompt.

Value Defaults: An Overview

By using value defaults, you can specify an initial value for an attribute. If no value is set, the repository sets the default value for the attribute. Should the default value be outside the value domain when the value domain validation is on, the Oracle *i*FS server throws an exception upon setting the attribute with the invalid default value.

Why Use a Value Default?

Value defaults are particularly useful if you need to apply the same default value to a large number of attributes. Using a value default lets you specify the default value once, and apply it to many attributes. If the default value needs to be changed, you can then change the value default in one place, and all attributes that use the value default will now have access to the new value.

Example

Assume that ExpenseReport has an attribute called ApproverName. You could use a value default on ApproverName to specify the name of your department’s vice president, Chris Stevens. When an ExpenseReport object is created, "Chris Stevens" would be the default Approver Name. This same ApproverName attribute might be used for approving hiring requests, as well as approving expense reports. If there was a reorganization and Kim Naser became the new vice president, you could change the value default object, and the new name would become the default approver for both expense reports and hiring requests.

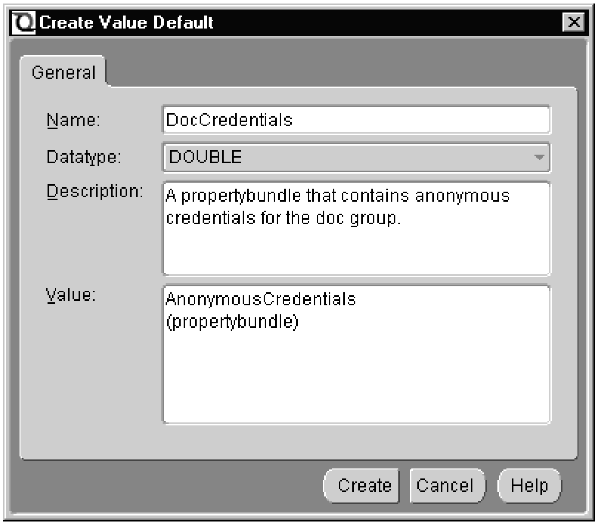
Note: changing the value default in this way would not affect any earlier expense reports or hiring requests, which would continue to indicate "Chris Stevens."

Only expense reports or hiring requests created after the ValueDefault was updated to "Kim Naser" would have the new name.

Creating Value Defaults

To create a new value default with Oracle *i*FS Manager:

1. Click the Create button on the toolbar.
2. Select Value Default from the Select Object Type dialog. You can also use the Create Like option.
3. Click Create. The Create Value Default window displays.



Create Value Default

General

Name: DocCredentials

Datatype: DOUBLE

Description: A propertybundle that contains anonymous credentials for the doc group.

Value: AnonymousCredentials (propertybundle)

4. Enter information in the following fields:
   - Name (required)—The name of the new value default. Each value default must have a unique name.
   - Datatype—The datatype of the new value default. This determines to which attributes the new value default may be applied. For example, an INTEGER value default may be applied to attributes whose data type is either INTEGER or INTEGER_ARRAY. However an INTEGER_ARRAY value default may only be applied to attributes whose data type is INTEGER_ARRAY.
   - Description (optional)—A description for the value domain.
   - Values—The value of the new value default.
5. Click Create.

Modifying Value Defaults

The Description field is the only field that can be updated.

1. Select the value default to be modified in the Navigator.
2. Enter a new description using the property sheet.
3. Click Apply.

Deleting Value Defaults

To delete a value default:

1. Select the value default to be deleted in the Navigator.
2. Click Delete on the toolbar.
3. Click Yes at the prompt.

Agents and Overrides: An Overview

Agents automate tasks in Oracle *i*FS. An agent is on option for customizing Oracle *i*FS. A developer writes a custom agent using Java. To run the agent, you will need to add it to a Server Manager .def file.

An override is another option for customizing Oracle *i*FS. A developer creates an overrides using Java. Overrides extend or replace methods in the Oracle *i*FS API to perform tasks specific to your business needs.

What Is an Agent?

An agent is a Java program used to automate a task. More specifically, an agent lets an application respond to specific actions taken within the Oracle *i*FS environment. An agent may respond when a certain type of document is inserted, updated, or deleted in the repository. Agents are written against the Java API. They can be triggered by a specific event, such as when a document is created, or by time, every 24 hours. Agents are especially useful for cleanup or background tasks when the machine is not heavily loaded, or for notification, such as sending e-mail notifications.

What is an Override?

An override is a Java program that overrides default server-side behavior. The difference between agents and overrides is that agents are run asynchronously, as separate activities at timed intervals, or as a response to a completed transaction. An override is processed during the transaction, and should be used when the behavior needs to occur immediately or if Oracle *i*FS needs to complete the behavior and react to it before finishing the transaction.

See Also

- For more information on agents and using the agent template, see the *Oracle Internet File System Developer's Guide*.
- For more information on the Server Manager .def files, see Chapter 8, "Using Server Manager to Start and Stop Servers".

- For more information on overrides, see the *Oracle Internet File System Developer's Guide*.

Using Java Server Pages to Display Documents in Oracle *i*FS

Java Server Pages (JSPs) are the recommended manner of creating Web-based applications based on Oracle *i*FS. (In fact, the Oracle *i*FS Web interface uses JSPs.) JSPs use Java Beans to extract data from the repository and display information on the Web by generating HTML documents. Although JSPs provide a way for users to view and manipulate documents, providing functionality parallel to that of renderers, using a JSP is not technically "rendering" because it does not involve a specific renderer class. The JSP lookup is created with Oracle *i*FS Manager.

JSP Execution

The `ifs/jsp-bin` folder is provided for developers to place JSPs to be executed. The list of these JSP executable directories is configurable. This is set in the property `ifs.http.jsp.alloweddirs` which is located in the `IfsHttpServletParameters.properties` file.

To add more directories where JSPs to be created can be placed, this parameter needs to be modified. This parameter is located in the `IfsHttpServletParameters.properties` file, which is located in the following directory:

```
$ORACLE_HOME/ifs/settings/oracle/ifs<version>/protocols/http/properties
```

For example, the value is set to `/ifs/webui/jsps|/ifs/jsp-bin`, the directory `/home/myhome/jsps` can be used to host JSPs that need to be executed. Please note that the **|** is used as a delimiter between directories.

Note: You cannot remove either of the two default directories.

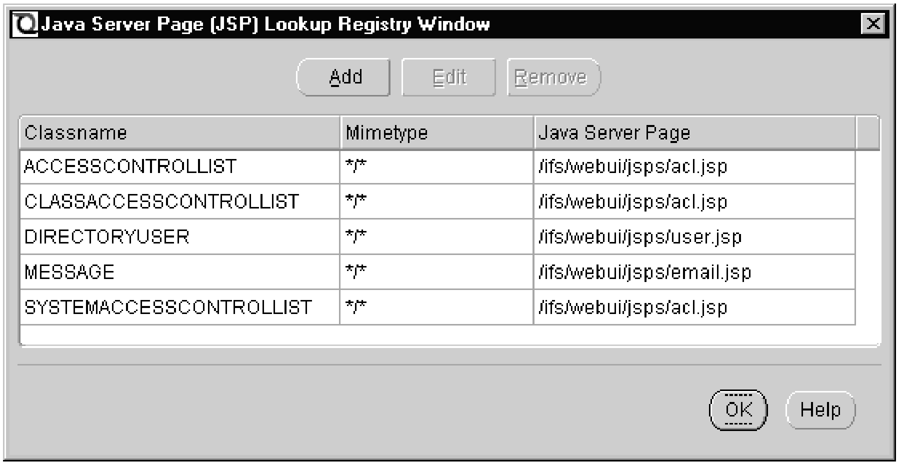
See Also

- For more information on JSPs, see the *Oracle Internet File System Developer's Guide*.

Registering JSPs

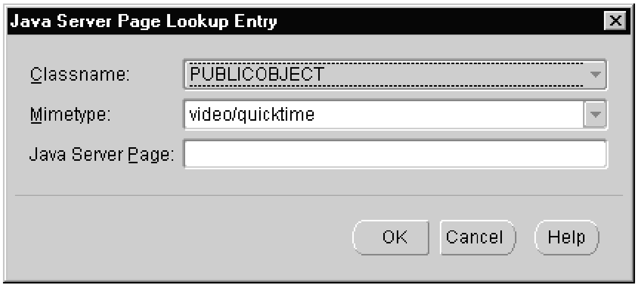
You can view, register, and edit JSPs. When a JSP is added, it is added to the JSP registry. To register a renderer:

1. From the Object menu, select Register.
2. Select Java Server Page (JSP) Lookup from the Select Object Type dialog.
3. Click Register. The Java Server Page (JSP) Lookup Register Window displays with the existing registered JSPs listed.



Classname	Mimetype	Java Server Page
ACCESSCONTROLLIST	*/*	/ifs/webui/jsps/acl.jsp
CLASSACCESSCONTROLLIST	*/*	/ifs/webui/jsps/acl.jsp
DIRECTORYUSER	*/*	/ifs/webui/jsps/user.jsp
MESSAGE	*/*	/ifs/webui/jsps/email.jsp
SYSTEMACCESSCONTROLLIST	*/*	/ifs/webui/jsps/acl.jsp

4. Click Add. The Java Server Page (JSP) Lookup Entry window displays.



Classname: PUBLICOBJECT

Mimetype: video/quicktime

Java Server Page:

5. Enter information in the following fields:
   - Classname—Select a classname for the JSP.

- Mimetype—Select a mimetype for the JSP.
- Java Server Page—Enter the JSP information. This is the complete path of the JSP inside of the Oracle *i*FS repository.

6. Click OK.
7. Click OK on the Java Server Page (JSP) Lookup Registry Window to return to the Oracle *i*FS Manager Navigator.

Modifying JSPs

To edit an existing JSP:

1. Access the Java Server Page (JSP) Lookup Registry Window as described in "Registering JSPs".
2. Select the JSP to be edited and click Edit. The Edit JSP window displays.
3. Edit the appropriate fields and click OK.
4. Click OK on the Java Server Page (JSP) Lookup Registry Window to return to the Oracle *i*FS Manager Navigator.

Deleting JSPs

To remove an existing JSP:

1. Access the Java Server Page (JSP) Lookup Registry Window as described in "Registering JSPs".
2. Select the JSP to be removed and click Remove.
3. Click Yes at the prompt.
4. Click OK on the Java Server Page (JSP) Lookup Registry Window to return to the Oracle *i*FS Manager Navigator.

Parsers: An Overview

A parser is a Java class that extracts attributes from a file as it is loaded into Oracle *i*FS and stores the information in the repository. The parser class creates one or more database objects, such as:

- a Document object
- a Folder object

When you want to create objects in Oracle *i*FS, you can either use the standard parsers provided, or, if your application requires special treatment, you can create a custom parser.

The IfsXmlParser is the standard parser class that can be registered using Oracle *i*FS Manager. The IfsXmlParser creates an object in the Oracle *i*FS repository from an XML document body and is used as the default parser for all XML documents stored in Oracle *i*FS.

There are three XML parsers that the dispatcher parser (IfsXmlParser) recognizes:

- IfsSimpleXmlParser—Used for parsing Oracle *i*FS objects.
- LiteralDocumentParser—Performs DTD validation (optional), and stores the XML document.
- CustomXmlParser—Used for parsing custom objects.

See Also

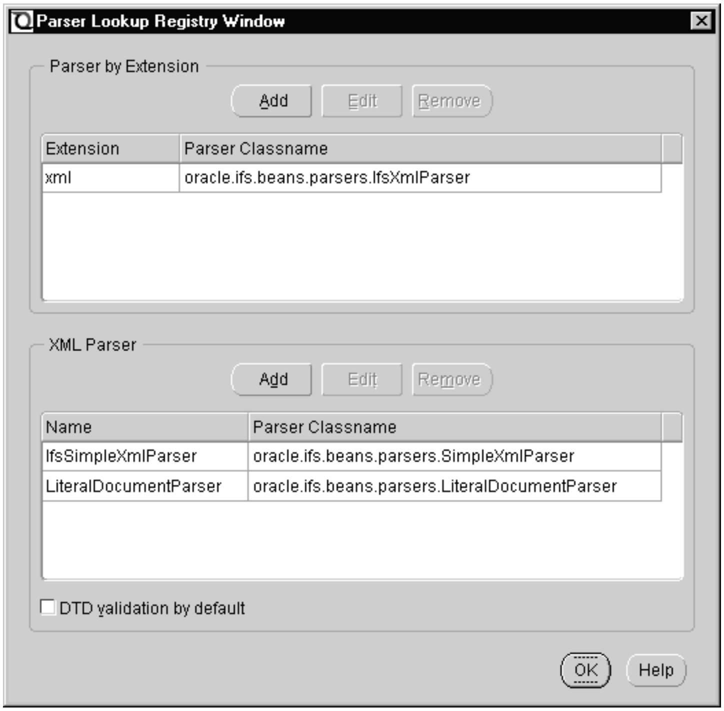
- For more information on parsers and creating custom parsers, see the *Oracle Internet File System Developer’s Guide*.

Registering Parsers

Using Oracle *i*FS Manager, you can register custom parsers created for special applications, and edit or delete parsers. When a parser is added, it is added to the parser registry. To register a custom parser:

1. From the Object menu, select Register.
2. Select Parser Lookup from the Select Object Type dialog and click Register.

   The Parser Lookup Registry Window displays with the existing parsers by extension and XML parsers listed. This dialog provides two functions: registering a parser by extension and registering an XML parser.



Extension	Parser Classname
xml	oracle.ifs.beans.parsers.IfsXmlParser

Name	Parser Classname
IfsSimpleXmlParser	oracle.ifs.beans.parsers.SimpleXmlParser
LiteralDocumentParser	oracle.ifs.beans.parsers.LiteralDocumentParser

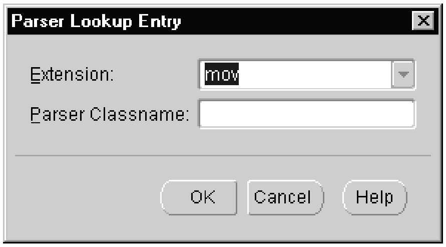
☐ DTD validation by default

   Also select the `DTD validation by default` checkbox to perform a DTD validation.

Registering a Parser by Extension

To register a parser by extension, do the following:

1. Click Add in the Parser by Extension box. The Parser Lookup Entry dialog displays.



Extension: mov

Parser Classname:

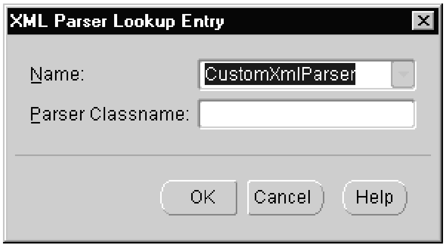
2. Select an extension for the custom parser.
3. Enter the parser classname and click OK.
4. Click OK on the Parser Lookup Registry Window to return to the Oracle *i*FS Manager Navigator.

Registering an XML Parser

In the XML Parser box, you can select the Do DTD validation by default box to have DTD validation as the default.

To register an XML parser, do the following:

1. Click Add in the XML Parser box. The Parser Lookup Entry dialog displays.



Name: CustomXmlParser

Parser Classname:

2. Enter a name for the XML Parser.
3. Enter a parser classname and click OK.

See Also

- For more information on parsers and creating custom parsers, see the *Oracle Internet File System Developer's Guide*.

Modifying Parsers

To modify an existing parser:

1. Access the Parser Lookup Registry Window as discussed in "Registering Parsers".
2. Select the parser to be edited and click Edit. The Parser Lookup Entry dialog displays.
3. Edit the appropriate fields and click OK.
4. Click OK on the Parser Lookup Registry Window to return to the Oracle *i*FS Manager Navigator.

Deleting Parsers

To delete an existing parser:

1. Access the Parser Lookup Registry Window as discussed in "Registering Parsers".
2. Select the parser to be removed and click Remove.
3. Click Yes at the prompt.
4. Click OK on the Parser Lookup Registry Window to return to the Oracle *i*FS Manager Navigator.

Renderers: An Overview

A renderer accesses information stored in Oracle *i*FS and outputs the information in a certain format. In a sense, it is the opposite of a parser. While the information output by a renderer may be identical to the document as it was input, it doesn't have to be. Once information has been stored in Oracle *i*FS, it can be rendered in any format the developer defines. For example, an XML renderer can perform the following task:

- Take a document that was input as HTML (using an HTML parser).
- Render it in an XML format.

Data can be combined from several documents and displayed as a single, new document. For example, information that is collected from e-mail messages may be displayed dynamically in a Web browser window.

The rendered representation is typically some combination of the following:

- Object attributes
- Static content (bytes or characters)
- Content (for Document objects)

The following table lists the Oracle *i*FS standard renderer classes provided out-of-the-box.

Class	Description
MFC822 Renderer	Renders a message object into a stream of bytes comprising the message in RFC822 format.
Octet Stream Renderer	Renders a document into a stream of bytes corresponding to the bytes in the document's content object.

See Also

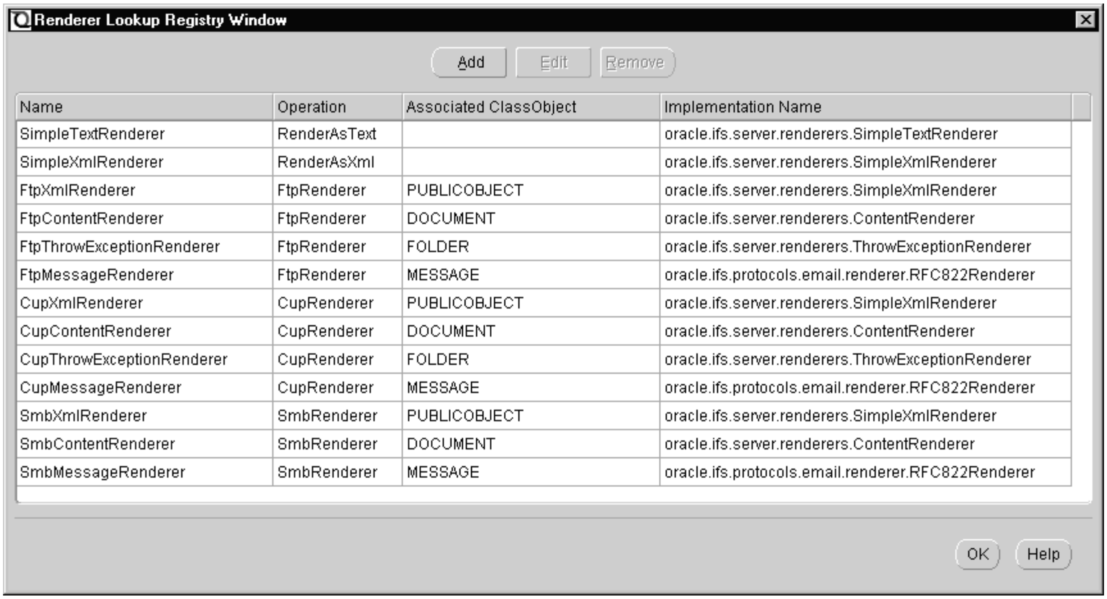
- For more information on renderers, see the *Oracle Internet File System Developer's Guide*.

Registering Renderers

Using Oracle *i*FS Manager, you can register and edit custom renderers. When a renderer is added, it is added to the renderer registry. To add and register a renderer:

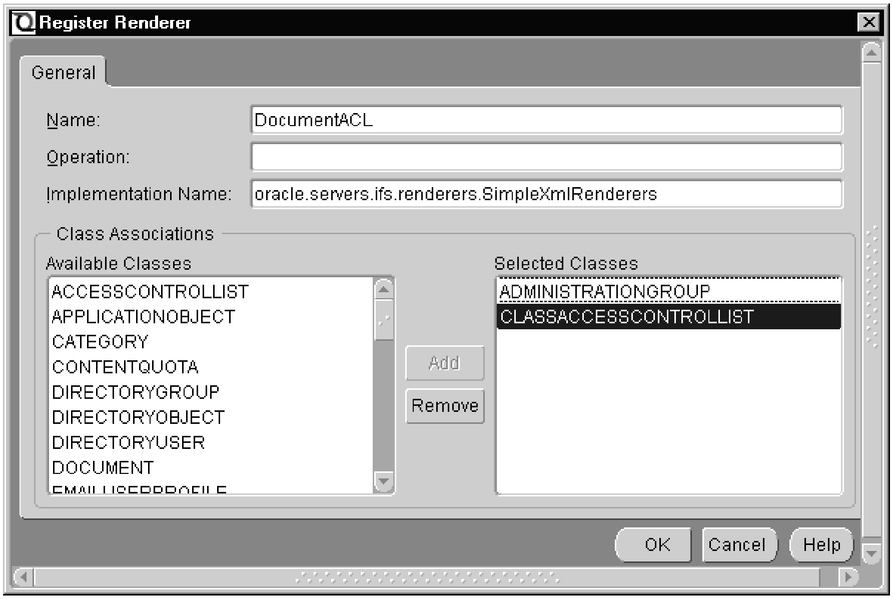
1. From the Object menu, select Register.
2. Select Renderer Lookup from the Select Object Type dialog.

3. Click Register. The Renderer Lookup Registry Window displays.



Name	Operation	Associated ClassObject	Implementation Name
SimpleTextRenderer	RenderAsText		oracle.ifs.server.renderers.SimpleTextRenderer
SimpleXmlRenderer	RenderAsXml		oracle.ifs.server.renderers.SimpleXmlRenderer
FtpXmlRenderer	FtpRenderer	PUBLICOBJECT	oracle.ifs.server.renderers.SimpleXmlRenderer
FtpContentRenderer	FtpRenderer	DOCUMENT	oracle.ifs.server.renderers.ContentRenderer
FtpThrowExceptionRenderer	FtpRenderer	FOLDER	oracle.ifs.server.renderers.ThrowExceptionRenderer
FtpMessageRenderer	FtpRenderer	MESSAGE	oracle.ifs.protocols.email.renderer.RFC822Renderer
CupXmlRenderer	CupRenderer	PUBLICOBJECT	oracle.ifs.server.renderers.SimpleXmlRenderer
CupContentRenderer	CupRenderer	DOCUMENT	oracle.ifs.server.renderers.ContentRenderer
CupThrowExceptionRenderer	CupRenderer	FOLDER	oracle.ifs.server.renderers.ThrowExceptionRenderer
CupMessageRenderer	CupRenderer	MESSAGE	oracle.ifs.protocols.email.renderer.RFC822Renderer
SmbXmlRenderer	SmbRenderer	PUBLICOBJECT	oracle.ifs.server.renderers.SimpleXmlRenderer
SmbContentRenderer	SmbRenderer	DOCUMENT	oracle.ifs.server.renderers.ContentRenderer
SmbMessageRenderer	SmbRenderer	MESSAGE	oracle.ifs.protocols.email.renderer.RFC822Renderer

4. Click Add to add an entry for a custom renderer in the registry. The Register Renderer dialog displays.



Name: DocumentACL

Operation:

Implementation Name: oracle.servers.ifs.renderers.SimpleXmlRenderers

Class Associations

Available Classes

ACCESSCONTROLLIST
APPLICATIONOBJECT
CATEGORY
CONTENTQUOTA
DIRECTORYGROUP
DIRECTORYOBJECT
DIRECTORYUSER
DOCUMENT
EMAILUSERPROFILE

Selected Classes

ADMINISTRATIONGROUP
CLASSACCESSCONTROLLIST

5. Enter information in the following fields:
   - Name (required)—The name of the renderer.
   - Operation—The renderer policy key.
   - Implementation Name—The renderer implementation class name.
6. Select from the Available Classes the classes to be associated with this renderer. Press the Add button to include the selected classes.
7. Click OK.
8. Click OK on the Renderer Lookup Registry Window to return to the Oracle *i*FS Manager Navigator.

Modifying Renderers

To modify an existing renderer:

1. Access the Renderer Lookup Registry Window as described in "Registering Renderers".
2. Select the renderer to be edited and click Edit. The Edit Renderer dialog displays. The fields that cannot be edited display as grayed-out.
3. Edit the appropriate fields and click OK.
4. Click OK on the Renderer Lookup Registry Window to return to the Oracle *i*FS Manager Navigator.

Deleting Renderers

To delete an existing renderer:

1. Access the Renderer Lookup Registry Window as described in "Registering Renderers".
2. Select the renderer to be removed and click Remove.
3. Click Yes at the prompt.
4. Click OK on the Renderer Lookup Registry Window to return to the Oracle *i*FS Manager Navigator.

Formats/Mimetypes: An Overview

Oracle *i*FS associates a format (mimetype) with each document. The format of a document specifies the way the document information is encoded. Internet browsers need to know the format of documents to determine how to store content and index their contents. The format determines:

- MIME type—Specifies the type of content stored in Oracle *i*FS.
- Extension type—Specifies the default extension to use for storing Oracle *i*FS content as a file.
- Binary setting—Determines whether the content stored in Oracle *i*FS is of binary type.
- Index setting—Determines whether the content of this format should be indexed in Oracle *i*FS.

You can either:

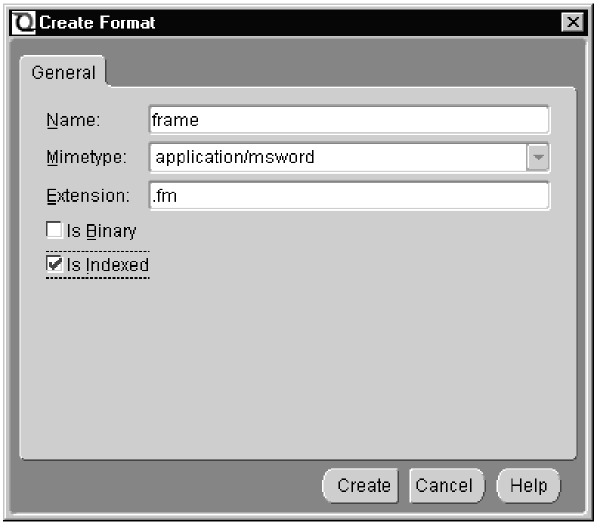
- Use the default system formats—Using these requires no administration.
- Supplement the default system formats—This is usually necessary only if you use custom applications or have special content requirements.

See Also

For more information on formats / mimetypes, see the *Oracle Internet File System Developer's Guide.*

Creating Formats/Mimetypes

1. Click the Create button on the toolbar.
2. Select Format / Mimetype from the Select Object Type dialog. You can also use the Create Like option.
3. Click Create. The Create Format dialog displays.



Create Format

General

Name: frame

Mimetype: application/msword

Extension: .fm

☐ Is Binary

☑ Is Indexed

4. Enter information in the following fields:
   - Name (required)—The name of the format / mimetype.
   - Mimetype—Specifies the type of content stored in Oracle *i*FS.

- Extension—Specifies the default extension to use for storing Oracle *i*FS content as a file.
- Is Binary—Determines whether the content stored in Oracle *i*FS is of binary type.
- Is Indexed—Determines whether the content of this format should be indexed in Oracle *i*FS.

5. Click Create.

Modifying a Format/Mimetype

To modify a format:

1. Select the format/mimtype to be modified in the Navigator.
2. On the property sheet, modify the following fields:
   - Mimetype
   - Extension
   - Is Binary
   - Is Indexed
3. Click Apply.

Deleting a Format/Mimetype

To delete a format/mimetype:

1. Select the format/mimetype to be deleted in the Navigator.
2. Click Delete on the toolbar.
3. Click Yes at the prompt.

11

Oracle *i*FS Repository Views

Using the Oracle *i*FS repository views, experienced database administrators can see the effects of changes they make to the Oracle *i*FS repository in a familiar format. This chapter provides information on the SQL views provided with Oracle *i*FS and an example of each. The following topics are included:

- Using the Oracle iFS Repository Views for System Maintenance
- Oracle iFS Has Eight Different Oracle iFS Repository Views

Using the Oracle *i*FS Repository Views for System Maintenance

Repository views are created when Oracle *i*FS is installed. The Oracle *i*FS repository views are created when Oracle *i*FS is installed. The views are owned by the repository owner: ifssys. The Oracle *i*FS SQL read-only views are created with the Oracle *i*FS schema and are for informational purposes only. Grant the SELECT permission to users who need access to these views.

Note: The Oracle *i*FS views should be used exclusively by system administrators. The data in the views bypass all security; therefore, caution should be taken in granting users access to these views.

The names of objects in the view may not be unique. Therefore, IDs are provided. Any column of type `date` has two columns in the view, for example:

- `CREATE_DATE`—An internal representation of the create date.

`CREATE_DATE_GMT`—The standard Oracle8 date format, based on Greenwich Mean Time.

Oracle *i*FS Has Eight Different Oracle *i*FS Repository Views

These eight repository views are provided out-of-the-box. Each view is described in a separate table which provides the name of the column, whether the column can be null, the type of column, and the length for each view.

- `ifs_versioned_documents`
- `ifs_all_documents`
- `ifs_folders`
- `ifs_folder_items`
- `ifs_users`
- `ifs_groups`
- `ifs_group_members`
- `ifs_acls`

View: ifs_versioned_documents

This view shows a list of all documents which are versioned.

Column Name	Null?	Type of Column and Length
Modified_By	Yes	VARCHAR2 (700)
Modifier_ID	Yes	NUMBER (20)
Creator	Yes	VARCHAR (700)
Creator_ID	Yes	NUMBER (20)
Owner	Yes	VARCHAR (700)
Owner_ID	Yes	NUMBER (20)
FAMILY	Yes	VARCHAR2(700)
FAMILY_ID	Yes	NUMBER(20)
FORMAT	Yes	VARCHAR2(700)
FORMAT_ID	Yes	NUMBER(20)
DOCUMENT_SIZE	Yes	NUMBER(20)
CHARACTERSET	Yes	VARCHAR2(40)
LANGUAGE	Yes	VARCHAR2(40)
MEDIA	Yes	VARCHAR2(700)
MEDIA_ID	Yes	NUMBER(20)
ACL	Yes	VARCHAR2(700)
ACL_ID	Yes	NUMBER(20)
NAME	Yes	VARCHAR2(700)
DOCUMENT_ID	Yes	NUMBER(20)
DESCRIPTION	Yes	VARCHAR2(2000)
CREATE_DATE	Yes	NUMBER(20)
CREATE_DATE_GMT	Yes	DATE
MODIFIED_DATE	Yes	NUMBER(20)
MODIFIED_DATE_GMT	Yes	DATE
EXPIRATION_DATE	Yes	NUMBER(20)

Column Name	Null?	Type of Column and Length
EXPIRATION_DATE_GMT	Yes	DATE
LOCKED	Yes	NUMBER(1)
READ_BY_OWNER	Yes	NUMBER(1)

View: ifs_all_documents

This view shows a list of all documents.

Column Name	Null?	Type of Column and Length
MODIFIED_BY	Yes	VARCHAR2 (700)
MODIFIER_ID	Yes	NUMBER (20)
CREATOR	Yes	VARCHAR (700)
CREATOR_ID	Yes	NUMBER (20)
OWNER	Yes	VARCHAR (700)
OWNER_ID	Yes	NUMBER (20)
FORMAT	Yes	VARCHAR2(700)
FORMAT_ID	Yes	NUMBER(20)
DOCUMENT_SIZE	Yes	NUMBER(20)
CHARACTERSET	Yes	VARCHAR2(40)
LANGUAGE	Yes	VARCHAR2(40)
MEDIA	Yes	VARCHAR2(700)
MEDIA_ID	Yes	NUMBER(20)
ACL	Yes	VARCHAR2(700)
ACL_ID	Yes	NUMBER(20)
NAME	Yes	VARCHAR2(700)
DOCUMENT_ID	Yes	NUMBER(20)
DESCRIPTION	Yes	VARCHAR2(2000)
CREATE_DATE	Yes	NUMBER(20)
CREATE_DATE_GMT	Yes	DATE

Column Name	Null?	Type of Column and Length
MODIFIED_DATE	Yes	NUMBER(20)
MODIFIED_DATE_GMT	Yes	DATE
EXPIRATION_DATE	Yes	NUMBER(20)
EXPIRATION_DATE_GMT	Yes	DATE
LOCKED	Yes	NUMBER(1)
READ_BY_OWNER	Yes	NUMBER(1)

View: ifs_folders

This view shows all folders and their attributes.

Column Name	Null?	Type of Column and Length
MODIFIED_BY	Yes	VARCHAR2 (700)
MODIFIER_ID	Yes	NUMBER (20)
CREATOR	Yes	VARCHAR (700)
CREATOR_ID	Yes	NUMBER (20)
OWNER	Yes	VARCHAR (700)
OWNER_ID	Yes	NUMBER (20)
ACL	Yes	VARCHAR2(700)
ACL_ID	Yes	NUMBER(20)
NAME	Yes	VARCHAR2(700)
FOLDER_ID	Yes	NUMBER(20)
DESCRIPTION	Yes	VARCHAR2(2000)
CREATE_DATE	Yes	NUMBER(20)
CREATE_DATE_GMT	Yes	DATE
MODIFIED_DATE	Yes	NUMBER(20)
MODIFIED_DATE_GMT	Yes	DATE
EXPIRATION_DATE	Yes	NUMBER(20)
EXPIRATION_DATE_GMT	Yes	DATE

Column Name	Null?	Type of Column and Length
LOCKED	Yes	NUMBER(1)

View: ifs_folder_items

This view lists all folders and the top level items.

Column Name	Null?	Type of Column and Length
CREATOR	Yes	VARCHAR (700)
CREATOR_ID	Yes	NUMBER (20)
OWNER	Yes	VARCHAR (700)
OWNER_ID	Yes	NUMBER (20)
TYPE	No	VARCHAR2(700)
NAME	Yes	VARCHAR2(700)
DESCRIPTION	Yes	VARCHAR2(2000)
CREATE_DATE	Yes	NUMBER(20)
CREATE_DATE_GMT	Yes	DATE
OBJECT_ID	Yes	VARCHAR2(700)
ACL	Yes	VARCHAR2(700)
ACL_ID	Yes	NUMBER(20)
FOLDER_NAME	Yes	VARCHAR2 (700)
FOLDER_ID	Yes	NUMBER(20)

View: ifs_users

This view lists all users.

Column Name	Null?	Type of Column and Length
MODIFIED_BY	Yes	VARCHAR2 (700)
MODIFIER_ID	Yes	NUMBER (20)
CREATOR	Yes	VARCHAR (700)

Column Name	Null?	Type of Column and Length
CREATOR_ID	Yes	NUMBER (20)
OWNER	Yes	VARCHAR (700)
OWNER_ID	Yes	NUMBER (20)
ACL	Yes	VARCHAR2(700)
ACL_ID	Yes	NUMBER(20)
NAME	Yes	VARCHAR2(700)
USER_ID	Yes	NUMBER(20)
DESCRIPTION	Yes	VARCHAR2(2000)
CREATE_DATE	Yes	NUMBER(20)
CREATE_DATE_GMT	Yes	DATE
DISTINGUISHED_NAME	Yes	VARCHAR2(700)
ADMIN_ENABLED	Yes	NUMBER(1)
CREDENTIAL_MANAGER	Yes	VARCHAR2(700)
UNIQUE_NAME	Yes	VARCHAR2(700)
HOME_FOLDER_NAME	Yes	VARCHAR2(700)
HOME_FOLDER_ID	Yes	NUMBER(20)

View: ifs_groups

This view lists all groups.

Column Name	Null?	Type of Column and Length
MODIFIED_BY	Yes	VARCHAR2 (700)
MODIFIER_ID	Yes	NUMBER (20)
CREATOR	Yes	VARCHAR (700)
CREATOR_ID	Yes	NUMBER (20)
OWNER	Yes	VARCHAR (700)
OWNER_ID	Yes	NUMBER (20)
NAME	Yes	VARCHAR2(700)

Column Name	Null?	Type of Column and Length
GROUP_ID	Yes	NUMBER(20)
DESCRIPTION	Yes	VARCHAR2(2000)
CREATE_DATE	Yes	NUMBER(20)
CREATE_DATE_GMT	Yes	DATE
ACL	Yes	VARCHAR2(700)
ACL_ID	Yes	NUMBER(20)

View: ifs_group_members

This view lists groups and their direct members.

Column Name	Null?	Type of Column and Length
MEMBER_NAME	Yes	VARCHAR2 (700)
MEMBER_ID	Yes	NUMBER (20)
MEMBER_DESCRIPTION	Yes	VARCHAR (700)
MEMBER_CREATE_DATE	Yes	NUMBER (20)
MEMBER_CREATE_DATE_GMT	Yes	DATE
GROUP_ID	Yes	NUMBER(20)
TYPE	No	VARCHAR2(2000)

View: ifs_acls

This view lists all ACLs in Oracle *i*FS.

Column Name	Null?	Type of Column and Length
CONTROLLING_ACL_NAME	Yes	VARCHAR2 (700)
CONTROLLING_ACL_ID	Yes	NUMBER (20)
NAME	Yes	VARCHAR2(700)
ACL_ID	Yes	NUMBER(20)
DESCRIPTION	Yes	VARCHAR2(2000)

Column Name	Null?	Type of Column and Length
CREATE_DATE	Yes	NUMBER(20)
CREATE_DATE_GMT	Yes	DATE
MODIFIED_BY	Yes	VARCHAR (700)
MODIFIER_ID	Yes	NUMBER (20)
CREATOR	Yes	VARCHAR (700)
CREATOR_ID	Yes	NUMBER (20)
OWNER	Yes	VARCHAR2(700)
OWNER_ID	Yes	NUMBER(20)

Example: Selecting Attributes of a Document

This example uses ifs_versioned_documents to select attributes. To accomplish this, the following command is run:

```
 SQL> select name, format, document_size, media, document_id,
acl, owner from ifs_versioned_documents
```

The following table displays the output.

Name	Format	Document_ Size	Media	ACL	Document_ ID	Owner
Business Plan	MS Word	11954	IndexedBlob	Published	7902	jsmith
June Report	Postscript	121	NonIndexedBlob	Private	7936	lfrench
Oracle logo	JPEG	121	NonIndexedBlob	Published	7926	pbrown

12

What Your Users Need to Know

This chapter gives a general overview of the concepts your users will need to know in order to use Oracle *i*FS:

- What Your Users Need to Know About Working with the Windows Interface
- What Your Users Need to Know About Working with the Web Interface

What Your Users Need to Know About Working with the Windows Interface

Your users can work with Oracle *i*FS through Microsoft Windows. Often, this is the primary tool they use to access Oracle *i*FS. To use the extended functions of Oracle *i*FS, such as versioning and check in/check out of files requires users to install the Oracle Oracle *i*FS Utilities.

Installing the Oracle Oracle *i*FS Utilities

To install the Oracle *i*FS Utilities, users need to run the `Setup.exe` program. This file is located in the following directory:

Platform	Directory
UNIX	`Root/ifs<version>/winui/install`
Windows NT	`Root\ifs<version>\winui\install`

You need to decide if users should install the Oracle *i*FS Utilities in the default directory, or in an alternate directory. If you choose an alternate directory, you need to provide the name of that alternate directory to users.

Installing the Oracle *i*FS Utilities is a standard installation task, so tell your users to plan to do installation when they aren’t in the middle of several other tasks. Also, make sure your users shut down all other applications when they begin the installation process.

Instruct your users to do the following to install the Oracle *i*FS Utilities:

1. Close down all Windows programs before beginning. You must de-install the previous version before continuing. After de-installing, map a network drive to Oracle *i*FS using SMB before continuing.
2. Locate the `Setup.exe` file and run the `Setup.exe` program.
3. Follow all instructions and accept the defaults unless you are told otherwise by the system administrator.

   By default, the Oracle *i*FS Utilities are installed in the following directory:

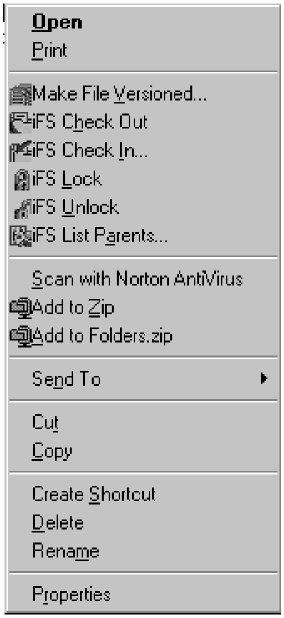
   ```
   c:\Program Files\Oracle\iFS<version>
   ```

The installation process includes installing several items. For Oracle *i*FS to be installed correctly, instruct your users to install all recommended items.

4. Reboot your machine after installing the Oracle *i*FS Utilities.

Right-Click an Oracle *i*FS Folder or File to Access Oracle *i*FS Functions

When users use Windows Explorer, their Oracle *i*FS drives look just like any other drive, or mount points. Users can drag and drop files into Oracle *i*FS, cut and paste, copy and paste, or perform any of the other operations they normally perform in a file management system. In addition, if a user right-clicks an Oracle *i*FS folder or file, additional commands appear on the pop-up menu. For example:



The following table describes the commands and a description of their function:

Command	Description
Make File Versioned	Specify that each time the file is saved, the previous version of the file is archived.
Oracle *i*FS Check In...	Release a versioned file so that others can use it.

Command	Description
Oracle *i*FS Check Out	Reserve a versioned file so that others cannot update it.
Oracle *i*FS Lock	Lock an unversioned file so that others can use it.
Oracle *i*FS Unlock	Unlock an unversioned file so that others can use it.
List Parents	List the folders in which the file appears. A file may appear in multiple folders. Only one copy of the file exists. That copy is linked to the folders.
Properties	View file attributes, security information, and version history.

Connecting to Mount Points

To connect to mount points using Windows, users can either browse the network or map a drive.

To Connect to a Mount Point By Browsing:

1. The user finds the server using Network Neighborhood.
2. The user selects the server by clicking. A list of mount points and their descriptions displays.
3. The user drills down through those descriptions. By drilling down, the user is automatically connected through the mount point to the appropriate folder path.

To Connect to a Mount Point By Mapping a Drive:

1. In Windows Explorer, the user selects Map Network Drive from the Tools menu.
2. In the field provided, the user types the server name and the desired mount point using the following syntax:

   ```
   \\<servername>\<virtualpathname>
   ```

   The default mount point "MyHome" displays in the browse list (in Windows, but not the Web interface). This maps automatically to the user's home directory. This mount point cannot be administered through the typical administration operations, it is embedded in the SMB server.

   The following list describes the possible mount points:

   - `\\<servername>\root`—The top level of the Oracle *i*FS hierarchy.

- `\\<servername>\MyHome`—The user’s home folder.
- `\\<servername>\public`—The public area.

See Also

- For information on mapping an Oracle *i*FS drive, see the *Oracle Internet File System User’s Guide*.

What Your Users Need to Know About Working with the Web Interface

The Oracle *i*FS Web interface provides your users a way of accessing and using Oracle *i*FS using a Web browser. Users can:

- Navigate the Folder Hierarchy—Users can navigate up or down the hierarchy by clicking one of the parent directories or the name or title of the folder.
- Open a File—Users can open a file by clicking the name or title of the file. The file opens with the appropriate application for users to view or edit on their local operating systems.
- Save or Upload a File—Users can save or upload files through FTP and HTTP.
- Delete a File or Folder—Users can delete a file or folder by clicking the Delete icon for the file or folder they want to delete.
- Rename a File or Folder—Users can rename a file or folder by selecting the item and selecting Rename from the Edit menu.
- Use the Content Management functions—Users can make files versioned, check them in and out, search the contents, and lock and unlock files.

Supply Your Users with the URL for the Web Interface

Your users access the Oracle *i*FS server through a URL. However, to log into Oracle *i*FS, users will need to point their browser to the Login page. You can change the URL for the Login page. The URL for this page is usually:

```
http://<server_name>/
```

The `<server_name>` is either the name you assigned to Oracle *i*FS server or the server’s IP address.

Accessing Oracle *i*FS with a Web browser requires no special plug-ins or other browser features. All standard browsers, such as Netscape Navigator and Microsoft Internet Explorer, can access Oracle *i*FS.

To Connect to a Mount Point Using the Web Interface:

When users log into the Web interface, they are connected to all mount points for which they have discovery permission.

Typically, it is not important for users to understand what directory in the overall Oracle *i*FS folder hierarchy they are in when they log into a particular mount point, and there is no way through Windows for them to determine this. As far as users are concerned, these are isolated islands of information. There may be overlap in the mount points; for example, a user could have two mount points mapped, one of which is inside another.

13

Oracle *i*FS Log Files and Troubleshooting Information

This chapter provides information on troubleshooting and configuration logs. The following topics are included:

- Troubleshooting General Problems
- Using the Oracle iFS Log Files
- Using the Configuration Log Files
- Using the Protocol Server Log Files
- Using the Server Manager Log Files
- The WCP Server Log File
- The Windows Interface Log Files
- Resolution for Upgrading to the Oracle Database
- National Language Support (NLS) Troubleshooting
- Common Administrative Errors

Troubleshooting General Problems

This section contains information to assist with general problems and problem diagnosis.

Oracle *i*FS Configuration Assistant Errors

The following information describes troubleshooting information for problems that occur during configuration of Oracle *i*FS.

Possible Error

When running the Configuration Assistant and trying to create class objects on Oracle 8.1.7 and JDK 1.1.8, the following error occurs:

```
java.lang.NoClassDefFoundError: org/omg/CORBA/userException
```

Possible Problem

If a custom installation of the Oracle 8.1.7 database has been completed and the OEM components were not installed.

Possible Solution

Perform the following steps on your UNIX machine where the Oracle 8.1.7 database is installed.

1. Verify that the following files are NOT present in `$ORACLE_HOME/lib`:
   - vbjob.jar
   - vbjapp.jar
   - vbjtools.jar
2. Run the Oracle Universal Installer for the Oracle 8.1.7 database by executing `/RunInstaller` from the Installation CD. Complete the following steps:
   1. **Available Products Page**

      On this page, select Oracle8*i* Enterprise Edition 8.1.7.0.0. Click Next.
   2. **Installation Types Page**

      On this page, select Custom and click Next.

3. **Available Product Components Page**

   On this page, select the following:

   * Oracle Enterprise Manager Products 8.1.7.0.0.
   * Oracle Enterprise Manager Client 2.2.0.0.0.
   * Oracle Enterprise Manager DBA Management Pack 2.2.0.0.0 (under Oracle Enterprise Manager Client).

   You can deselect all other options and click Next.

4. **Summary Page**

   On this page, verify that the following files are present in `$ORACLE_HOME/lib`:

   * vbjorb.jar
   * vbjapp.jar
   * vbjtools.jar

3. Rerun the Oracle *i*FS Configuration Assistant by executing `$ORACLE_HOME/ifs/bin/ifsconfig`.

Stopping Multiple Server Instances of the Same Name

The following information describes troubleshooting information for problems that occur when shutting down ambiguously named servers.

Possible Error

After running `ifsstop`, the following error message displays:

```
IFS-46114: Server name IfsProtocols is ambiguous; specify the server
identifier:(47471, 47613, 47679, 47863, 47954).
```

Possible Problem

The problem is multiple instances of Server Manager were started with the same name. This can occur when:

- `ifsstarts` is run multiple times without first stopping Oracle *i*FS.

- Making a copy of either `IfsAgents.def` or `IfsProtocols.def` to intentionally start multiple Server Managers, but not modifying the `ManagerName` field to be unique.

`ifsstop` cannot shutdown the Server Managers due to the ambiguity in the names.

Possible Solution

The solution is to manually stop each Server Manager. To do this:

1. Start Oracle *i*FS Manager and select Monitor from the File menu. This will start the Oracle *i*FS Monitor.
2. Select the Server Manager you want to stop and click Stop.

Using the Oracle *i*FS Log Files

When using Oracle *i*FS, there are three types of log files generated. These log files are:

- Configuration Logs—Log files generated during installation.
- Protocol Logs—Log files generated by the protocol servers.
- Server Manager Logs—Log files generated by Server Manager when it starts and stops agents, including the ExternalServer Agent, which starts and stops the protocol servers.

All log files, except the Java Web Server (JWS) log file, are generated and placed in the `$ORACLE_HOME/ifs<version>/log` directory. The JWS log file is placed in the `$ORACLE_HOME/ifs<version>/jws/logs/javawebserver/webpageservice` directory.

Using the Configuration Log Files

During the configuration of Oracle *i*FS, the `IfsConfigOut.log` and the `IfsConfigSql.log` files are generated and stored in the following directory:

Platform	Directory
UNIX	`$ORACLE_HOME/ifs<version>/log`
Windows NT	`%ORACLE_HOME%\ifs<version>\log`

IfsConfigOut.log

The IfsConfigOut.log file is a log of the configuration, starting with saving properties to configuration files, then creating the schema and tablespaces. This log file lists which files are executed, what .adm and .typ files are executed by the AdminManager and CreateClasses.

The IfsConfigOut.log file is similar to the make_create log, but contains more detail. At the top of the file, when the process began is displayed. At the bottom of the file, the time it took for the process to run is displayed. This file is always created in "verbose" mode.

If an error occurs during the Oracle *i*FS configuration, the problem is reported at the bottom of the log file. Notice a large Java stack trace. The trace is printed twice: once, exactly when the error occurred (by the thread actually executing the operation), and again by the main thread when it is displaying the error. Sometimes the first trace has more information, so be sure to examine both traces.

IfsConfigSql.log

This log file contains a SQL "dump" of all the SQL statements that were executed directly from the SQL scripts that create the Oracle *i*FS schema, for example, odmmain.sql. This file is always created in "verbose" mode.

Common Errors Found in the IfsConfigOut.log and IfsConfigSql.log Files

The following list displays the common errors found and a resolution:

- `java.lang "class not found"`—A jar or library file is missing from the $ORACLE_HOME/lib or other directory.
- `java.sql.SQLException: procedure or package not available`—A required package has not been installed in the Oracle software that Oracle *i*FS is being installed on top of, notably, the `DBMS_OBFUSCATION_PACKAGE`, or certain interMedia packages.
- File not found or IO exceptions—Often caused by permission problems. For example, a configuration file has been made read-only by the root user.

Using the Protocol Server Log Files

This section discusses the log files generated by the various protocol servers, such as SMB and FTP.

The SMB Log File

The SMB log file includes a trace level, a trace file, and an append log. For the SMB log file, the trace level can be set to the following:

- OFF—No log file is generated.
- ERR—Default, only prints unexpected exceptions and very high level messages, for example, when users log on and off. Generally, it is more useful to examine the log file after noticing unusual behavior instead of reading the log file looking for errors.
- LOW—Provides information about every operation, such as list directory, open file, and write data, performed by the SMB server. This is a useful level for tracking the general activity of the server.
- HIGH—Provides detailed information on every operation, including most of the packet data. This is typically used for Oracle *i*FS development tasks, but may provide administrators some clues about erroneous behavior. At HIGH level the log file can use up disk space quickly, so only use this while diagnosing or reproducing a specific problem.

TraceFile = {localfile}

The trace file is where the trace will output to, if you do not want it to go to standard output (stdout). Tracing from all clients is written to the same log file, so if you are diagnosing a specific problem, it is probably wise to limit the number of concurrent users, ideally to one.

The most useful lines in the trace file are those starting with a ’<’. If you just want to get a general idea of the operations on the server, you can GREP (Global Regular Expression and Print) for all lines starting with ’<<’. This gives you one line for each SMB command issues, for example:

```
<<SMB_COM_RENAME>>\mydoc.txt
```

You can also show any errors for that command, for example:

```
<<-SMB_COM_RENAME->>DOS/fileexists
```

You can also include lines starting with '<-' if you want an indication of the server side performance.

AppendLog = {true, false}

The default setting for the append log is true. If left with this setting, the new trace information is appended to the old trace file. If set to false, the trace file is overwritten each time the server is started.

The FTP and CUP Log Files

Both the FTP and CUP servers offers the option LogCommands, which list all communications to and from the server. FTP also offers VerboseLogging, which lists every exception, regardless of its significance.

For both protocols, if an exception occurs outside of normal operation, that exception is logged. For normal operation, it is not necessary to keep a log of all commands or exceptions. For diagnosing problems, more information is better. For FTP, this means both LogCommands and VerboseLogging should be set to true. For CUP, this means that LogCommands should be set to true.

Oracle Corporation suggests that administrators run the servers with minimal logging. If a problem occurs, the logs might contain the stack trace (if the exception was not expected) or they might not. Once a problem is suspected with a specific protocol, the more logging the better. The amount of information in the log can become overwhelming quickly, so only turn on the extra logging when diagnosing problems.

FTP

For FTP, there are two login options, LogCommand and VerboseLogging, both in the FtpServer.def file. If the LogCommand is set to true, then the communication, both to and from the server, will be logged. If VerboseLogging is set to true, then all exceptions, regardless of their significance, will be logged. The FtpServer.def file is located in the following directory:

Platform	Directory
UNIX	`$ORACLE_HOME/ifs<version>/settings`
Windows NT	`%ORACLE_HOME%\ifs<version>\settings`

CUP

For CUP, there is only one login option, `LogCommand`. If this option is set to true, then the communication, both to and from the server, will be logged.

The JWS Log Files

Oracle *i*FS HTTP output goes into standard web server log files. In the case of JWS, which is what comes installed with Oracle *i*FS, there are three files to view, all of which are located in the following directory:

Platform	Directory
UNIX	`IFS_HOME/jws/logs/javawebserver/webpageserver`
Windows NT	`IFS_HOME%\jws\logs\javawebserver\webpageserver`

The following list describes each log file:

- `access_log`—Contains a line of information for every web request made to the server. This file is marginally useful.
- `error_log`—Contains error output when things go wrong. This file can be very useful. The things to look for in this file are exception traces and the accompanying error messages. It is safe to ignore "IOException:Connection Reset By Peer" messages. These messages show up when the server is in the middle of serving a request but the user has hit the stop button in the browser.
- `event_log`—Contains mostly start up and shutdown information. This file is marginally useful.

The verbosity is controlled by the Web server configuration, for which there are instructions.

Using the Server Manager Log Files

Each Server Manager instance emits logging information to its own log file. The logging information includes the following categories:

- Start up logging for the Server Manager instance.
- Agent loading and initial starting of each agent registered in the configuration file used by this Server Manager instance.
- Significant agent events and status changes.
- Agent errors and warnings.
- Specific information pertaining to the ExternalServer agent instances. This agent is responsible for managing protocol servers:
  - Protocol server start up and shutdown notifications
  - Protocol server unexpected failures and/or restarts.
- Shutdown logging for the Server Manager instance.

The default Oracle *i*FS installation will configure two Server Manager instances: one that manages the ExternalServer agents (which in turn manage the protocol servers), and another that manages all other agents. By default, the log file produced by each of these two Server Manager instances are IfsProtocols.log and IfsAgents.log, respectively.

How To Control Logging Verbosity and Other General Characteristics

The following table describes the options which can be specified in the Server Manager configuration file to control the amount of logging and other general logging characteristics.

Option Name	Datatype	Description	Example
outputfile	String	The absolute file name for the log file.	`outputfile = /private/oracle/ifs<version/logs/IfsAgents.log`
VerboseLogging	Boolean	Set to true if logging should be verbose. The default is false.	`VerboseLogging = true`
AppendLog	Boolean	Set to true if the specified log file should be appended to rather than overwritten. The default is false, meaning the log will be overwritten.	

The WCP Server Log File

The WCP server has a single log file in the same location as other protocols. It logs the WCP commands received from the Windows interface, the intermediate output when commands are processed, and the WCP response sent back to the Windows interface.

General Notes About the WCP Server Log

The following list describes some general information about the WCP server log.

- After a WCP connection request, each line will be preceded by "WcpSession #", where "#" is a number identifying the WCP session thread created to serve one Windows machine. This is important to identify the relevant events corresponding to a particular client machine as the WCP server is a multi-threaded server.

- The green lines will be logged only if the flay `LogGC=true` is set in the WcpServer.def files. You can use this information to estimate the total memory allocation necessary to support your typical user load, and allocate more or les depending on your requirements.
- Lines with `"XMLCMD>"` are intermediate output when commands are processed.

The Windows Interface Log Files

The Windows interface also has its own log files. There are two types of log files, all located in the "log" subdirectory under the directory where the Windows interface is installed. The following list describes the two types of log files:

- `IfsSockMgr.log`—Logs events such as Oracle *i*FS SocketManager starts and exceptions encountered by SocketManager.
- `IfsWcpSessions?.log`—The ’?’ represents the drive letter to which the Oracle *i*FS (SMB) server is mapped. This file logs any exceptions during the interaction with the WCP server associated to the mapped drive and is normally blank.

Resolution for Upgrading to the Oracle Database

The following information will help when upgrading to the Oracle database. You do not need to have Oracle *i*FS installed nor do you need to de-install Oracle *i*FS before upgrading.

Problem Description

In the process of upgrading your database, the interMedia Text views, packages, indexes need to be re-compiled. If you do not recompile these objects, you will be unable to add or update any documents to Oracle *i*FS. The following error will display:

```
IFS-30002 Unable to create new LibraryObject
```

Solution Description

To compile the interMedia Text objects, follow these steps using SQL*Plus:

1. Log on to SQL*Plus using SQL*Plus.
2. Run the `$ORACLE_HOME/ctx/admin/upgrade/s0801070.sql` file as the SYS user.
3. Run the `$ORACLE_HOME/ctx/admin/upgrade/u0801070.sql` file as CTXSYS.
4. Run the `$ORACLE_HOME/ctx/admin/dr0typec.pkh` file as CTXSYS.
5. Run the `$ORACLE_HOME/ctx/admin/dr0pkh.sql` file as CTXSYS.
6. Run the `$ORACLE_HOME/ctx/admin/dr0plb.sql` file as CTXSYS.
7. Run the `$ORACLE_HOME/ctx/admin/dr0type.plb` file as CTXSYS.
8. Run the `$ORACLE_HOME/ctx/admin/dr0typec.plb` file as CTXSYS.
9. Run the `$ORACLE_HOME/ctx/admin/dr0itypc.sql` file as CTXSYS.
10. Create a SQL script named `recompile.sql` that will recompile all invalid interMedia Text objects. To do this, log on to SQL*Plus as SYS, and type the following:

    ```
    spool recompile.sql;
    set heading off;

    select 'alter '||object_type||' '||owner||'.'||object_name||' compile;'
    from dba_objects
    where status='INVALID'
    and object_type != 'JAVA CLASS';

    spool off;

    exit;
    ```

11. Run recompile.sql as the SYS user.

See Also

- For more information on installing Oracle *i*FS, see the *Oracle Internet File System Installation Guide*.
- For more information on troubleshooting, see the *Oracle Internet File System Installation Guide*.

National Language Support (NLS) Troubleshooting

The following information describes troubleshooting information to assist with general problems with the National Language Support (NLS).

Possible Problem: FTP Client

Unable to `ls` and `put` filenames and folders with non-ASCII characters from the MS-DOS FTP client.

Possible Solution

The character encoding of the command prompt window needs to match the character encoding of the FTP session. The character encoding of the FTP session will indicate to the FTP server the character set to use to encode the filename strings. This will ensure proper storage of the filenames in the repository as well as ensure proper display of filenames in the command prompt window.

Possible Solution for Microsoft Windows 95, 98, 2000, NT

Windows uses Unicode as its main character set. But for compatibility with MS-DOS, traditional code pages are supported. The MS-DOS command `chcp` allows viewing and setting the active code-page for the command prompt window.

To display the number of the active console code page, use a command line to enter the following:

```
C:\>chcp
Active code page: 437
```

437 is the United States code page. For example, if you are using 437, you will not be able to handle German characters, ö and ß, in your FTP session. To handle German characters, the code page 1252 should be used. 1252 is for Western European languages.

To change the active code page to 1252, use a command line to enter the following:

```
C:\>chcp 1252
Active code page: 1252
```

Only the Oracle Enterprise Manager (OEM) code page installed with Windows NT will display correctly in a command prompt window using Raster fonts. Other code pages will display correctly in windows using TrueType fonts or Lucida Console.

To change the fonts for the window:

1. Access the Properties / Font tab.
2. Start up your FTP session in this same command prompt window.

The character encoding for the session should be set using the quote command, `setcharencoding`, to the same encoding for the command prompt window. If you change the code page with the `chcp` command, it is recommended that you use the Lucida Console font as the console font to avoid incorrect glyphs on the screen.

Use the following code to set the character encoding for the session:

```
ftp> quote setcharencoding <IANA character set name>
```

For example:

```
ftp> quote setcharencoding Windows-1252
```

Caveats

The German Windows NT 4.0 Operating System has a default active code page of 850 for the command prompt windows. 850 is the multilingual Latin1 code page. The command prompt MS-DOS FTP client does not recognize German characters using 850, so the code page should be changed to 1252, which will work properly.

MS-DOS Code Pages

The following tables lists some MS-DOS code pages and their corresponding language.

MS-DOS Code Page	Language
437	United States
850	Multilingual (Latin I)
852	Slavic (Latin II)

MS-DOS Code Page	Language
855	Cyrillic (Russian)
857	Turkish
860	Portuguese
861	Icelandic
863	Canadian-French
865	Nordic
866	Russian
869	Modern Greek

Windows ANSI Code Pages

The following table lists Windows ANSI code pages and their language/font.

Windows ANSI Code Page	Language/Font
1250	Central European
1251	Eastern European using Cyrillic letters
1252	Western European using Roman letters
1253	Greek
1254	Turkish
1255	Hebrew
1256	Arabic
1257	Baltic
1258	Vietnamese

Possible Problem: Filenames Truncated or Corrupted

Filenames truncated or corrupted when using the drag and drop feature of the Web interface to upload files named with Western European characters using the Netscape browser.

Possible Solution

Netscape uses the Windows code page for its FTP client. If the Oracle database character set is UTF8, the FTP server needs to have the DefaultCharacterSet property in the properties file must be set to ISO-8859-1 for Western European languages.

Possible Problem: Unable to Drag and Drop Files Through the Web Interface

Unable to drag and drop files through the Web interface using multibyte operating system.

Possible Solution

The drag and drop feature only supports the character set (or subsets) of the Oracle *i*FS FTP server default character set. Check the value in the properties file.

Possible Problem: Unable to Insert Files Through FTP

Unable to insert files correctly through FTP even if the default character set of the FTP server is the same as the client OS.

Possible Solution

Be sure to have the NLS_LANG environment variable set to the same character set as the database character set. The OCI driver reads this environment variable and uses for string conversion before inserting into the database. For example, if the database character set is UTF8, then set NLS_LANG=.UTF8 This is only a problem with the OCI drivers, and not thin JDBC drivers.

Possible Problem: Unable to Perform Context Searches on Multilingual Content

Unable to context search multilingual content in a document after inserting a document into the repository through FTP.

Possible Solution

Be sure to set the language for the current FTP session. Documents inserted via the 'put' command will have its LANGUAGE attribute set to this value.

To view the current language for the FTP session, use the following quote command:

```
ftp> quote showlanguage
226 Language English
```

To set the language for the FTP session, use the following quote command:

```
ftp> quote setlanguage <Oracle language name>
```

See Also

For more information on NLS, see the *Oracle8i National Language Support Guide, Release 2 (8.1.6)*.

Common Administrative Errors

The following table describes some common administration problems, what may have caused them, and what you should do to correct the problem.

Problem	Probable Cause	Corrective Action
An out-of-memory exception is raised when running a protocol	The maximum Java heap size is too low.	Increase the heap size by modifying the `-mx` setting for that particular protocol in the `IfsProtocols.def` file in the `<ORACLE_HOME>\ifs<version>\settings` directory.

Problem	Probable Cause	Corrective Action
Server is generally slow for read and write activity	Server memory is overcommitted. The server is excessively swapping memory blocks to disk.	Run `vmstat` and look for excessive page swapping to verify the problem. Adjust the following parameters in your database's `init.ora` file: ■ Number of `processes` ■ Number of `open_cursors` ■ Reduce `db_block_buffers` Stop unneeded protocols. You may also need to add RAM to your server or, if you are running a single-tier configuration, reconfigure your Oracle *i*FS server into a two-tier configuration. For more information on adjusting the parameters in the init.ora file, see the *Oracle Internet File System Installation Guide*.
Server is slow only on read or search activity	Large amounts of documents have been loaded or updated and the Oracle8*i* Cost-Based Optimizer is using old statistics.	Run the `analyze.sql` script located in the `<ORACLE_HOME>\ifs<version>\admin\sql` directory. For more information, see the *Oracle Internet File System Release Notes*.
Server is slow only on content-based search activity	interMedia Text tablespaces are on the same disk as other database files.	Move the interMedia Text tablespaces to other disks. See the *Oracle Internet File System Installation Guide*. See the *Oracle8i Administrator's Guide* for more information on moving tablespaces.
	interMedia Text indexes have become fragmented.	Regularly optimize the interMedia Text Oracle index `INDEXEDLOB_I`. For more information, see the *Oracle8i interMedia Text Reference*.

Problem	Probable Cause	Corrective Action
Server is slow only on write activity	Large amounts of documents are being loaded and the Redo logs are too small.	Add two or more 100 MB or larger Redo logs. See the *Oracle8i Administrator's Guide* for more information.
	Large amounts of documents are being loaded and the Redo logs are on the same disk as the database files.	Place the Redo logs on a separate disk from the database files. See the *Oracle8i Administrator's Guide* and the *Oracle8i Designing and Tuning for Performance Guide* for more information.
		For optimal performance, dedicate one or more disks (and, if possible, a disk controller) exclusively to the Redo logs, and optimize the disks for sequential write activity. For example, on Sun SPARC Solaris, you may choose raw partitions or UNIX filesystems for the disks. If you choose UNIX filesystems, on Solaris 2.6, use the "forcedirectio" option when mounting the filesystems, and on Solaris 2.7 and above, use the "noatime" option when mounting the filesystems. These options should only be used if the filesystems are dedicated exclusively to the Redo logs.
Content queries through the Web and Windows return no rows.	interMedia Text indexing of the documents has not occurred.	Run `ctxsrv` as described in the Starting ifs section of the Installation Guide. (Replace with correct cross-ref).

On Windows NT, the NTFS server crashes regularly. The error log contains the following:
`no ifsproxy in shared library path`
`Cache performance for session cache`
`CurrentSession Count=0`
`HighestSession Count=0`
`CurrentSession Cache=0`
	ORACLE_HOME is not set correctly.	Make sure the `ifsproxy.dll` file is located in `%ORACLE_HOME%\ifs<version>\bin`.

Problem	Probable Cause	Corrective Action
When transferring old passwords to new schema, users do not have access to WebDAV.	Because old passwords were transferred to a new schema, existing Oracle *i*FS users are unable to authenticate using WebDAV. This only applies to users created before the schema upgrade.	Users must reset their Oracle *i*FS password, even if they keep the same password they had before the schema upgrade.
IfsProtocols server process crashes regularly or after all Oracle *i*FS processes are started, the IfsProtocols server dies.	After examining the IfsProtocols log, it is possible that two IfsProtocols processes were started. This can happen when the first IfsProtocols process was not shut down properly. In Oracle *i*FS, each IfsProtocols process is unique by identifier which is permitted by Oracle *i*FS, but confusing to the user.	Either start the IfsProtocols with unique names, or refer to them by their identifier. For more information, see Chapter 8, "Using Server Manager to Start and Stop Servers".
The following error message displays when trying to login to Oracle *i*FS: `IFS-10620: Unable to construct connection pool.`	This error indicates that Oracle *i*FS could not connect to the database. The possible cause was that the JDBC driver is missing from the `CLASSPATH`.	Check the JDBC driver is in the `CLASSPATH`. Access `$ORACLE_HOME/jdbc/lib/classes111.zip` to check this. Also make sure the parameter for the `DatabaseUrl` field is as follows: `DatabaseUrl=jdbc:oracle: oci8:@<tnsnames entry>`.
If a new folder is created using the Windows interface, an agent is triggered.	When a folder is created in Windows Explorer, a new file is created (testdir.tmp) and then is deleted. Windows Explorer does this to make sure the right information is available to create information inside the folder. Therefore, the agent is triggered.	No corrective action, this is the way Windows Explorer is designed to work.
Want to change the Oracle *i*FS system password.	Have forgotten the Oracle *i*FS system password.	You will need to contact Oracle Support or re-install the database. If you choose to re-install the database, you will need to make sure that you have copied any files stored in the Oracle *i*FS database out to a local hard drive before starting the re-install process. The re-install will result in the loss of all information currently stored in the database.

A

Command Line Utilities Reference

This appendix provides instructions for running the Command Line Utilities and a complete listing of the commands. The Command Line Utilities allows experienced system administrators to perform tasks from the command line prompt rather than the graphical user interface.

If an argument to the Command Line Utilities has spaces or an "=", it must be enclosed within double quotes (""). For example, `ifscat "test document"` or `ifsls "backup folder 1999"`or `ifsls -class Document "name='Mydoc'"`.

Topics include:

- Command Line Utilities: An Overview
- Running the Command Line Utilities
- Starting the Command Line Utilities
- Navigation Commands
- File Manipulation Commands
- Miscellaneous Commands
- National Language Support Commands

Command Line Utilities: An Overview

The Command Line Utilities are automatically installed into `$ORACLE_HOME/ifs<version>/bin` on the server machine where Oracle *i*FS is installed. The Command Line Utilities can also be installed on any UNIX or Windows NT client machine.

> **Note:** To use the Command Line Utilities, the CUP server must be running.

Setting Environment Variables to Run the Command Line Utilities

The Command Line Utilities use three environment variables. Two of these variables are set at install time in the `ifscmdenv` file located in the following directory:

Platform	Directory
UNIX	`$ORACLE_HOME/ifs<version>/bin`
Windows NT	`%ORACLE_HOME%\ifs<version>\bin`

You can modify this script to change any of the install defaults.

When the Command Line Utilities are installed on a client machine, the variables in this script (which is `ifscmdenv.bat` if installed on a Windows machine) will need to be modified since the utilities are no longer running on the same machine as the Command Line Utility Protocol Server.

You can also set these environment variables directly on the command line itself, or in your local environment script, for example, the `.cshenv` file if you use `csh`.

Variables

`IFS_COOKIE_DIRECTORY`—The session information for the Command Line Utilities is maintained in the `IfsCookie` file, which is generated each time a user logs in. This variable specifies where this file should be placed on the local system. By default, this directory is set to `$ORACLE_HOME/ifs<version>/log`. This default should be changed if either of the following cases is true:

- Multiple Users—If more than one user will be using the Command Line Utilities on the same machine. If two different users have the `IFS_COOKIE_DIRECTORY` set to the same local directory, and are logged in at the same time, the `IfsCookie` file will be overwritten by the most recent user's credential information. This means that both users will have the most recent user's permissions, and any action either user takes will affect both users.
- Security—When an `IfsCookie` file exists, which is only during the life of a user's logon, anyone who has access to the directory containing the `IfsCookie` file could potentially use that cookie to logon as the already logged on user. If security is a concern, it is highly recommended that the `IFS_COOKIE_DIRECTORY` be changed from the default, since `$ORACLE_HOME/ifs<version>/log` is a commonly known directory. CUP commands will not work without setting the `IFS_COOKIE_DIRECTORY`.

`IFS_CUP_SERVER`—If this variable is not set, it defaults to `localhost`. This default will only work if the user is using the Command Line Utilities against a Command Line Utility Protocol Server (CUP) on the same machine. To access a different instance of the server other than `localhost`, set this variable to the appropriate server, for example, `ifstestmachine`. This variable will always need to be set when running the Command Line Utilities on a client machine since the Command Line Utility Protocol Server (CUP) will not be locally available.

`IFS_CUP_PORT`—All Oracle *i*FS commands access the CUP using the default port, 4180. If the CUP server is started on a port other than 4180, this variable should be set to that port number. For example, if you start the CUP on 4182, this variable should be set to 4182.

See Also

For more information on installation of the Command Line Utilities, see the *Oracle Internet File System Installation Guide*.

Running the Command Line Utilities

There are four different ways to run the Command Line Utilities:

1. On the Command Line.

   Depending on your operating system, there are .bat or .sh (Windows NT and Unix, respectively) that run individual Command Line Utilities commands. These files are located in the following directory:

Platform	Directory
UNIX	`$ORACLE_HOME/ifs<version>/bin`
Windows NT	`%ORACLE_HOME%\ifs<version>\bin`

   For example, `ifslogin.bat` is used on Windows NT as `ifslogin tuser1/tuser1`.

2. In an interactive mode.

   To use the commands interactively, you must use the script. On the command line, type `ifsshell`. When using the shell, you do not need to enter the entire command. For example, if you want to log in, type `login <name>/<password>`.

3. In a batch mode.

   Using a batch mode, you have the ability to perform many actions at once. To start the Command Line Utilities in batch mode, at the command prompt, type `ifsshell -i <filename>`. This tells the operating system to execute the batch file. The file should contain a list of commands that you use in `ifsshell`.

4. In scripts.

   Use any commands in a script, just as you would from the operating system command line. For example, a .bat or .sh file.

In most cases, you will use the Command Line Utilities in an interactive mode by using your operating system. The following sections describe the procedures for running the Command Line Utilities. To use the Command Line Utilities, the Command Line Utilities Protocol (CUP) server needs to be started. You can do this using Server Manager.

Starting the Command Line Utilities

Use Server Manager to start the Command Line Utilities protocol server. From your operating system, you can log on by using the `ifslogin` command.

Navigation Commands

The following commands are used for navigation throughout Oracle *i*FS.

ifscd

Changes the current working directory to the directory specified. If no working directory is specified, this command changes the current working directory to the home directory of the user in Oracle *i*FS. The default home directory is `/home/<username>`.

Syntax:

```
ifscd <directory>
```

Examples:

`ifscd /osh` or `ifscd ./osh`
Changes the current working directory to `/osh`.

`ifscd`
Changes the current working directory to the home directory of the user in Oracle *i*FS.

`ifscd ..`
Changes the current working directory one level up from the current directory.

ifspwd

Displays the current working directory in Oracle *i*FS.

Syntax:

```
ifspwd
```

Example:

`ifspwd /home/scott`
Displays the current directory, `/home/scott`.

File Manipulation Commands

ifscp

Copies the source file to the specified destination. If the destination is an existing directory, the file is copied into that directory with the same name. If the destination is a filename, the source file is copied to that file. If the destination file already exists, the copy defaults to versioning semantics.

Syntax:

```
ifscp <source> <destination>
```

Examples:

```
ifscp specifications.doc /osh/specifications2.doc
```

Copies `specifications`.doc to `/osh/specifications2.doc`, only if `/osh` exists.

```
ifscp specifications.doc /osh
```

Copies `specifications.doc` to `/osh`.

ifsget

Renders the file to the specified local directory.

Syntax:

```
ifsget <iFSfile> <localfile>
```

Examples:

```
ifsget specifications.txt
ifsget specifications.txt /osh/specifications.txt
```

Picks up the file, `specifications`.txt, and copies it to the local directory, `/osh/specifications.txt`.

ifsgetattr

Display the attributes of an object. By default, all attributes are displayed.

Syntax:

```
ifsgetattr [<object options>] <objectvalue> [<listing options>]
```

Options

Object Options:

`-path` This is the default option but can be set explicitly. The object value will be evaluated as a path to an object.

`-id` The object value will be evaluated as an ID of an object.

`-systemacl` The object value will be evaluated as the name of a SystemAccessControlList.

`-user` The object value will be evaluated as the name of a DirectoryUser.

`-valuedefault` The object value will be evaluated as the name of a ValueDefault.

`-class <classname>` Identifies objects based on the classname and the object value. The object value will be evaluated as a where clause.

Listing Options:

`-l` In addition to the name of each object, this option displays the class name of the object, the ACLs, owner name, size in bytes, and timestamp (the modification time unless other times are selected).

`-attr <attribute list>` Displays the attributes specified for the identified object.

`-attrall` Displays all attributes for the identified object.

Examples:

```
ifsgetattr signature.txt
```

Displays all attributes for `signature.txt`.

```
ifsgetattr signature.txt -attr owner acl creationDate
```

Displays the owner, ACL and creation date for `signature.txt`.

```
ifsgetattr -id 1234 -attrall
```

Displays all attributes for the object identified by ID 1234.

ifsmkdir

Creates a directory with the specified name.

Syntax:

```
ifsmkdir [<options>] <directory>
```

Options:

`-p` Creates the complete directory path.

Examples:

```
ifsmkdir testDir
```

Creates `testDir` in the current directory.

```
ifsmkdir -p a/b/testDir
```

Creates a/b/testDir, creating each directory in the path as needed.

ifssetattr

Sets the attribute value on an object. Given an object and an attribute name, the datatype can be determined. For simple attributes types (string, double, integer, boolean, long, and date) the `attrvalue` is interpreted. For attribute types that refer to objects, the `attrvalue` is interpreted as a path unless `attr` options are used.

Syntax:

```
ifssetattr [<object options>] <objectvalue> <attrname> [<attr options>]
[<attrvalue>]
```

Options:

Object Options:

`-path` This is the default option but can be set explicitly. The object value will be evaluated as a path to an object.

`-id` The object value will be evaluated as an ID of an object.

`-systemacl` The object value will be evaluated as the name of a SystemAccessControlList.

`-user` The object value will be evaluated as the name of a DirectoryUser.

`-valuedefault` The object value will be evaluated as the name of a ValueDefault.

`-class <classname>`	Identifies objects based on the classname and the object value. The object value will be evaluated as a where clause.

Attribute Options:

-avpath	This is the default option, but it can be set explicitly. The attribute value will be evaluated as a path to an object.
-avid	The attribute value will be evaluated as an id of an object.
-avsystemacl	The attribute value will be evaluated as the name of a SystemAccessControlList.
-avuser	The attribute value will be evaluated as the name of a Directory User.
-avvaluedefault	The attribute value will be evaluated as the name of a ValueDefault.
-avclass <classname>	Identifies a unique object based on the classname and the attribute value. The attribute value will be evaluated as a where clause. If more than one object is identified, an error message is generated.
-avnull	Sets an attribute to null.

Examples:

```
ifssetattr -id 1234 owner -avuser scott
```

Changes the owner attribute of the object identified by ID 1234 to be scott.

```
ifssetattr -user scott description "System Administrator"
```

Changes the description attribute of the `DirectoryUser` object identified by scott.

ifsshowallpaths

Displays all folder references to the specified object. The specified object must be a Public Object.

Syntax:

```
ifsshowallpaths [<object options>] <objectvalue>
```

Options:

`-path` This is the default option but can be set explicitly. The object value will be evaluated as a path to an object.

`-id` The object value will be evaluated as an id of an object.

`-systemacl` The object value will be evaluated as the name of a SystemAccessControlList.

`-user` The object value will be evaluated as the name of a DirectoryUser.

`-valuedefault` The object value will be evaluated as the name of a ValueDefault.

`-class <classname>` Identifies objects based on the classname and the object value. The object value will be evaluated as a where clause.

Examples:

```
ifsshowallpaths signature.txt
```

Displays all folder references to `signature.txt`.

```
ifsshowallpaths -id 1234
```

Displays all folder references to the object identified with the id 1234.

ifsls

Lists the items as specified. If no arguments are specified, the contents of the current directory are listed alphabetically. If the object identified is a folder, the contents of the folder are listed. Wildcard strings are not supported.

Syntax:

```
ifsls [<options>] [<objectvalue>] [<listing options>]
```

Options:

Object Options:

`-path` This is the default option but can be set explicitly. The object value will be evaluated as a path to an object.

`-id` The object value will be evaluated as an ID of an object.

`-systemacl` The object value will be evaluated as the name of a SystemAccessControlList.

`-user` The object value will be evaluated as the name of a DirectoryUser.

`-valuedefault` The object value will be evaluated as the name of a ValueDefault.

`-class <classname>` Identifies objects based on the classname and the object value. The object value will be evaluated as a where clause.

Listing Options:

`-l` In addition to the name of the object, this option displays the file type, ACLs, owner name, size in bytes, and modification date. Last line is the total number of objects.

`-attr` This option displays the attributes specified by the identified object.

`-attrall` Displays all attributes for the identified object.

Examples:

```
ifsls
```

Lists the names of the items in the current directory.

```
ifsls osh
```

Lists the names of the items in the `osh` directory.

```
ifsls -id 1234 -attrall
```

Lists all the attribute values of the object identified by the ID, 1234. If the object identified is a folder, all the attributes for the contents of the folder are listed.

```
ifs -class DOCUMENT -attr name description
```

Lists the names and descriptions of all the documents in Oracle *i*FS.

ifsmv

Moves the source file to the destination. If the destination is an existing directory, the source file is moved to the destination directory. If the destination is a file, the source file is renamed to the destination filename. Wildcards are not supported.

Syntax:

```
ifsmv <source> <destintaion>
```

Examples:

```
ifsmv sample.html sample2.html
```

Renames `sample.html` to the `sample2.html`.

```
ifsmv specifications.txt testdir/specifications.txt
```

Moves `specifications.txt` into `testdir`.

```
ifsmv groups.fm documention/groups2.fm
```

Moves `groups.fm` into `documentation` and renames the file to `groups2.fm`.

ifsput

Stores the specified local file in Oracle *i*FS. If `iFSfile` is not specified; the file is stored in the current Oracle *i*FS directory. If `iFSfile` exists, the file is updated according to the versioning semantics.

Syntax:

```
ifsput <localfile> [<iFSfile>]
```

Examples:

```
ifsput sample.html
```

Stores the local file in the current Oracle *i*FS directory as `sample.html`.

```
ifsput sample.html sample2.html
```

Updates `sample2.html` with the local file, `sample.html` following versioning semantics.

ifsrm

Removes an object from Oracle *i*FS. If the object is identified by path and the object is not in any other directory, then it is actually deleted from Oracle *i*FS. Otherwise, the object is only deleted from the current directory. If the object value is a directory, by default, the directory must be empty unless the -r (recursive) flag is specified.

Syntax:

```
ifsrm [<options>] [<object options>] <objectvalue>
```

Options:

-r This option causes the recursive removal of a directory and all its contents without prompting.

Object Options:

`-path`	This is the default option, but can be set explicitly. The object value will be evaluated as a path to an object.
`-id`	The object value will be evaluated as an ID of an object.
`-systemacl`	The object value will be evaluated as the name of a SystemAccessControlList.
`-user`	The object value will be evaluated as the name of a DirectoryUser.
`-valuedefault`	The object value will be evaluated as the name of a ValueDefault.
`-class <classname>`	Identifies objects based on the classname and the object value. The object value will be evaluated as a where clause.

Examples:

```
ifsrm sample.html
```

Removes `sample.html` from the current directory.

```
ifsrm -r a/b/myDir
```

Removes `myDir` from `b` and any items in `myDir`.

```
ifsrm -id 1234
```

Deletes the object identified by the ID, 1234.

ifsrmref

Removes a reference from a directory. If the source is not in any other directory, then it is actually deleted from Oracle *i*FS. Otherwise, the source is only deleted from the current directory. If the source is a directory, by default, the directory must be empty unless the `-r` (recursive) flag is specified.

Syntax:

```
ifsrmref [-r] <source>
```

Options:

`-r`	This option removes a directory and all of its contents from the current directory without prompting.

Examples:

`ifsrmref signature.txt`
Removes `signature.txt` from the current directory.

`ifsrmref -r testDir`
Removes `testDir` and all of its contents from the current directory.

ifscat

Prints an ASCII rendering of the identified object to the screen. For Document and its subclasses, the content of the document will be rendered by default. Folders and its subclasses are not rendered. All other objects are rendered as XML by default. For information regarding changing the default rendering behavior, refer to Oracle *i*FS documentation.

Syntax:

`ifscat [object options] [object value]` or `ifstype`

Options:

Object Options:

`-path`	This is the default option, but can be set explicitly. The object value will be evaluated as a path to an object.
`-id`	The object value will be evaluated as an ID of an object.
`-systemacl`	The object value will be evaluated as the name of a SystemAccessControlList.
`-user`	The object value will be evaluated as the name of a Directory User.
`-valuedefault`	The object value will be evaluated as the name of a ValueDefault.
`-class <classname>`	Identifies objects based on the classname and the object value. The object value will be evaluated as a where clause.

Examples:

```
ifscat myFile
```

Displays the contents of `myFile`.

```
ifscat -id 1234
```

Displays the rendering of object with ID 1234.

ifsaddref

Adds an item to a directory. The result will be that the source will show up in the destination directory. This does not affect any other directories in which the source may already be in.

Syntax:

```
ifsaddref <filename> <pathDir>
```

Examples:

```
ifsaddref signature.txt a/oshDir
```

Adds `signature.txt` to `a/oshDir`.

```
ifsaddref oshDir a/oshDir2
```

Adds `oshDir` to `a/oshDir2`.

ifsfind

Returns the objects based on the classname and an optional where clause. By default, only the name of the attribute is displayed.

Syntax:

```
ifsfind classname [<whereclause>] [<listing option>]
```

Options:

Listing Options:

`-l` In addition to the name of each file, this option displays the file type, ACLs, owner name, size in bytes, and timestamp (the modification time unless other times are selected).

`-attr <attribute list>` Displays the attributes specified for the identified object.

`-attrall` Displays all attributes for the identified object.

Examples:

```
ifsfind FOLDER -attrall
```

Displays all the attributes for all the folders in Oracle *i*FS.

```
ifsfind DOCUMENT "description='FY 2000'" -attrall
```

Displays all the attributes for the folders with the description `"FY 2000"`.

Miscellaneous Commands

ifshelp

Displays help on all commands or on the specified command.

Syntax:

```
ifshelp [cmd]
```

Examples:

```
ifshelp
```

Displays usage for all commands.

```
ifshelp cd
```

Displays the description and command usage for `cd`.

Note: The command is `cd` not `ifscd`.

ifslogin/ifslogon

Establishes an Oracle *i*FS session.

Syntax:

```
ifslogin <username/password> or ifslogon <username>/<password>
```

Example:

```
ifslogin scott/tiger
```

Logs scott in to Oracle *i*FS.

ifslogout

Terminates the current Oracle *i*FS session.

Syntax:

```
ifslogout
```

ifsmode

Sets the modes for the session.

Syntax:

```
ifsmode <modetype> [<object options>] <object value>
```

Options:

Mode Types:

`admin`	Sets the administration mode (on/off). The object value is a Boolean value.
`acl`	When creating new objects, sets the ACL for new objects to the ACL specified. The ACL can also be specified with the `-parent` or the `-profile` option.
`dateformat`	Sets the format for dates. The object value is a date format.
`parse`	Sets the parse mode (on/off). The object value is a boolean value.

Object Options:

`-path`	This is the default option but can be set explicitly. The object value will be evaluated as a path to an object.
`-id`	The object value will be evaluated as an ID of an object.
`-systemacl`	The object value will be evaluated as the name of a SystemAccessControlList.
`-user`	The object value will be evaluated as the name of a DirectoryUser.
`-profile`	Sets the ACL mode to take the ACL from the user profile.
`-parent`	Sets the ACL mode to take the ACL from the parent directory.

Boolean Values:

The following Boolean values are valid:

- True—`true`, `t`, `on`, `1`, `y`, and `yes` are legal true values.
- False—`f`, `off`, `0`, `n`, and `no` are legal false values.

Examples:

```
ifsmode admin T
```

Sets administration mode on.

```
ifsmode admin off
```

Sets administration mode off.

```
ifsmode -name Private
```

Sets the default ACL to the Private ACL.

```
ifsmode acl -id 1234
```

Sets the default ACL to be an ACL identified by ID 1234.

```
ifsmode acl -profile
```

Sets the default ACL to be taken from the user profile.

```
ifsmode acl -parent
```

Sets the ACL determination to be taken from the parent folder.

```
ifsmode parse off
```

Sets the parse mode to off, so that files put in Oracle *i*FS are not parsed.

ifssu

Allows user to impersonate another user without logging off. The command with a username impersonates that user. The command without arguments reverts back to the original user. Only users with administration permissions can use this command.

Syntax:

```
ifssu [<username>]
```

Examples:

```
ifsu jsmith
```

Impersonates the user jsmith.

```
ifsu
```

Cancels the impersonation command.

ifswhoami

Displays the current user.

Syntax:

```
ifswhoami
```

Example:

```
ifswhoami
```

If a user has logged in, that user is displayed.

ifsascii

Changes to ascii mode.

Syntax:

```
ifsascii
```

ifsbin

Changes to binary mode.

Syntax:

```
ifsbin
```

ifsversion

Displays the Oracle *i*FS version information.

Syntax:

```
ifsversion
```

ifsshell

Starts the Command Line Utilities shell. If no arguments are specified, an interactive shell is started.

Syntax:

```
ifsshell [-i <filename>]
```

Options:

`-i`	Indicates a file containing Command Line Utilities commands. This will run all the commands in a single JVM.

Examples:

```
ifsshell
```

Starts up an interactive shell that takes in Command Line Utilities commands.

```
ifsshell -i batchcommands.txt
```

Reads in the commands from the input file, `batch commands.txt`, and runs them.

National Language Support Commands

ifsshowchar

Shows the character set for this session.

Syntax

```
ifsshowchar
```

Example:

```
ifsshowchar
```

Character Set: UTF-8

ifssetchar

Sets the character set for this session. Documents inserted using this CUP session will have its attributes set to this character set.

Syntax

```
ifssetchar [IANA charset name]
```

Example:

```
ifssetchar UTF-8
```

Sets to UTF-8 for this session.

ifsshowlang

Shows the language for this session.

Syntax

```
ifsshowlang
```

Example:

```
ifsshowlang
```

Displays the language for this CUP session.

ifssetlang

Sets the language for this session. Documents inserted using this CUP session will be indexed in this language.

Syntax

```
ifssetlang
```

Example:

```
ifssetlang French
```

Sets the language for this CUP session to French.

B

Secondary Properties

This appendix provides information on the secondary properties in the Oracle *i*FS service properties files. For information regarding the properties you are required to set, refer to Chapter 1, "Getting Started with Oracle iFS". Topics include:

- Oracle iFS Properties Files: An Overview
- Database Properties
- Directory Service Properties
- Read Only Connection Pool Properties
- Service Data Cache Properties
- Service Properties
- Session Properties
- Tracing Properties
- Writeable Connection Pool Properties
- National Language Support Properties

Oracle *i*FS Properties Files: An Overview

This file provides properties you must set and some secondary properties you can modify if you so choose. Oracle Corporation recommends that you retain the default settings for the secondary properties. Use any text editor to set the properties in the Properties file. There are eight types of secondary properties, each described in a separate table.

- Database
- Service
- Service Data Cache
- Writeable Connection Pool
- Read Only Connection Pool
- Directory Service
- Session
- Tracing

Database Properties

Property	Specifies	Example
`JdbcTracing`	Whether JDBC tracing is enabled; must be either true or false. If true, JDBC trace information is sent to the standard output stream of the Oracle *i*FS server. Optional, defaults to false.	`JdbcTracing=`*`false`*
`DefaultRowPrefetch`	The default number of result set rows prefetched from the database by the JDBC driver. Optional, if unspecified or if set to 0, a value of 10 rows is used.	`DefaultRowPrefetch=`*`0`*
`AuroraService`	Reserved. Must be false.	`AuroraService=`*`false`*
`Driver`	The name of the JDBC driver.	`Driver=`*`oracle.jdbc.driver.OracleDriver`*
`User`	The Oracle *i*FS database schema. The default is `ifssys`. A different database schema can be created, in which case the user name would be the name created.	`User=`*`ifssys`*
`DatabaseUrl`	The JDBC database URL.	`DatabaseUrl=`*`jdbc:oracle:oci8:@`*

Directory Service Properties

Property	Specifies	Example
CaseSensitiveAuthentication	Whether user names and passwords are case sensitive.	CaseSensitiveAuthentication=*false*
CredentialManagers	The primary names of the credential managers for this service.	CredentialManagers=*Ifs, Custom*
CredentialManager<primaryname>AcceptTokenCredential	Specifies whether the <primaryname> credential manager accepts token credentials.	CredentialManager*Ifs*AcceptTokenCredential=*true*
CredentialManager<primaryname>RdbmsUserMustExit	Specifies whether the <primaryname> credential manager requires an Oracle *i*FS user to also be a database user in order to authenticate.	CredentialManager*Ifs*RdbmsUserMustExit=*false*
CredentialManager<*primaryname*>Classname	The Java class name for the <*primaryname*> credential manager.	CredentialManager*Ifs*Classname= *oracle.ifs.server.IfsCredentialManager*
CredentialManager<*primaryname*>AlternateNames	The alternate names for the <*primaryname*> credential manager.	CredentialManager*Ifs*AlternateNames=*IFS*, Default
CredentialManager<*primaryname*>AcceptCleartextCredential	Whether the <*primaryname*> credential manager accepts clear text credentials.	CredentialManager*Ifs*AcceptCleartextCredential=*true*
CredentialManager<*primaryname*>AcceptChallengeResponseCredential	Whether the <*primaryname*> credential manager accepts challenge/response credentials.	CredentialManager*Ifs*AcceptChallengeResponseCredential=*true*
CredentialManager<*primaryname*>AcceptHHPDigestCredential	Whether the <*primaryname*> credential manager accepts HTTP Digest credentials (used by WebDAV); optional, defaults to false.	CredentialManagerIfsAcceptHttpDigestCredential=*true*
CredentialManager<*primaryname*>Schema	The name of the credential manager schema; required.	CredentialManagerIfsSchemaSchema= *ifssys$cm*

Read Only Connection Pool Properties

Property	Specifies	Example
ReadonlyConnectionPoolMinimumSize	The initial number of connections in the pool. Optional, defaults to 2.	ReadonlyConnectionPoolMinimumSize=*2*
ReadonlyConnectionPoolTargetSize	The target maximum number of connections in the pool. Must be greater than or equal to ReadonlyConnectionPoolMinimumSize. Optional, defaults to 5.	ReadonlyConnectionPoolTargetSize=*5*
ReadonlyConnectionPoolMaximumSize	The absolute maximum number of connections in the pool. Must be greater than or equal to ReadonlyConnectionPoolTargetSize. Optional, defaults to 10.	ReadonlyConnectionPoolMaximumSize=*10*
ReadonlyConnectionPoolTargetSizeTimeout	The maximum period, in milliseconds, that the Oracle *i*FS server will postpone a session's request for a connection if there are no unallocated connections in the pool and the current pool size is less than the target pool size. Optional, defaults to 1000.	ReadonlyConnectionPoolTargetSizeTimeout=*1000*
ReadonlyConnectionPoolMaximumSizeTimeout	The maximum period, in milliseconds, that the Oracle *i*FS server will postpone a session's request for a connection if there are no unallocated connections in the pool and the current pool size equals or exceeds the target pool size. Optional, defaults to 10000.	ReadonlyConnectionPoolMaximumSizeTimeout=*10000*

Service Data Cache Properties

Property	Specifies	Example
`ServiceDataCacheSize`	The size of the service data cache. This value specifies the initial size of the Service Data cache, which caches the data for library objects. Optional, defaults to 7500.	`ServiceDataCacheSize=`*`7500`*
`ServiceDataCacheNormalTrigger`	The cache size at which the service data cache schedules a low-priority purge of data that has not been recently used. Should be less than `ServiceDataCacheSize.` Optional, defaults to 5000.	`ServiceDataCacheNormalTrigger=`*`5000`*
`ServiceDataCacheUrgentTrigger`	The cache size at which the service data cache schedules a high-priority purge of data that has not been recently used. Should be greater than `ServiceDataCacheNormalTrigger` but not exceed `ServiceDataCacheSize`. Optional, defaults to 5500.	`ServiceDataCacheUrgentTrigger=`*`5500`*
`ServiceDataCacheEmergencyTrigger`	The cache size at which the service data cache begins an immediate purge of data that has not been recently used. This value sets an absolute maximum cache size and should be greater than `ServiceDataCacheUrgentTrigger`. Optional, defaults to 6000.	`ServiceDataCacheEmergencyTrigger=`*`6000`*
`ServiceDataCachePurgeTarget`	The target cache size upon completion of a cache purge. Should be less than `ServiceDataCacheNormalTrigger`. Optional, defaults to 4000.	`ServiceDataCachePurgeTarget=`*`4000`*

Service Properties

Property	Specifies	Example
`SessionTimeoutPeriod`	The period of inactivity, in seconds, after which a session is considered idle. Optional, defaults to 600 seconds.	`SessionTimeoutPeriod=`*600*
`CheckForIdleSessionsPeriod`	The period, in seconds, between successive checks by the Oracle *i*FS server for idle sessions. A value of 0 disables checking for idle sessions. Optional; defaults to 60 seconds.	`CheckForIdleSessionsPeriod=`*60*
`PollForEventsFromOtherServicesPeriod`	The period, in seconds, between successive checks by an Oracle *i*FS server process for events from other Oracle *i*FS server processes. Specifying 0 disables inter-process event processing. Optional, defaults to 2 seconds.	`PollForEventsFromOtherServicesPeriod=`*2*
`TransportEventsToOtherServicesPeriod`	The maximum period, in seconds, that outgoing events are buffered by the originating Oracle *i*FS server process before being made available to other Oracle *i*FS server processes. Specifying 0 disables outgoing event buffering. Optional, defaults to 2 seconds.	`TransportEventsToOtherServicesPeriod=`*2*
`ServiceKeepAliveEventPeriod`	The period, in seconds, between succesive service keep-alive events for a `S_LibraryService`. Defaults to 60 seconds; if 0, service keep-alive events are disabled.	`ServiceKeepAliveEventPeriod=`*60*

Session Properties

Property	Specifies	Example
`SessionObjectCacheSize`	Target size of the session object cache in the client process. Optional; defaults to 20000.	`SessionObjectCacheSize=`*`20000`*
`SSessionObjectCacheSize`	Target size of the session object cache in the server process. Optional; defaults to 20000.	`SSessionObjectCacheSize=`*`20000`*
`TransactionStackSize`	Maximum number of concurrent nested transactions per session. Optional, defaults to 100	`TransactionStackSize=`*`100`*
`EventPollerPeriod`	Period, in milliseconds, between successive event heartbeats. Optional, defaults to 2500 milliseconds.	`EventPollerPeriod=`*`2500`*
`FolderPathCacheEnabled`	Indicates whether Oracle *i*FS caches the resolution of folder path strings. Optional, defaults to `true`.	`FolderPathCacheEnabled=true`
`FolderPathCacheSize`	How many resolved folder paths are cached when folder path caching is enabled. Should be 150% of `FolderPathCacheNormalTrigger.` Optional, defaults to `150`.	`FolderPathCacheSize=150`
`FolderPathCacheNormalTrigger`	The folder path cache size at which a low-priority purge is scheduled. Optional, defaults to `100`.	`FolderPathCacheNormalTrigger=100`
`FolderPathCacheUrgentTrigger`	The folder path cache size at which a high-priority purge is scheduled. Should be 110% of `FolderPathCacheNormalTrigger.` Optional, defaults to `110`.	`FolderPathCacheUrgentTrigger=110`

Property	Specifies	Example
FolderPathCachePurgeTarget	Target folder path cache size upon completion of a purge. Should be 80% of FolderPathCacheNormalTrigger. Optional, defaults to 80.	FolderPathCachePurgeTarget=80

Tracing Properties

Property	Specifies	Example
TraceLoggerChannelCount	Number of trace logger channels. Oracle *i*FS reserves channels 0 to TraceLogger.LAST_RESERVED_CHANNEL for internal use. Optional, defaults to 20.	TraceLoggerChannelCount=*20*
SSessionTraceLoggerTraceType	The destination of trace data generated by a session in the client process. Optional, defaults to TRACETYPE_NONE.	SSessionTraceLoggerTraceType=*TRACETYPE_NONE*
SSessionTraceLoggerTraceType	The destination of trace data generated by the session in the server process. Optional, defaults to TRACETYPE_NONE.	SSessionTraceLoggerTraceType=*TRACETYPE_LOCAL*
ServiceTraceLoggerTraceType	The destination of trace data generated by the Oracle *i*FS server. Optional, defaults to TRACETYPE_NONE.	ServiceTraceLoggerTraceType=*TRACETYPE_BOTH*
TraceLoggerLevelChannel*n*	The tracing verbosity for trace channel. Optional, no defaults.	TraceLoggerLevelChannel*10*=*5* to enable detailed tracing for method invocation TraceLoggerLevelChannel*6*=*0* to disable tracing for events
TraceLoggerLevelAllChannels	The tracing verbosity for all trace channels. Optional, no defaults.	TraceLoggerLevelAllChannels=*10*

Writeable Connection Pool Properties

Property	Specifies	Example
WriteableConnectionPoolMinimumSize	The initial number of connections in the pool. Optional, defaults to 2.	WriteableConnectionPoolMinimumSize=*2*
WriteableConnectionPoolTargetSize	The target maximum number of connections in the pool. Must be greater than or equal to WriteableConnectionPoolMinimumSize. Optional, defaults to 5.	WriteableConnectionPoolTargetSize=*5*
WriteableConnectionPoolMaximumSize	The absolute maximum number of connections in the pool. Must be greater than or equal to WriteableConnectionPoolTargetSize. Optional, defaults to 10.	WriteableConnectionPoolMaximumSize=*10*
WriteableConnectionPoolTargetSizeTimeout	The maximum period, in milliseconds, that the Oracle *i*FS server will postpone a session's request for a connection if there are no unallocated connections in the pool and the current pool size is less than the target pool size. Optional, defaults to 1000.	WriteableConnectionPoolTargetSizeTimeout =*1000*
WriteableConnectionPoolMaximumSizeTimeout	The maximum period, in milliseconds, that the Oracle *i*FS server will postpone a session's request for a connection if there are no unallocated connections in the pool and the current pool size equals or exceeds the target pool size. Optional, defaults to 10000.	WriteableConnectionPoolMaximumSizeTimeout =*10000*

National Language Support Properties

Property	Specifies	Example
`DefaultCharacterSet`	The character set can be set to any valid IANA character set name. The LibrarySession's character set will be obtained from this parameter as a default. If this parameter is not set, the database character set will be used. The LibrarySession's character set can be explicitly set through the API, FTP, CUP, and the Web. FTP uses the character set to determine the character encoding to use for the client connections. The repository uses the character set as a default for the CharacterSet attribute of documents inserted into the repository with a null attribute value.	`DefaultCharacterSet =` *`UTF-8`*
`DefaultLanguage`	The language can be set in any valid Oracle language name. The LibrarySession's language will be obtained from this parameter as a default. If the language cannot be guessed from the character set, the repository uses the language as a secondary default for the Language attribute of documents inserted into the repository with a null attribute value.	`DefaultLanguage=`*`English`*

The NLS parameters in the following table are involved in creating a Java Locale object. The LibrarySession contains a default Locale object. The values for this object are obtained from these 3 parameters in the properties file. If these are not set, the default Locale from the JVM is used.

Property	Specifies	Example
`DefaultLocaleLanguage`	The first argument to a Java Locale constructor is a valid ISO Language Code. These codes are the lower-case two-letter codes as defined by ISO-639.	`DefaultLocaleLanguage=`*en*
`DefaultLocaleCountry`	The second argument to a Java Locale constructor is a valid ISO Country Code. These codes are the upper-case two-letter codes as defined by ISO-3166.	`DefaultLocaleCountry=`*US*
`DefaultLocaleVariant`	The Variant codes are vendor and browser specific. It is the third argument when constructing a Java Locale object. This is most often left blank.	`DefaultLocaleVariant =` *`WIN`*

Index