

Oracle Application Server Portal

Technical Note

USING WEBDAV CLIENTS TO REPLICATE EXTERNAL CONTENT INTO ORACLEAS PORTAL

August 2004

INTRODUCTION

In any enterprise, documents are frequently spread across many different data sources. To make the information accessible and easy to find for everybody, the ideal solution is to save the data in one central content repository, such as the one provided by OracleAS Portal. The problem is, how do you move distributed content into OracleAS Portal and publish it there? For simple, distributed, low volume file transfer, you can simply map a Portal Repository as a Web folder. But what if you need to move a great deal of content, in hundreds or even thousands of files? This Technical Note describes how to use WebDAV clients like Sitecopy and Cadaver to move or copy file-based content from a file system or external repository into a Portal Repository, and to publish that content in OracleAS Portal. These specialized clients can greatly improve transfer efficiency for one-time bulk copy/move operations of large file volumes or periodic copy/move operations between source repositories and Portal Repositories. This Technical Note covers how to install, configure, and use these two open source clients – Cadaver and Sitecopy – to accomplish these tasks, and explains how to set the right attribution after uploading content with a WebDAV client.

CONTENT PUBLISHING WITH WEBDAV CLIENTS

OracleAS Portal supports the WebDAV protocol, which allows you to externally mount the Portal Repository and perform operations on it. These operations include not only moving external files into the Portal Repository and automatically publishing those files on the desired page within Portal, but also copying/moving/editing files within Portal. For more information, please check the pointers in the Related Material section below.

Oracle has enhanced two of the most popular open source WebDAV clients, Sitecopy and Cadaver, by adding cookie support to the source code of those clients. This enables the user to authenticate only once to the Portal Repository, thus improving performance dramatically.

WHEN TO USE WHAT ?

Now that you can choose among Microsoft Web folders, Sitecopy, and Cadaver, how do you know which client is best for which tasks?

Microsoft Web Folders

This WebDAV client, which is available out-of-the-box in Windows, is the best of these three choices for end users like content contributors. This WebDAV client allows users to map the Portal Repository like a windows drive, edit documents in-place, and save them back to the Portal Repository directly. Since the Portal Repository is displayed like a folder hierarchy in Windows Explorer, users can copy/move/delete files without any additional knowledge.

Cadaver

The Cadaver WebDAV client is similar to an ftp or telnet client. Commands like put and get are used to upload and download files, so you can copy/move/delete items within the Portal Repository just as you would in telnet. This client is best for administrators who want to perform bulk operations. Because Cadaver is an open source client, it is available on every Unix and Windows operating system.

Sitecopy

Sitecopy is ideal for performing script-based bulk operations on the OracleAS Portal Repository. One can replicate file system directory structures to OracleAS Portal as well as Portal page structures to the file system. Choose Sitecopy if you plan to replicate other repositories supporting WebDAV and the Portal Repository automatically.

USING CADAVER

Installing Cadaver on UNIX

1. Download the file `cadaver-0.20.5.tar.gz` to your local file system.
2. Extract the decompressed file by typing **gunzip cadaver-0.20.5.tar.gz**
3. Use the tar command to extract the install directory: **tar -xvf cadaver-0.20.5.tar**
4. Change to the created directory and type **./configure**
5. When the configure command finishes, type **make**
6. Login as the root user, change to this directory, and type **make install**. Logout as the root user.
7. Start cadaver by typing **cadaver**.

Installing Cadaver on Windows

Before installing Cadaver on Windows, you must install an environment called Cygwin. Cygwin is a free bash shell emulator for Windows that allows you to build the Cadaver executable.

1. Go to the [Cygwin Website](#) and click the *Install Cygwin now* link to execute a setup program.
2. On the first screen, choose *Install from Internet*.
3. On the second screen, specify the install location. Make sure the target folder name does not contain any spaces. For all other options, accept the default.
4. On the third screen, specify the directory in which to store the downloaded packages.
5. On the next two screens you have to specify the proxy settings for the package download and you can choose a server from the list where the packages gets downloaded from.
6. At the *Select Packages* screen click the *View* button to change to the *Full View* that displays all available packages. Accept all the defaults and in addition select the following packages by clicking on the icon left to *Skip* in the *New* column: **bash, cygwin, libiconv, libxml2 and gettext, tar, gzip, make, gcc**. Click *Next* to install the packages.
6. Start the Cygwin bash shell by clicking the Cygwin icon on the desktop. All executables from the selected packages are installed under `<CYGWIN_DIRECTORY\bin>` and are accessible from the bash shell under `/bin`.
7. Add the `/bin` directory to the path:

```
export PATH=$PATH:/bin
```

Now you are ready to build the Cadaver executable. To do so, follow the Cadaver UNIX installation instructions.

Hint: To change drive letters in the Cygwin bash shell, type:

```
cd <drive_letter>:
```

After you built the Cadaver executable, you may want to add the directory containing `cadaver.exe` to your bash shell path.

Command Overview

After you start Cadaver, you can display a list of available commands by typing **help** on the command line.

Command	Description
ls [path]	List contents of current [or other] collection
cd path	Change to specified collection
pwd	Display name of current collection
put local [remote]	Upload local file
get remote [local]	Download remote resource
mget remote...	Download many remote resources
mput local...	Upload many local files
edit resource	Edit given resource
less remote...	Display remote resource through pager
mkcol remote...	Create remote collection(s)
cat remote...	Display remote resource(s)
delete remote...	Delete non-collection resource(s)
rmcol remote...	Delete remote collections and ALL contents
copy source... dest	Copy resource(s) from source to dest
move source... dest	Move resource(s) from source to dest
lock resource	Lock given resource
unlock resource	Unlock given resource
discover resource	Display lock information for resource
steal resource	Steal lock token for resource
showlocks	Display list of owned locks
propnames res	Names of properties defined on resource
chexec [+ -] remote	Change isexecutable property of resource
propget res [proprname]	Retrieve properties of resource
propdel res proprname	Delete property from resource
propset res proprname value	Set property on resource
set [option] [value]	Set an option or display options
open URL	Open connection to given URL
close	Close current connection
quit	Exit program
unset [option] [value]	Unsets or clears value from option.
lcd [directory]	Change local working directory
lls [options]	Display local directory listing
lpwd	Print local working directory

logout	Logout of authentication session
help [command]	Display help message
Aliases: rm=delete, mkdir=mkcol, mv=move, cp=copy, more=less, quit=exit=bye	

Using Cadaver with OracleAS Portal

To connect to the Portal Repository, use the **open** command, as in this example:

open http://chauser-pc.at.oracle.com:7778/dav_portal/portal

You are prompted for your user name and password. Note that you connect with your Single Sign-on user name.

Using Cadaver to Copy a File From the File System to the Portal Repository

Before copying a file, navigate to the page where you want to publish the file. After signing into Portal:

- To display a list of page groups you can access, use the **ls** command.
- To navigate to a page within a page group, use the **cd** command.
- To copy a single file from the file system to a page, use the **put** command.
- To copy multiple files from the file system to Portal at the same time, use the **mput** command. For example, to copy all files from a directory to a OracleAS Portal page, use **mput *.*.**
- To copy a file from one page to another page, use the **copy** command. For example, to copy the file README to the folder aa (assuming aa is at the same folder hierarchy level as the folder you are in) use: **copy README ../aa/README.**

USING SITECOPY

Installing Sitecopy on UNIX

1. Download `sitecopy-0.11.4-with_cookies.tar.gz` files to your machine
2. Extract the decompressed file by typing **gunzip sitecopy-0.11.4-with_cookies.tar.gz**
3. Use the tar command to extract the install directory: **tar -xvf sitecopy-0.11.4-with_cookies.tar**
4. Change to the created directory and type **./configure**
5. When the configure command finishes, type **./make**
6. Login as the root user, change to this directory, and type **./make install**
Logout as the root user.

Installing Sitecopy on Windows

Follow the instructions given in the *Installing Cadaver on Windows* section above. After you have installed and configured the Cygwin environment, complete the steps in Installing Sitecopy on UNIX.

Command Overview

Once you have installed Sitecopy, you can display a list of all available options by typing **sitecopy -help**:

Usage: sitecopy [OPTIONS] [MODE] [sitename]...

Command	Description
OPTIONS	
	Turn debugging on for each KEY, which may be: socket, files, rcfile, ftp, http, httpbody, xml, xmlparse, cleartext
-d, --debug=KEY[,KEY]	Warning: cleartext displays (normally hidden) passwords in plain text
-g, --logfile=FILE	Append debugging messages to FILE (else use stderr)
-r, --rcfile=FILE	Use alternate run control file
-p, --storepath=PATH	Use alternate site storage directory
-y, --prompting	Request confirmation before making each update
-a, --allsites	Perform the operation on ALL defined sites
-k, --keep-going	Carry on an update regardless of errors
-o, --show-progress	Display total percentage file transfer complete
-q, --quiet	Be quiet while performing the operation
-qq, --silent	Be silent while performing the operation
MODE	
-l, --list	List changes between remote and local sites (default)
-ll, --flatlist	Flat list of changes between remote and local sites
-v, --view	Display a list of the site definitions
-i, --initialize	Mark all files and directories as not updated
-f, --fetch	Find out what files are on the remote site
-e, --verify	Verify stored state of site matches real remote state
-c, --catchup	Mark all files and directories as updated
-s, --synchronize	Update the local site from the remote site
-u, --update	Update the remote site
-h, --help	Display this help message
-V, --version	Display version information

Using Sitecopy with OracleAS Portal

Before using Sitecopy with OracleAS Portal, you must create a run control file to define WebDAV access to the Portal Repository, as well as to the directory on the file system you plan to synchronize from/to.

In this example, a file called .sitecopyrc is created, which contains the following:

```
site chauser
server chauser-pc.at.oracle.com
port 7778
protocol webdav
username portal
password manager1
local /home/jferns/chauser
remote /dav_portal/portal/chauser/aa
```

site = Unique name that distinguishes multiple OracleAS Portal site entries in this file

server = Mid-tier machine

port = Mid-tier port

protocol = webdav, as this is the protocol used to connect to the Portal Repository

username = SSO OracleAS Portal user that will perform the WebDAV file upload

password = Password of the OracleAS Portal SSO user

local = Directory on the local file system used as the root for replication

remote = Path to the OracleAS Portal page to which files are replicated
/dav_portal/portal/<page group>/<page>/<page>]

After creating the file, set the file permissions to allow other users to read it:

```
chmod 0600 .sitcopyrc
```

When you start Sitecopy the first time, you will get an error similar to this:

```
sitecopy: Error: Could not open storage directory: /home/oracle/.sitecopy/
```

```
sitecopy: You need to create this directory and set the permissions to 0700.
```

Create the specified directory and grant the privileges to it.

```
cd /home/oracle
mkdir .sitecopy
chmod 0700 .sitecopy
```

Using Sitecopy to Copy Files From the File System to the Portal Repository

To initialize a connection to one or all sites defined in the run control file, use this command:

```
sitecopy --rcfile=/home/oracle/webdav/.sitcopyrc --
allsites --initialize
```

To list the directory (OracleAS Portal page) on the remote side, execute this command:

```
sitecopy --rcfile=/home/oracle/webdav/.sitcopyrc --
allsites --fetch
```

To see the differences between the target and source repositories, use the `--flatlist` option. This read-only command allows you to see what would happen if you synchronized the source with the target directory, without making any changes. The `--flatlist` option lists the files/folders that would be added to or deleted from the target repository should synchronization actually occur

```
sitecopy --rcfile=/home/oracle/webdav/.sitcopyrc --
allsites --flatlist
```

To synchronize the source repository (file system) with the target repository (the Portal Repository), use this command:

```
sitecopy --rcfile=/home/oracle/webdav/.sitcopyrc --
allsites --update
```

Note: The source repository is always the single source of truth. In other words, upon synchronization the source repository's directory/file structure is replicated to the target repository. This means that the target repository will look exactly the same as the source repository after the synchronization. The important thing to know is that elements in the target repository are physically deleted without notification.

REPLICATING CONTENT FROM AN EXTERNAL REPOSITORY USING SITECOPY: AN EXAMPLE

This section guides you through a complete example scenario using the Sitecopy WebDAV client. In this example, content from an external target repository is replicated to a Portal Repository, which acts as the source. To simplify the example, the same Portal Repository is used as the source and the target repositories, although any external repository supporting the WebDAV protocol could be used.

The replication is a two way process:

- Copying content from the source repository to the file system
- Replicating content from the file system to the Portal Repository

After replicating the content, attribution is set for the uploaded files using Oracle Portal's Content Management API's. All of these operations are performed as script-based bulk operations in batch, without any user interaction.

Here is the command configuration file for this example:

```
site sourcesite
server chauser-pc.at.oracle.com
port 7778
protocol webdav
username portal
password manager1
local /home/oracle/test
remote /dav_portal/portal/chauser/aa
site targetsite
server chauser-pc.at.oracle.com
port 7778
protocol webdav
username portal
password manager1
local /home/oracle/test
remote /dav_portal/portal/chauser/target
```

In this example, two sites are defined, the source and the target site. Content will be replicated from the source repository (page group chauser, page aa) to the file system (directory /home/oracle/test). The second site, called targetsite (page group chauser, page target) will use the file system as the source from which to replicate content.

Step 1 - Establish a connection to the source repository

Here is the command used to connect to the source repository:

```
sitecopy --rcfile=/home/oracle/webdav/.sitecopyrc --
initialize sourcesite
```

This message is displayed:

```
sitecopy: Initializing site `sourcesite' (on chauser-pc.at.oracle.com in
/dav_portal/portal/chauser/aa/)
```

```
sitecopy: All the files and directories are marked as NOT updated remotely.
```

Step 2 - Copying files from the target repository to the file system

Command:

```
sitecopy --rcfile=/home/oracle/webdav/.sitecopyrc --  
synchronize sourcesite
```

Message:

sitecopy: Synchronizing site `sourcesite' (on chauser-pc.at.oracle.com in /dav_portal/portal/chauser/aa/)

Downloading Top.csv: [...] done.

Downloading 132843.gif: [...] done.

Downloading quickinstall.pdf: [.....] done.

sitecopy: Synchronize completed successfully.

Note that this command copies all the files from the source repository to the target file system directory and deletes all files from the file system directory. There is always only one source of truth – you can decide which side this is by running in *synchronize* or *update* mode.

Step 3 - Establish a connection to the target repository

Command:

```
sitecopy --rcfile=/home/oracle/webdav/.sitecopyrc --  
initialize targetsite
```

Message:

sitecopy: Initializing site `targetsite' (on chauser-pc.at.oracle.com in /dav_portal/portal/chauser/target/)

sitecopy: All the files and directories are marked as NOT updated remotely.

Step 4 - Replicating files from the file system directory to the target page

Command:

```
sitecopy --rcfile=/home/oracle/webdav/.sitecopyrc --  
update targetsite
```

Message:

sitecopy: Updating site `targetsite' (on chauser-pc.at.oracle.com in /dav_portal/portal/chauser/target/)

Uploading quickinstall.pdf: [.....] done.

Uploading 132843.gif: [...] done.

Uploading Top.csv: [...] done.

sitecopy: Update completed successfully.

Note that if there are sub-directories in the file system directory, corresponding sub-pages on the target repository page are created.

Important: OracleAS Portal allows you to pre-define the item types for uploaded files. For example, you may want to upload images as image item types instead of the default file item type. To configure this, set the page properties in the browser-based interface on the Page Properties > Optional tab, as shown here:

Default WebDAV Types

Choose the item types that you want to use for files added to this page using WebDAV. For example, you can choose to add files as Zip File items rather than Simple File items so that they can be unzipped within the page.

Default Regular Files	Use Default - File
Default Zip Files	Use Default - Zip File
Default Image Files	Use Default - File

Use Default - File
File
Image
Simple File
Simple Image
Simple Image Map
Zip File

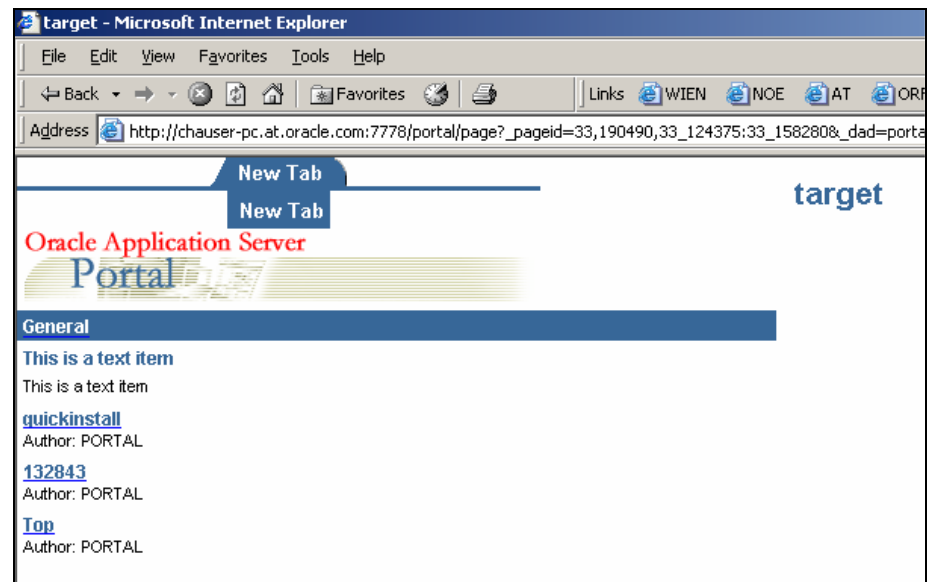
Optional Page Properties

Select the level of item versioning to use for the page: None always overwrites the previous version or archiving it in the database, Audit, etc. Select the user to contact about the page. Choose the category, and enter keywords.

Step 5 - Resetting attribution for uploaded files

The WebDAV standard does not support the ability to transfer content attributes. In use cases where item attributes/attribute values are important, you can set them individually using the OracleAS Portal browser interface, or in bulk using the OracleAS Portal [Content Management APIs](#) and supported views. In the Page Group Properties tab set, you can choose which default category is assigned to all uploaded files through WebDAV. On the *Configure* tab in the *Content Type and Classification* section you can configure this by moving the default category for uploaded files on the top of the list of visible categories. In many cases you will want to modify the files' metadata information, such as display name, category, description or author.

In this example, three files are copied from the file system to the Portal Repository. Looking at the uploaded files, you can see that the category is defaulted to *General* and the author defaults to the SSO user who uploaded the files through WebDAV, in this case *PORTAL*.



Now let's assume that you want to modify the attribution of the uploaded files, change the category from *General* to *Sales Documents*, and change the author to *John Doe*. You do not want to influence any other items on this page, such as the text item shown here, so only items of the item type *File* should be modified.

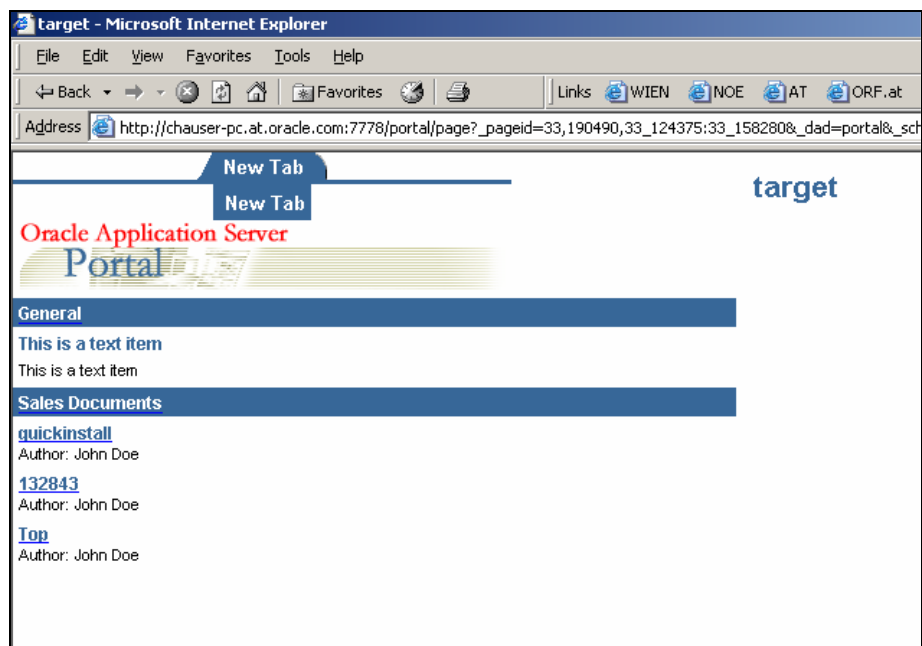
The following procedure shows you how to use the Content Management API's to perform this task.

```

declare
  var_funct_return number;
  var_masterid number;
  var_itemid number;
  var_displayname varchar2(200);
  cursor c1 is select masterid, id, display_name from
  wwsbr_all_items where caid=33 and folder_id=190490
  and itemtype='basefile';
  /* Cursor that selects attributes of the items that
  have been uploaded to this page. Only items with item
  type basefile (=file) are taken into account. The
  assumption is that you know to which page group and
  page the files have been uploaded. As specified in the
  sitecopy run command file the pagegroup is chauser (id
  33). The page is target (id 190490). These id's can be
  selected in the view WWSBR_ALL_ITEMS */
begin
  portal.wwctx_api.SET_CONTEXT('portal','manager1','');
  -- set the context and authenticate with SSO username
  -- and password
  open c1;
  loop
    fetch c1 into var_masterid, var_itemid,
    var_displayname;
  /* displayname is a mandatory argument; since we don't
  want to change it we select it in the cursor and pass
  it back to the modify_item function. */
    var_funct_return := wwsbr_api.modify_item(
      p_master_item_id => var_masterid,
      p_item_id => var_itemid,
      p_caid => 33,
      p_folder_id => 190490,
      p_region_id => 41093,
      p_display_name => var_displayname,
      p_author => 'John Doe',
  /* The author gets defaulted to the user who opens the
  WebDAV connection. We will reset it to the Author
  'John Doe' */
      p_category_id => 190504);
  /*The category gets defaulted to GENERAL. we will
  reset it to 'Sales Documents' that has the category_id
  190504 */
    exit when c1%notfound;
  end loop;
  close c1;
  wwpro_api_invalidation.execute_cache_invalidation;
  -- Invalidate cache so that changes are immediately --
  -- visible in portal
end;
/

```

Here is the final result:



Note:

This is just a simple example of how you can use the Content Management API's. The API's can be used to implement exactly the same attribution as in the source repository. If the source repository is an OracleAS Portal metadata repository, you can query exactly the same tables (e.g. WWSBR_ALL_ITEMS) to get the metadata information available for each item and use it for the uploaded files in the target repository. If the source repository is a third party repository and the metadata information is accessible somewhere else, an additional step is needed to prepare the metadata to use it with the Portal Content Management API's.

INSTALLATION TROUBLESHOOTING

This section describes some known installation problems and their workarounds.

Compiler environment missing

If you do a `./configure` and you get the following error message, either you do not have a compiler in your path or you do not have the building environment installed on your machine.

```
$ ./configure
checking for gcc... no
checking for cc... no
checking for cc... no
checking for cl... no
configure: error: no acceptable cc found in $PATH
```

Sitecopy on RedHat Advanced Server 3.0

The file `<SITECOPY_DIRECTORY>/lib/yesno.c` must be modified before running `configure`, `make`, and `make install`. Add the following to the beginning of the include section:

```
add #include <xlocale.h>
```

The file <**SITECOPY_DIRECTORY**>/libneon/ne_xml.c must be modified.
Change the following include:

```
# include <parser.h>
```

to:

```
# include <libxml/parser.h>
```

Sitecopy on Sun Solaris
No known installation problems.

Cadaver on RedHat Advanced Server 3.0
No known installation problems.

Cadaver on RedHat Advanced Server 2.1
No known installation problems.

Cadaver on RedHat Advanced HP-UX
No known installation problems.

Cadaver on RedHat Advanced HP Tru64
No known installation problems.

Cadaver on Sun Solaris
No known installation problems.

Cadaver on Windows
No known installation problems.

SUPPORT

Any problems or questions related to this Technical Note are supported through the [Content Management Forum](#) on OTN. Please do not open any Service Request with Oracle Support Services regarding this Technical Note.

RELATED MATERIAL

[OracleAS Portal User's Guide 10g\(9.0.4\) – Chapter 2.5](#)

[OracleAS Portal Configuration Guide 10g\(9.0.4\) – Chapter 4.10](#)

[Cadaver Website](#)

[Sitecopy Website](#)

[Portal on OTN](#)

[Content Management API Documentation](#)

[Cygwin Homepage](#)

[Sitecopy on Windows](#)



Using WebDAV Clients to replicate external content into Portal
August 2004

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

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com

Copyright © 2004, Oracle. All rights reserved.

This document is provided for information purposes only
and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to
any other warranties or conditions, whether expressed orally
or implied in law, including implied warranties and conditions of
merchantability or fitness for a particular purpose. We specifically
disclaim any liability with respect to this document and no
contractual obligations are formed either directly or indirectly
by this document. This document may not be reproduced or
transmitted in any form or by any means, electronic or mechanical,
for any purpose, without our prior written permission.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective owners.