

# **VERITAS Storage Foundation™ 4.1 *for Oracle RAC***

## **Installation and Configuration Guide**

**HP-UX**

---

## Disclaimer

The information contained in this publication is subject to change without notice. VERITAS Software Corporation makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. VERITAS Software Corporation shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual.

## VERITAS Legal Notice

Copyright © 2005 VERITAS Software Corporation. All rights reserved. VERITAS and the VERITAS Logo are trademarks or registered trademarks of VERITAS Software Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

VERITAS Software Corporation  
350 Ellis Street  
Mountain View, CA 94043  
USA  
Phone 650-527-8000 Fax 650-527-2908  
[www.veritas.com](http://www.veritas.com)

## Third-Party Legal Notices

### Apache Software

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work.

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source

---

code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

- (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
- (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
- (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
- (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks.** This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty.** Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability.** In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. **Accepting Warranty or Additional Liability.** While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.



---

## Data Encryption Standard (DES)

Support for data encryption in VCS is based on the MIT Data Encryption Standard (DES) under the following copyright:

Copyright © 1990 Dennis Ferguson. All rights reserved.

Commercial use is permitted only if products that are derived from or include this software are made available for purchase and/or use in Canada. Otherwise, redistribution and use in source and binary forms are permitted.

Copyright 1985, 1986, 1987, 1988, 1990 by the Massachusetts Institute of Technology. All rights reserved.

Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.

WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of M.I.T. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission. M.I.T. makes no representations about the suitability of this software for any purpose. It is provided as is without express or implied warranty.

## SNMP Software

SNMP support in VCS is based on CMU SNMP v2 under the following copyright:

Copyright 1989, 1991, 1992 by Carnegie Mellon University

All Rights Reserved

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of CMU not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.

CMU DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL CMU BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.



# Contents

---

<b>Preface</b> .....	<b>xix</b>
How This Guide Is Organized .....	xix
Related Documentation .....	xxi
Conventions .....	xxiv
Getting Help .....	xxv
Documentation Feedback .....	xxv
<b>Chapter 1. Overview: Storage Foundation for Oracle RAC</b> .....	<b>1</b>
What is RAC? .....	1
RAC Architecture .....	1
Operating System-Dependent Layer .....	3
Veritas Cluster Server Membership Manager (VCSMM) .....	3
Cluster Inter-Process Communication (VCSIPC) .....	3
Lock Management and Cache Fusion .....	3
Shared Disk Subsystems .....	3
Storage Foundation for Oracle RAC Overview .....	4
Data Stack Overview .....	4
Communications Stack Overview .....	5
Storage Foundation for Oracle RAC Communications .....	6
Low Latency Transport (LLT) .....	6
VCSIPC .....	6
Traffic Distribution .....	6
Heartbeat .....	6



---

Group Membership Services/Atomic Broadcast (GAB) .....	7
Cluster Membership .....	7
Cluster Communications .....	7
Cluster Volume Manager (CVM) .....	8
CVM Architecture .....	8
Cluster File System .....	9
CFS Architecture .....	9
CFS Usage in Storage Foundation <i>for Oracle RAC</i> .....	9
Oracle Disk Manager .....	9
ODM Clustering Extensions .....	10
VERITAS Cluster Server .....	10
VCS Architecture .....	10
Storage Foundation <i>for Oracle RAC</i> Service Groups .....	10
Storage Foundation <i>for Oracle RAC</i> OSD Layer Support .....	12
Cluster Membership (VCSMM) .....	12
Inter-Process Communications (VCSIPC) .....	12
I/O Fencing .....	13
Understanding Split Brain and the Need for I/O Fencing .....	13
SCSI-3 Persistent Reservations .....	13
I/O Fencing Components .....	14
Data Disks .....	14
Coordinator Disks .....	15
I/O Fencing Operations .....	15
<b>Chapter 2. Preparing to Install Storage Foundation <i>for Oracle RAC</i> .....</b>	<b>17</b>
High-Level View of Storage Foundation for Oracle RAC .....	17
Phases of Installation and Configuration .....	19
Phase One: Preparing and Configuring Hardware .....	19
Phase Two: Installing SFRAC and Configuring its Components .....	19
Phase Three: Installing Oracle9i .....	20



---

Phase Four: Creating the Database .....	20
Phase Five: Setting up VCS to Manage RAC Resources .....	20
Requirements for Installing Components .....	21
License Keys .....	21
Using the VERITAS vLicense™ Web Site to Obtain a License Key .....	21
Faxing the License Key Request Form to Obtain a License Key .....	22
Oracle RAC Versions .....	22
HP Operating Systems .....	22
Operating Environments .....	23
HP Patches .....	23
Disk Space (Local) .....	24
Synchronized Cluster Nodes .....	24
System-to-System Communication .....	24
Removing Pre-existing License Keys .....	25
Information Required to Install SFRAC .....	26
Cluster Configuration .....	26
Web-based Cluster Manager (Optional) .....	26
SMTP/SNMP (Optional) .....	26
<b>Chapter 3. Installing SFRAC 4.1 and Configuring Components .....</b>	<b>27</b>
Installing SFRAC 4.1 Packages .....	27
Setting the PATH Variable .....	28
Setting the MANPATH Variable .....	28
Mounting the VERITAS SFRAC Disc .....	28
Using the VERITAS Product Installer to Install SFRAC .....	30
Using the installsfrac Script to Install SFRAC .....	31
Verifying Installation Requirements .....	31
installer: Installing Packages .....	34
Summary of Installation .....	35



---

Configuring the Cluster and Optional Features .....	36
Mounting the VERITAS SFRAC Disc and Configuring VxSS .....	36
Using the VERITAS Product Installer to Configure SFRAC .....	38
Using the installsfrac -configure Script to Configure SFRAC .....	39
Checking Requirements for Configuring Cluster .....	39
Configuring Heartbeat Links, VxSS, Cluster Manager, and SMTP/SNMP Notifi- cation .....	41
Starting VxVM and Specifying VxVM Information .....	45
Summary of Configuration .....	47
Setting Up Shared Storage and I/O Fencing for SFRAC .....	48
Adding and Initializing Disks .....	48
Verifying the Nodes See the Same Disk .....	49
Testing Disks Using vxfcntlshdw .....	51
Reviewing Guidelines on vxfcntlshdw .....	51
Running vxfcntlshdw .....	51
Setting Up Coordinator Disks .....	54
Requirements for Coordinator Disks .....	54
Creating the vxfcntlshdw Disk Group .....	55
Requirements for Testing the Coordinator Disk Group .....	55
Using vxfcntlshdw -c to Test the Coordinator Disk Group .....	55
Configuring /etc/vxfcntlshdw Disk Group for I/O Fencing .....	57
Starting I/O Fencing .....	58
Removing remsh Permissions and Restoring Public Network Connections ..	58
Editing the UseFence Attribute in VCS Configuration .....	59
Starting VCS, CVM, and CFS on All Nodes .....	60
Running gabconfig -a to Verify GAB Port Membership .....	60
Verifying the CVM Group is Online .....	61
Unmounting the Software Disc .....	61
Rebooting Nodes (General) .....	61





---

Example of VCS Configuration File After SFRAC Installation .....	62
Sample main.cf: After SFRAC Installation and Before Oracle9i Installation .....	63
<b>Chapter 4. Installing Oracle9i Software .....</b>	<b>65</b>
Prerequisite for Installing Oracle9i .....	65
Before the Installation: Reviewing Shared Disk Groups and Volumes .....	66
Viewing Information on a Disk Group .....	66
Checking the Connectivity Policy on a Shared Disk Group .....	66
Determining Whether a Node is CVM Master or Slave .....	66
Enabling Write Access to Volumes in the Disk Groups .....	67
Exporting and Importing Shared Disk Groups .....	67
Creating the Oracle9i User and Group .....	68
Installing and Configuring Oracle9i: Overview .....	69
Creating Shared Storage Location for the SRVM .....	69
Method 1: Creating a Separate Location for SRVM .....	69
Method 2: Creating a Cluster File System for SRVM (Optional) .....	70
Installing Oracle9i Release 2 Software .....	71
Method 1: Installing Oracle9i Release 2 on Cluster File System .....	71
Method 2: Installing Oracle9i Release 2 on Local File System .....	75
Adding Oracle9i Release 2 Patches .....	80
Linking to VERITAS Libraries .....	81
Disabling ODM Library (Optional) for Databases Using Raw Devices .....	82
Linking to VERITAS Libraries When Upgrading Oracle Patch Level .....	83
Creating the Databases .....	83
Upgrading the Databases .....	83
Stopping Applications Not Under VCS Control .....	83
Stopping gsd .....	83



---

Stopping Applications using CFS (Outside of VCS Control) .....	83
Unmounting CFS File Systems Outside of VCS Control .....	84
<b>Chapter 5. Configuring VCS Service Groups for Oracle9i .....</b>	<b>85</b>
Service Groups in SFRAC: Overview .....	85
The CVM Service Group .....	86
Configuring Service Groups to Automate the RAC Environment .....	87
Is Oracle Installed on Shared or Local Storage? .....	87
Is SRVM Located in ORACLE_HOME or a Separate Disk Group? .....	87
Does the Listener Use a Virtual IP Address? .....	87
Applications to Address RAC Databases on the Same Server? .....	88
Understanding VCS Service Group Dependency Capabilities and Limitations ..	88
Typical Configurations for Oracle and CVM Service Groups .....	90
Configuring CVM and Oracle Service Groups .....	95
Configuring CVM and Oracle Service Groups Manually .....	95
Example 1: CVM Group with Oracle Binaries on CFS .....	95
Example 2: CVM Group with Oracle Binaries Installed Locally .....	98
Creating Oracle Service Group in the main.cf File .....	101
Saving and Checking the Configuration .....	103
Additional RAC Processes Monitored by the VCS Oracle Agent .....	103
Attributes of CVM, Oracle Groups Defined as Local .....	104
Modifying the VCS Configuration .....	105
Location of VCS and Oracle Agent Log Files .....	105
Configuring the Service Groups Using the Wizard .....	105
Creating Oracle Service Group - Overview .....	105
Before Starting the Wizard .....	105
Establishing Graphical Access for the Wizard .....	106
Starting the Configuration Wizard .....	107
The Welcome Window .....	107



---

The Wizard Discovers the RAC Configuration .....	108
The Wizard Options Screen .....	108
Creating a RAC Service Group .....	109
Database Selection Screen .....	109
Instance Configuration Screen .....	110
Detail Monitoring Screen - Oracle Resource .....	111
Oracle Advanced Configuration Screen .....	112
Database Configuration Screen .....	113
Listener Configuration Screen .....	114
Service Group Summary Screens .....	116
Modifying Oracle RAC Service Groups Using the Wizard .....	118
<b>Chapter 6. I/O Fencing Testing and Scenarios .....</b>	<b>119</b>
I/O Fencing of Shared Storage .....	119
Verifying Data Storage Arrays Using the vxfststhdw Utility .....	120
General Guidelines for Using vxfststhdw .....	120
vxfststhdw Options .....	121
Using the -r Option for Non-destructive Testing .....	121
Using the -m Option .....	121
Using the -f Option .....	122
Using the -g Option .....	122
Testing a Disk with Existing Keys .....	123
How I/O Fencing Works in Different Event Scenarios .....	124
The vxfenadm Utility .....	127
Registration Key Formatting .....	128
<b>Chapter 7. Using Storage Checkpoints and Storage Rollback .....</b>	<b>129</b>
Using Storage Checkpoints and Storage Rollback for Backup and Restore .....	129
Understanding Storage Checkpoints and Storage Rollback .....	130
Determining Space Requirements for Storage Checkpoints .....	131
Performance of Storage Checkpoints .....	132



---

Setting Up the SFRAC Storage Checkpoint Feature for Use .....	133
Backing Up and Recovering the Database Using Storage Checkpoints .....	135
Backing Up Using the Command Line .....	136
Usage Notes .....	136
Verifying a Storage Checkpoint Using the Command Line .....	137
Usage Notes .....	137
Guidelines for Oracle Recovery .....	139
Using the Storage Checkpoint Command Line Interface (CLI) .....	141
Overview of Commands .....	141
Examples of Using the Command Line Interface .....	142
Prerequisites .....	142
Creating or Updating The Repository Using <code>sfrac_update</code> .....	143
Prerequisites .....	143
Usage Notes .....	143
Creating Storage Checkpoints Using <code>sfrac_ckptcreate</code> .....	144
Usage Notes .....	144
Displaying Storage Checkpoints Using <code>sfrac_ckptdisplay</code> .....	145
Usage Notes .....	145
Scheduling Storage Checkpoints Using <code>sfrac_ckptcreate</code> and <code>cron</code> .....	146
Usage Notes .....	146
Scheduling Storage Checkpoint creation in a cron Job .....	147
Mounting Storage Checkpoints Using <code>sfrac_ckptmount</code> .....	148
Usage Notes .....	148
Unmounting Storage Checkpoints Using <code>sfrac_ckptumount</code> .....	149
Usage Notes .....	149
Performing Storage Rollback Using <code>sfrac_ckptrollback</code> .....	150
Usage Notes .....	150
Removing Storage Checkpoints Using <code>sfrac_ckptremove</code> .....	151
Usage Notes .....	151



---

<b>Chapter 8. Using Storage Mapping</b> .....	<b>153</b>
Understanding Storage Mapping .....	153
Verifying VERITAS Storage Mapping Setup .....	154
Using vxstorage_stats .....	155
Displaying Storage Mapping Information .....	155
Displaying I/O Statistics Information .....	155
Oracle File Mapping (ORAMAP) .....	157
Mapping Components .....	157
Storage Mapping Views .....	158
Verifying Oracle File Mapping Setup .....	159
Enabling Oracle File Mapping .....	159
Accessing Dynamic Performance Views .....	160
Using Oracle Enterprise Manager .....	161
Configuring Arrays for Storage Mapping and Statistics .....	163
<b>Chapter 9. Adding and Removing SFRAC Nodes</b> .....	<b>165</b>
Adding a Node .....	165
Phase 1 - Checking System Requirements for New Node .....	166
Phase 2 - Physically Adding a New Node to the Cluster .....	166
Phase 3 - Installing SFRAC on the New Node .....	166
Checking the New Node for Installation .....	166
Installing SFRAC on New Node Without Configuration .....	168
Phase 4 - Running vxinstall .....	169
Phase 5 - Configuring LLT, GAB, VCSMM, ODM, and VXFEN Drivers .....	174
Phase 6 - Configuring CVM .....	175
Phase 7 - Setting Up or Installing Oracle .....	177
Phase 8 - Configuring New Oracle Instance .....	180
Removing a Node .....	183
Stopping gsd .....	183
Stopping Applications Using CFS Outside of VCS Control .....	183



---

Stopping VCS .....	183
Uninstalling Oracle if Installed Locally (Optional) .....	184
Unlinking the Oracle Binary from VERITAS Libraries (Required if Oracle is Installed Locally) .....	184
Running the uninstallsfrac Utility .....	185
Editing VCS Configuration Files on Existing Nodes .....	187
Editing /etc/llthosts .....	187
Editing /etc/gabtab .....	187
Modifying the VCS Configuration to Remove the Node .....	188
Removing Temporary Files .....	189
Removing the Infrastructure Packages (Optional) .....	189
Removing Licenses .....	189
<b>Chapter 10. Upgrading to SFRAC 4.1 .....</b>	<b>191</b>
Uninstalling DBE/AC for Oracle9i RAC .....	191
Stopping gsd and Applications Using CFS .....	192
Unmounting CFS File Systems Outside of VCS Control .....	192
Taking Oracle and Sqlnet Resources Offline .....	193
Unlinking the Oracle Binary From VERITAS Libraries .....	193
Taking the Oracle Service Groups Offline .....	195
Using the uninstallDBAC Script .....	195
Backing up and Removing Configuration Files .....	201
Upgrading the HP-UX Operating System .....	201
Installing and Configuring SFRAC 4.1 .....	202
Installing License Key for Storage Mapping and Storage Checkpoints .....	202
Copying the VCS Configuration File .....	202
Editing the VCS Configuration File .....	202
Verifying and Copying the main.cf to Cluster Nodes .....	203
Starting VCS, CVM, and CFS on All Nodes .....	203



---

Upgrading the Disk Layout .....	203
Linking SFRAC Libraries with Oracle .....	204
<b>Chapter 11. Uninstalling SFRAC 4.1 .....</b>	<b>205</b>
Stopping gsd .....	205
Stopping Applications Using CFS (Outside of VCS Control) .....	205
Unmounting CFS File Systems (Outside of VCS Control) .....	205
Taking the Oracle and Netlsnr Resources Offline .....	206
Removing the Oracle Database (Optional) .....	206
Uninstalling Oracle9i (Optional) .....	206
Unlinking the Oracle Binary From VERITAS Libraries .....	207
Stopping VCS .....	208
Running the installer or uninstallsfrac Script .....	209
Using the VERITAS Product Installer to Uninstall SFRAC .....	209
Using the uninstallsfrac Script to Uninstall SFRAC .....	209
Removing VERITAS Infrastructure Packages (Optional) .....	212
Removing License Files .....	213
Removing Other Configuration Files (Optional) .....	213
<b>Chapter 12. Troubleshooting .....</b>	<b>215</b>
Running Scripts for Engineering Support Analysis .....	215
getdbac .....	215
getcomms .....	215
hagetcf .....	215
Troubleshooting Topics .....	216
Oracle User Must be Able to Read /etc/llttab File .....	216
Missing Dialog Box During Installation of Oracle9i Release 2 .....	216
Error When Starting an Oracle Instance .....	217
Instance Numbers Must be Unique (Error Code 205) .....	217
ORACLE_SID Must be Unique (Error Code 304) .....	217



Oracle Log Files Show Shutdown Called Even When Not Shutdown Manually	217
Set MLOCK Privilege for DBA User	217
File System Configured Incorrectly for ODM Shuts Down Oracle	218
VCSIPC Wait Warning Messages in Oracle Trace/Log Files	219
VCSIPC Errors in Oracle Trace/Log Files	219
Shared Disk Group Cannot be Imported	220
CVMVolDg Does Not Go Online Even Though CVMCluster is Online	220
Restoring Communication Between Host and Disks After Cable Disconnection	220
Node is Unable to Join Cluster While Another Node is Being Ejected	221
vxfsentsthdw Fails When SCSI TEST UNIT READY Command Fails	221
Removing Existing Keys From Disks	222
System Panics to Prevent Potential Data Corruption	223
How vxfsen Driver Checks for Pre-existing Split Brain Condition	223
Case 1: System 2 Up, System 1 Ejected (Actual Potential Split Brain)	224
Case 2: System 2 Down, System 1 Ejected (Apparent Potential Split Brain)	224
Using vxfsenclearpre Command to Clear Keys After Split Brain	225
Removing or Adding Coordinator Disks	226
<b>Appendix A. Sample Configuration Files</b>	<b>229</b>
Sample 1: main.cf	229
Sample 2: types.cf	234
Sample 3: CFSTypes.cf	239
Sample 4: CVMTypes.cf	240
Sample 5: OracleTypes.cf	240
Sample 6: VRTSWebAppType.cf	241
<b>Appendix B. CVMCluster, CVMVxconfigd, CVMVolDg, and CFSSMount Agents</b>	<b>243</b>
CVMCluster Agent	243
CVMCluster Agent: Entry Points	243
CVMCluster Agent: Type Attributes	244





---

CVMCluster Agent: Type Definition .....	244
CVMCluster Agent: Sample Configuration .....	245
CVMVxconfigd Agent .....	246
CVMVxconfigd Agent: Entry Points .....	246
CVMVxconfigd Agent: Type Attribute .....	246
CVMVxconfigd: Type Definition .....	247
CVMVxconfigd Agent: Sample Configuration .....	247
Configuring the CVMVolDg and CFSSMount Resources .....	248
CVMVolDg Agent: Entry Points .....	248
CVMVolDg Agent: Type Attributes .....	249
CVMVolDg Agent: Type Definition .....	249
CVMVolDg Agent: Sample Configuration .....	249
CFSSMount Agent: Entry Points .....	250
CFSSMount Agent Type: Attribute Descriptions .....	250
CFSSMount Agent: Type Definition .....	251
CFSSMount Agent: Sample Configuration .....	252
<b>Appendix C. Tunable Kernel Driver Parameters .....</b>	<b>253</b>
LMX Tunable Parameters .....	253
Configuring LMX Parameters .....	254
VCSMM Tunable Parameters .....	255
Configuring VCSMM Parameters .....	255
VXFEN Tunable Parameters .....	256
Configuring the VXFEN Parameters .....	256
<b>Appendix D. Error Messages .....</b>	<b>259</b>
LMX Error Messages, Critical .....	259
LMX Error Messages, Non-Critical .....	260
VxVM Error Messages Related to I/O Fencing .....	262
VXFEN Driver Error Messages .....	262



---

VXFEN Driver Informational Message .....	262
Node Ejection Informational Messages .....	263
<b>Appendix E. Creating a Starter Database .....</b>	<b>265</b>
Creating a Starter Database on Raw Volumes Using dbca .....	265
Creating Shared Raw Volumes for Tablespaces .....	265
Running the dbca Utility for Raw VxVM Volumes .....	266
Creating a Starter Database on CFS Using dbca .....	267
Preparing a Cluster File System (CFS) for Tablespaces .....	267
Running the dbca Utility for Cluster File System .....	268
<b>Appendix F. Linking VERITAS Libraries to Oracle .....</b>	<b>269</b>
Running linkrac9i .....	269
<b>Index .....</b>	<b>271</b>



# Preface

---

The VERITAS Storage Foundation 4.1™ *for Oracle RAC* software is an integrated set of software products. This software enables administrators of Oracle Real Application Clusters (RAC) to operate a database in an environment running VERITAS Cluster Server™ (VCS) and the cluster features of VERITAS Volume Manager™ and VERITAS File System™, also known as CVM and CFS, respectively.

The *VERITAS Storage Foundation for Oracle RAC Installation and Configuration Guide* is intended for system administrators responsible for configuring and maintaining Oracle Real Application Cluster running on a VCS cluster with VxVM disk management.

This guide assumes that the administrator has a:

- ◆ Basic understanding of system and database administration
- ◆ Working knowledge of the operating system
- ◆ Working knowledge of Oracle databases

## How This Guide Is Organized

[Chapter 1. “Overview: Storage Foundation for Oracle RAC” on page 1](#) describes the components of Oracle RAC and the Storage Foundation *for Oracle RAC* components in support of Oracle Real Application Cluster.

[Chapter 2. “Preparing to Install Storage Foundation for Oracle RAC” on page 17](#) describes the steps required in preparation for installing Storage Foundation *for Oracle RAC*.

[Chapter 3. “Installing SFRAC 4.1 and Configuring Components” on page 27](#) describes the sequence of steps to install the required software and how to perform initial configuration. The chapter includes procedures for testing the storage arrays for compliance with SCSI-3 persistent reservations.

[Chapter 4. “Installing Oracle9i Software” on page 65](#) describes how to set up shared storage and install Oracle9i or Oracle 10g software. Installation is typically on shared storage, but procedures for installing on local systems are also provided.



[Chapter 5. “Configuring VCS Service Groups for Oracle9i” on page 85](#) describes configuring VCS service groups for Oracle, the database, and the listener process in parallel mode. It describes configuring the agents within the service group.

[Chapter 6. “I/O Fencing Testing and Scenarios” on page 119](#) describes I/O fencing behavior in several scenarios.

[Chapter 7. “Using Storage Checkpoints and Storage Rollback” on page 129](#) describes using SFRAC storage checkpoints for backup and recovery of Oracle databases.

[Chapter 8. “Using Storage Mapping” on page 153](#) describes using SFRAC to map datafiles to physical devices.

[Chapter 9. “Adding and Removing SFRAC Nodes” on page 165](#) describes how to modify the number of nodes in a cluster.

[Chapter 10. “Upgrading to SFRAC 4.1” on page 191](#) describes how to upgrade VERITAS DBE/AC 3.5 for Oracle9i RAC to SFRAC 4.1.

[Chapter 11. “Uninstalling SFRAC 4.1” on page 205](#) describes how to uninstall and remove the components of SFRAC.

[Chapter 12. “Troubleshooting” on page 215](#) describes typical problems and recommended solutions. This section describes scripts that generate information about your systems used by VERITAS support.

[Appendix A. “Sample Configuration Files” on page 229](#) provides sample files for the `main.cf`, `types.cf`, `CFSTypes.cf`, `CVMTypes.cf`, `OracleTypes.cf`, and `VRTSWebAppType.cf`.

[Appendix B. “CVMCluster, CVMVxconfigd, CVMVolDg, and CFSSMount Agents” on page 243](#) provides details for various agents used in SFRAC configurations.

[Appendix C. “Tunable Kernel Driver Parameters” on page 253](#) describes tunable parameters for the LMX and VXFEN drivers.

[Appendix D. “Error Messages” on page 259](#) lists LMX and VXFEN messages you may encounter.

[Appendix E. “Creating a Starter Database” on page 265](#) contains procedures for creating starter databases using `dbca` or VERITAS scripts on raw shared volumes or in VxFS file systems.

[Appendix F. “Linking VERITAS Libraries to Oracle” on page 269](#) covers procedures to link Oracle binaries and SFRAC libraries using the `linkrac9i` command.

## Related Documentation

The Storage Foundation *for Oracle RAC* documentation set consists of the following manuals grouped according to the SFRAC component products:

<b>Title</b>	<b>File Name</b>
<b>VERITAS Storage Foundation <i>for Oracle RAC</i></b>	
<i>VERITAS Storage Foundation for Oracle RAC Installation and Configuration Guide</i>	sfrac_install.pdf
<i>VERITAS Storage Foundation for Oracle RAC Release Notes</i>	sfrac_notes.pdf
<b>VERITAS Cluster Server</b>	
<i>VERITAS Cluster Server Release Notes</i>	vcs_notes.pdf
<i>VERITAS Cluster Server Installation Guide</i>	vcs_install.pdf
<i>VERITAS Cluster Server User's Guide</i>	vcs_users.pdf
<i>VERITAS Cluster Server Agent Developer's Guide</i>	vcs_agent_dev.pdf
<i>VERITAS Cluster Server Bundled Agent Reference</i>	vcs_bundled_agents.pdf
<b>VERITAS Cluster Server Enterprise Agents</b>	
<i>VCS Enterprise Agent for Oracle Installation and Configuration Guide</i>	vcs_oracle_install.pdf
<i>VCS Enterprise Agent for DB2 Installation and Configuration Guide</i>	vcs_db2_install.pdf
<i>VCS Enterprise Agent for Sybase Installation and Configuration Guide</i>	vcs_sybase_install.pdf
<i>VCS Enterprise Agent for EMC SRDF Installation and Configuration Guide</i>	vcs_srdf_install.pdf
<i>VCS Enterprise Agent for IBM PPRC Installation and Configuration Guide</i>	vcs_pprc_install.pdf
<b>VERITAS Storage Foundation</b>	
<i>VERITAS Storage Foundation Release Notes</i>	sf_notes.pdf
<i>VERITAS Storage Foundation Installation Guide</i>	sf_install.pdf



<b>Title</b>	<b>File Name</b>
<i>VERITAS Storage Foundation Intelligent Storage Provisioning Administrator's Guide</i>	sf_isp_admin.pdf
<i>VERITAS Storage Foundation Cross-Platform Data Sharing Administrator's Guide</i>	sf_cds_admin.pdf
<b>VERITAS Volume Manager</b>	
<i>VERITAS Volume Manager Administrator's Guide</i>	vxvm_admin.pdf
<i>VERITAS Volume Manager Troubleshooting Guide</i>	vxvm_tshoot.pdf
<i>VERITAS Volume Manager Hardware Notes</i>	vxvm_hwnotes.pdf
<i>VERITAS FlashSnapPoint-In-Time Copy Solutions Administrator's Guide</i>	flashsnap_admin.pdf
<b>VERITAS File System</b>	
<i>VERITAS File System Administrator's Guide</i>	vxfs_admin.pdf
<i>VERITAS File System Programmer's Reference Guide</i>	vxfs_prg.pdf
<b>VERITAS Storage Foundation Database Administration</b>	
<i>VERITAS Storage Foundation Oracle Administrator's Guide</i>	sf_ora_admin.pdf
<i>VERITAS Storage Foundation DB2 Administrator's Guide</i>	sf_db2_admin.pdf
<i>VERITAS Storage Foundation Sybase Administrator's Guide</i>	sf_sybase_admin.pdf
<i>VERITAS Array Integration Layer Configuration Guide</i>	vail_config.pdf
<b>VERITAS FlashSnap Agent for Symmetrix</b>	
<i>VERITAS FlashSnap Agent for Symmetrix Release Notes</i>	vxfas_notes.pdf
<i>VERITAS FlashSnap Agent for Symmetrix Administrator's Guide</i>	vxfas_admin.pdf
<i>VERITAS FlashSnap Agent for Symmetrix Installation Guide</i>	vxfas_install.pdf
<i>VERITAS Cluster Server Agents for VERITAS FlashSnap Agent for Symmetrix</i>	vxfas_agent_install.pdf
<b>VERITAS Volume Replicator</b>	



---

<b>Title</b>	<b>File Name</b>
<i>VERITAS Volume Replicator Release Notes</i>	vvr_notes.pdf
<i>VERITAS Volume Replicator Installation Guide</i>	vvr_install.pdf
<i>VERITAS Volume Replicator Administrator's Guide</i>	vvr_admin.pdf
<i>VERITAS Volume Replicator Planning and Tuning Guide</i>	vvr_planning.pdf
<i>VERITAS Volume Replicator Web Console Administrator's Guide</i>	vvr_web_admin.pdf
<i>VERITAS Volume Replicator Advisor User's Guide</i>	vvr_advisor_users.pdf
<i>VERITAS Cluster Server Agents for VERITAS Volume Replicator Configuration Guide</i>	vvr_agents_config.pdf
<b>VERITAS Storage Foundation Cluster File System</b>	
<i>VERITAS Storage Foundation Cluster File System Release Notes</i>	sfdfs_notes.pdf
<i>VERITAS Storage Foundation Cluster File System Installation and Administration Guide</i>	sfdfs_install.pdf

---



## Conventions

The prompt indicating a command is issued as Oracle user is:

```
$ <command>
```

Here are the typographical conventions used throughout the manuals:

Convention	Usage	Example
monospace	Used for path names, commands, output, directory and file names, functions, and parameters.	Read tunables from the <code>/etc/vx/tunefstab</code> file. See the <code>ls(1)</code> manual page for more information.
<b>monospace (bold)</b>	Indicates user input.	<b># ls pubs</b> <b>C:\&gt; dir pubs</b>
<i>italic</i>	Identifies book titles, new terms, emphasized text, and variables replaced with a name or value.	See the <i>User's Guide</i> for details. The variable <code>system_name</code> indicates the system on which to enter the command.
<b>bold</b>	Depicts GUI objects, such as fields, list boxes, menu selections, etc. Also depicts GUI commands.	Enter your password in the <b>Password</b> field. Press <b>Return</b> .
<a href="#">blue text</a>	Indicates hypertext links.	See " <a href="#">Getting Help</a> " on page xxv.
#	Unix superuser prompt (all shells).	<b># cp /pubs/4.0/user_book /release_mgnt/4.0/archive</b>
C:\>	Windows user prompt.	<b>C:\&gt; copy \pubs\4.0\user_book c:\release_mgnt\4.0\archive</b>



## Getting Help

For technical assistance, visit <http://support.veritas.com> and select phone or email support. This site also provides access to resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and our customer email notification service. Use the Knowledge Base Search feature to access additional product information, including current and past releases of VERITAS documentation.

For license information, software updates and sales contacts, visit <https://my.veritas.com/productcenter/ContactVeritas.jsp>. For information on purchasing product documentation, visit <http://webstore.veritas.com>.

## Documentation Feedback

Your feedback on product documentation is important to us. Send suggestions for improvements and reports on errors or omissions to [clusteringdocs@veritas.com](mailto:clusteringdocs@veritas.com). Include the title and part number of the document (located in the lower left corner of the title page), and chapter and section titles of the text on which you are reporting. Our goal is to ensure customer satisfaction by providing effective, quality documentation. For assistance with topics other than documentation, visit <http://support.veritas.com>.





# Overview: Storage Foundation for Oracle RAC

1

This chapter describes the components of Storage Foundation for Oracle RAC and interaction between these components.

## What is RAC?

Real Application Clusters (RAC) is a parallel database environment that takes advantage of the processing power of multiple computers. A cluster comprises two or more computers, also known as nodes or servers. In RAC environments, all nodes concurrently run Oracle instances and execute transactions against the same database. RAC coordinates access to the shared data for each node to provide consistency and integrity. Each node adds its processing power to the cluster as a whole and can increase overall throughput or performance.

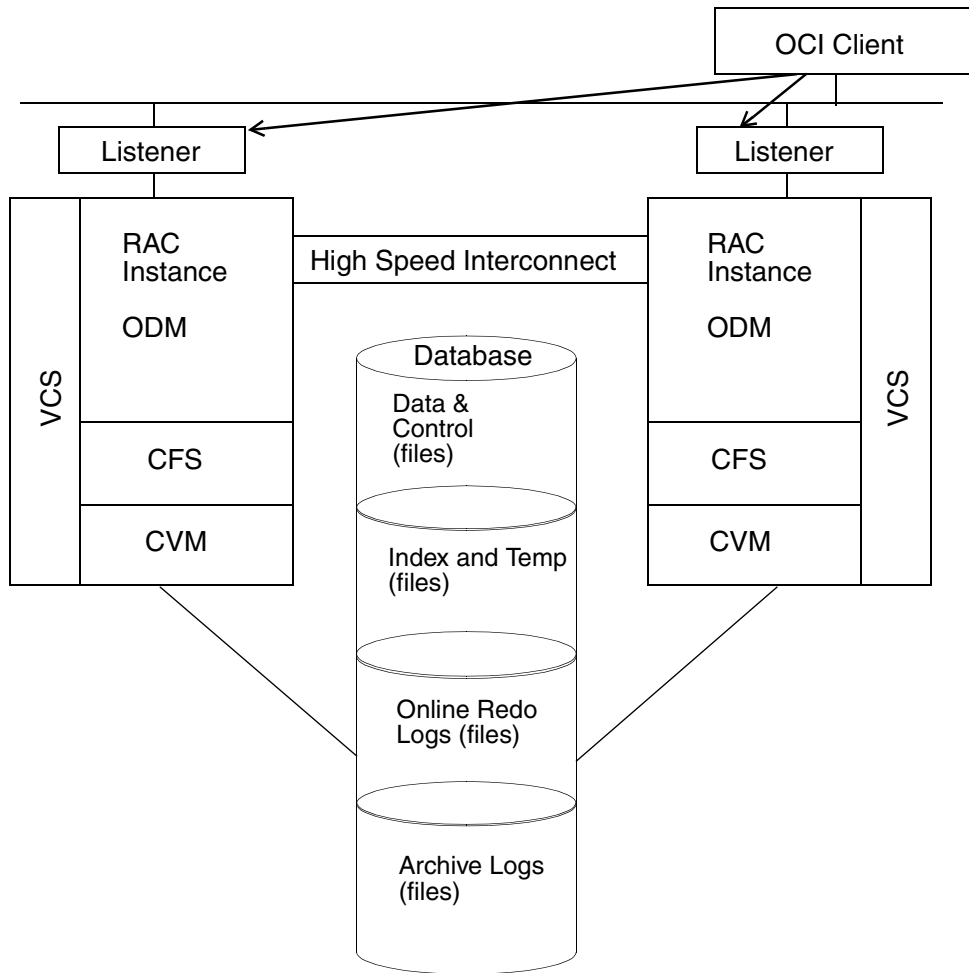
RAC serves a critical role in a robust solution for high availability. A properly configured RAC environment can tolerate failures with minimal downtime and interruption to users. If a node fails as clients access the same database on multiple nodes, clients attached to the failed node can reconnect to a surviving node and resume access. Recovery after failure in a RAC environment is far quicker than recovery for a failover database because another instance is already up and running. The recovery process involves applying outstanding redo log entries from the failed node.

## RAC Architecture

From a high-level perspective, RAC involves multiple Oracle instances accessing a single Oracle database and carrying out simultaneous transactions. An Oracle database is the physical data stored in tablespaces on disk. An Oracle instance is a set of processes and shared memory that provide access to the physical database. Specifically, the instance involves server processes acting on behalf of clients to read data into shared memory and make modifications to it, and background processes to write changed data to disk.

In traditional environments, only one instance accesses a database at a specific time. Oracle RAC enhances scalability and availability by enabling multiple instances to access the same database. This requires significant coordination between the instances to keep each instance's view of the data consistent.





## Operating System-Dependent Layer

Oracle RAC relies on several support services provided by VCS. The important features are cluster membership carried out by the cluster membership manager and inter-node communication. The actual implementation of these functions is described later in this chapter (see “[Storage Foundation for Oracle RAC OSD Layer Support](#)” on page 12).

### Veritas Cluster Server Membership Manager (VCSMM)

The Veritas Cluster Server Membership Manager provides a global view of the cluster. VCSMM determines cluster membership and enforces protection of data by preventing nodes outside of the cluster from corrupting stored data.

### Cluster Inter-Process Communication (VCSIPC)

RAC relies heavily on an underlying high-speed interprocess communication mechanism (VCSIPC). This mechanism defines the protocols and interfaces required for the RAC environment to transfer messages between instances.

### Lock Management and Cache Fusion

Lock management coordinates access for multiple instances to the same data to maintain data consistency and integrity. Oracle’s Cache Fusion provides memory-to-memory transfers of data blocks between RAC instances; these transfers are faster than transfers involving writing to and reading from disk across the cluster nodes.

### Shared Disk Subsystems

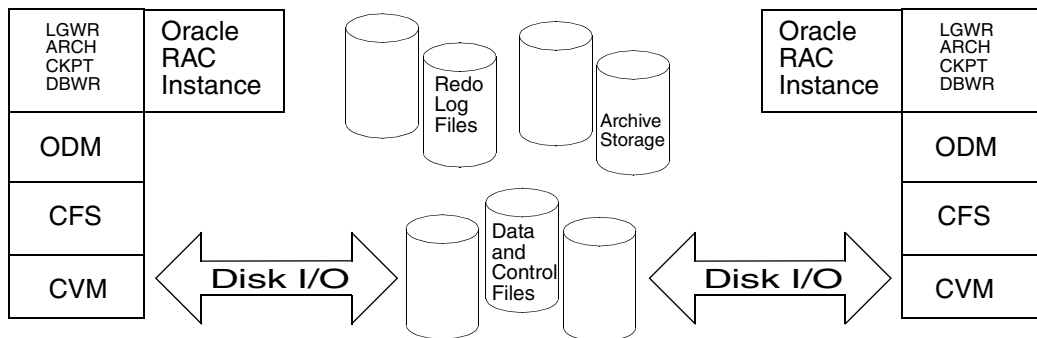
RAC requires all nodes have simultaneous access to database storage. This requirement gives multiple instances concurrent access to the same database.



## Storage Foundation *for Oracle RAC* Overview

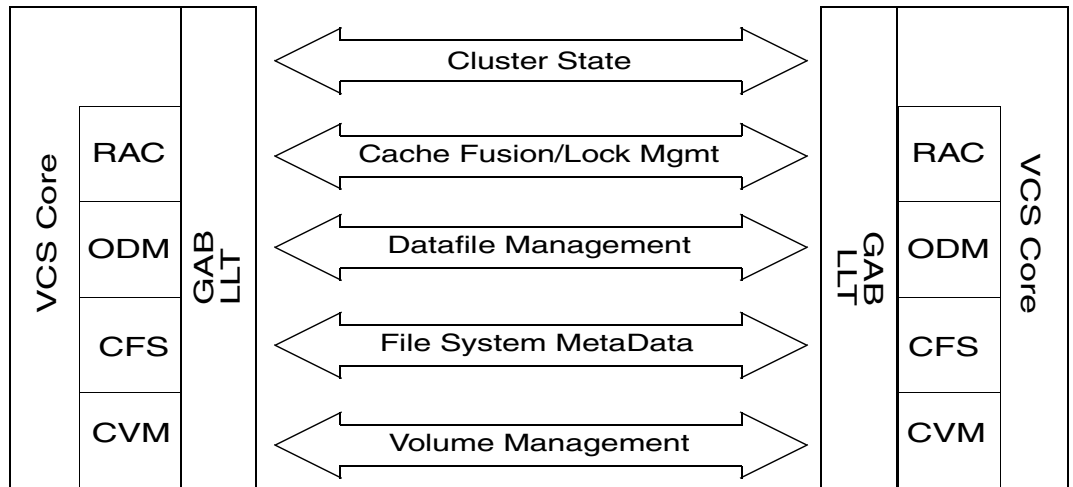
Storage Foundation *for Oracle RAC* provides a complete I/O and communications stack to support Oracle RAC. This software also monitors and manages the startup and shutdown of instances. The following section describes the overall data and communications flow of the Storage Foundation *for Oracle RAC* stack.

### Data Stack Overview



The diagram above details the overall data flow from an instance running on a server to the shared storage. The various Oracle processes making up an instance (such as DB Writer, Log Writer, Checkpoint, and Archiver) read and write data to storage via the I/O stack shown in the diagram. Oracle communicates via the Oracle Disk Manager (ODM) interface to the VERITAS Cluster File System (CFS), which in turn accesses the storage via the VERITAS Cluster Volume Manager (CVM). Each of these components in the data stack is described in this chapter.

## Communications Stack Overview



The diagram above shows the data and communications stacks. Each of the components in the data stack requires communications with its peer on other systems to function properly. RAC instances must communicate to coordinate protection of data blocks in the database. ODM processes must communicate to coordinate data file protection and access across the cluster. CFS coordinates metadata updates for file systems, and CVM coordinates the status of logical volumes and distribution of volume metadata across the cluster. VERITAS Cluster Server (VCS) controls starting and stopping of components in the Storage Foundation *for Oracle RAC* stack and provides monitoring and notification on failure. VCS must communicate the status of its resources on each cluster node. For the entire system to work, each layer must communicate properly.

The diagram also shows Low Latency Transport (LLT) and Group Membership Services/Atomic Broadcast (GAB), which make up the communications package central to the operation of Storage Foundation *for Oracle RAC*. During an operational steady state, the only significant traffic through LLT and GAB results from Lock Management and Cache Fusion, while the traffic for the other data is relatively sparse.



## Storage Foundation *for Oracle RAC* Communications

The communications package for Storage Foundation *for Oracle RAC* consist of LLT and GAB.

### Low Latency Transport (LLT)

LLT provides fast, kernel-to-kernel communications, and monitors network connections. LLT serves as a high performance replacement for the IP stack and runs directly on top of the Data Link Protocol Interface (DLPI) layer. The use of LLT rather than IP removes latency and overhead associated with the IP stack. LLT has several major functions.

### VCSIPC

RAC Inter-Process Communications (VCSIPC) uses the VCSIPC shared library for inter-process communication. VCSIPC leverages all of LLT's features and uses LMX, an LLT multiplexer, to provide fast data transfer between Oracle processes on different cluster nodes.

### Traffic Distribution

LLT distributes (load-balances) inter-node communication across all available private network links. All cluster communications are evenly distributed across as many as eight network links for performance and fault resilience. If a link fails, LLT redirects traffic to the remaining links.

### Heartbeat

LLT is responsible for sending and receiving heartbeat traffic over network links. This heartbeat is used by the Group Membership Services function of GAB to determine cluster membership.





## Group Membership Services/Atomic Broadcast (GAB)

GAB is responsible for Cluster Membership and Cluster Communications as described below.

### Cluster Membership

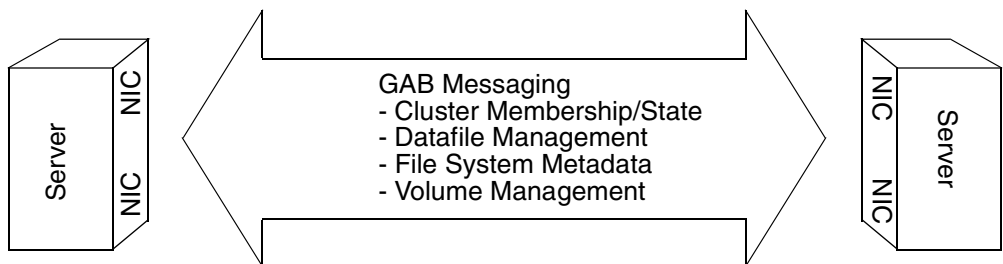
All nodes in a distributed system, such as Storage Foundation *for Oracle RAC*, must be aware of each node currently participating in the cluster. Nodes can leave or join the cluster because of shutting down, starting up, rebooting, powering off, or faulting processes.

Storage Foundation *for Oracle RAC* uses LLT heartbeats to determine cluster membership. When systems no longer receive heartbeats from a peer for a predetermined interval, a protocol excludes the peer from the current membership. When systems start receiving heartbeats from a peer that is not part of the membership, a protocol enables the peer to join the current membership.

Storage Foundation *for Oracle RAC* delivers the new membership information consistently to all nodes and initiates actions specific to each module. For example, if a node faults, CVM initiates volume recovery and CFS performs a parallel file system check.

### Cluster Communications

The second function of GAB is to provide reliable cluster communications for Storage Foundation *for Oracle RAC* modules. GAB provides guaranteed delivery of point-to-point messages and broadcast messages to all nodes. Point-to-point messaging involves sending and acknowledging the message. Atomic-broadcast messaging ensures all systems within the cluster receive all messages. If a failure occurs while transmitting a broadcast message, GAB ensures all systems have the same information after recovery.



## Cluster Volume Manager (CVM)

CVM is an extension of VERITAS Volume Manager (VxVM), the industry standard storage virtualization platform. CVM extends the concepts of VxVM across multiple nodes. Each node recognizes the same logical volume layout and state of all volume resources.

Use standard VxVM commands from one node in the cluster to manage all storage. All other nodes immediately recognize any changes in disk group and volume configuration with no interaction. CVM supports performance-enhancing capabilities such as striping, mirroring, and mirror break-off (snapshot) for off-host backup.

### CVM Architecture

CVM is designed with a “master/slave” architecture. One node in the cluster acts as the configuration master for logical volume management, and all other nodes are slaves. Any node can take over as master if the existing master fails. The CVM master exists on a per-cluster basis and uses GAB and LLT to transport its configuration data.

Since CVM is an extension of VxVM, it operates in a similar fashion. The volume manager configuration daemon, `vxconfigd`, maintains the configuration of logical volumes. This daemon handles changes to the volumes by updating the operating system at the kernel level. For example, if a mirror of a volume fails, the mirror detaches from the volume and `vxconfigd` determines the proper course of action, updates the new volume layout, and informs the kernel of a new volume layout. CVM extends this behavior across multiple nodes and propagates changes to a volume to the master `vxconfigd`. The `vxconfigd` process on the master pushes these changes out to slave `vxconfigd` processes, each of which updates the local kernel.

CVM does not impose any write locking between nodes. Each node is free to update any area of the storage. All data integrity is the responsibility of the upper application. From an application perspective, standalone systems access logical volumes in the same way as CVM systems.

CVM imposes a “Uniform Shared Storage” model. All systems must be connected to the same disk sets for a given disk group. Any system unable to see the entire set of physical disks for a given disk group cannot import the group. If a node loses contact with a specific disk, CVM excludes the node from participating in the use of that disk.

## Cluster File System

CFS is an extension of the industry standard VERITAS File System (VxFS). CFS enables you to simultaneously mount the same file system on multiple nodes. Unlike other clustered file systems, CFS is a true SAN file system; all I/O occurs over the storage area network. Coordination between nodes occurs through messages across the cluster interconnects.

### CFS Architecture

CFS incorporates a primary/secondary architecture. Though any node can initiate an operation to create, delete, or resize data, the master node carries out the actual operation. Since CFS is an extension of VxFS, it operates in a similar fashion and caches metadata and data in memory (typically called buffer cache or vnode cache). A distributed locking mechanism called Global Lock Manager (GLM) provides metadata and cache coherency across multiple nodes. GLM offers a way to ensure all nodes have a consistent view of the file system. When any node wishes to read data, GLM requests a shared lock. If another node wishes to write to the same area of the file system, it must request an exclusive lock. GLM revokes all shared locks before granting the exclusive lock and informs reading nodes that their data is no longer valid.

### CFS Usage in Storage Foundation *for Oracle RAC*

Storage Foundation *for Oracle RAC* uses CFS to manage a file system in a large database environment. Oracle uses the ODM interface to access data files stored on CFS file systems. This process essentially bypasses the file system buffer and locking for data; only Oracle (and not GLM) buffers data and coordinates writing to files. A single point of locking and buffering ensures maximum performance.

## Oracle Disk Manager

The Oracle Disk Manager (ODM) is a standard API specified by Oracle for database I/O. When Oracle performs a write operation, it calls the `odm_io` function. ODM improves both performance and manageability of the file system.

The VERITAS implementation of ODM improves performance by providing direct access for the database to the underlying storage without passing through the actual file system interface. The resulting performance is equivalent to performance when using raw devices. The administrator sees the storage as easy-to-manage file systems that support the resizing of data files while in use.



## ODM Clustering Extensions

All ODM features can operate in a cluster environment. Nodes communicate with each other before performing any operation that could potentially affect another node. For example, before creating a new data file with a specific name, ODM checks with other nodes to see if the file name is already in use.

## VERITAS Cluster Server

VCS functions as the director of operations for Storage Foundation *for Oracle RAC*. VCS controls startup and shutdown of the component layers of RAC. In the Storage Foundation *for Oracle RAC* configuration, the RAC service groups run as parallel service groups. VCS does not attempt to migrate a failed service group, but the software enables you to configure it to restart on failure. VCS also notifies users of any failures.

Storage Foundation *for Oracle RAC* provides specific agents for VCS to operate in a Storage Foundation *for Oracle RAC* environment, including CVM, CFS and Oracle.

## VCS Architecture

VCS communicates the status of resources running on each system to all systems in the cluster. The High Availability Daemon (HAD) is the main VCS daemon running on each system. HAD collects all information about resources running on the local system and forwards it to all other systems in the cluster. This daemon also receives information from all other cluster members to update its own view of the cluster.

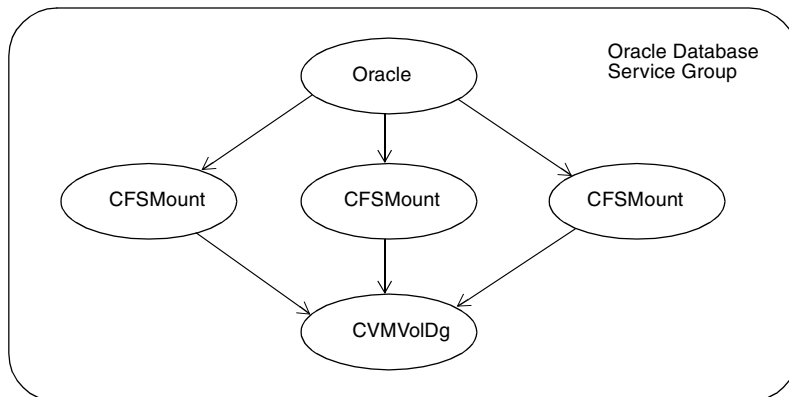
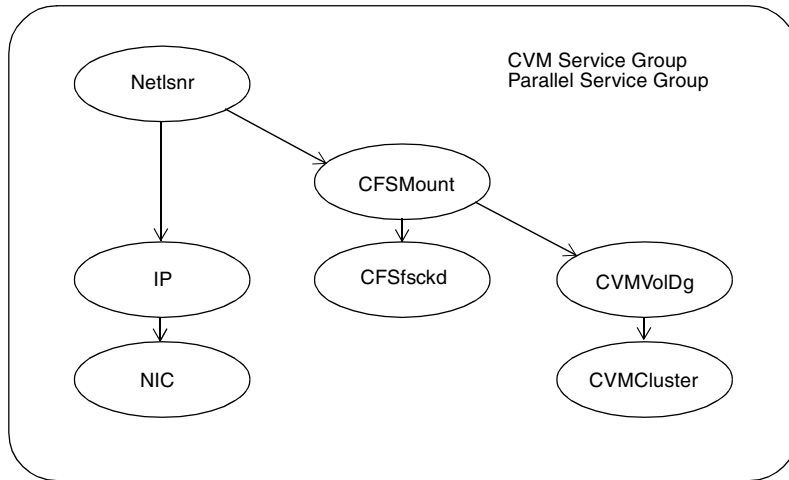
Each type of resource supported in a cluster is associated with an agent. An agent is an installed program designed to control a particular resource type. Each system runs necessary agents to monitor and manage resources configured to run on that node. The agents communicate with HAD on the node. HAD distributes its view of resources on the local node to other nodes in the cluster using GAB and LLT.

## Storage Foundation *for Oracle RAC* Service Groups

Storage Foundation *for Oracle RAC* uses parallel service groups to support RAC. A single CVM group exists per server. This group has the CVM resource and the necessary resources to support CFS. Users can modify this group to contain all common components needed by Oracle to support RAC. These components include a shared ORACLE\_HOME directory and the Oracle Net Services process (LISTENER).



Users create a service group for each Oracle database on shared storage, specifying the supporting CVM and CFS resources.



## Storage Foundation *for Oracle RAC* OSD Layer Support

Storage Foundation *for Oracle RAC* OSD layer support includes the VCSMM and VCSIPC components.

### Cluster Membership (VCSMM)

Oracle provides an API called `skgxn` (system kernel generic interface node membership) to obtain information on membership. Storage Foundation *for Oracle RAC* implements this API as a library linked to Oracle after you install RAC. The `skgxn` library makes `ioctl` calls to a kernel module for membership information.

This module is known as VCSMM (VCS membership module). Oracle 9iR2 uses the shared library `libskgxn9.so`, which is dynamically linked with the Oracle binary at run-time.

### Inter-Process Communications (VCSIPC)

To coordinate access to a single database by multiple instances, Oracle uses extensive communications between nodes and instances. Specifically, Oracle uses Inter-Process Communications (VCSIPC) for lock management traffic and Cache Fusion. Storage Foundation *for Oracle RAC* uses LLT to support VCSIPC in a cluster and leverages its high-performance and fault-resilient capabilities.

Oracle has an API for VCSIPC, System Kernel Generic Interface Inter-Process Communications (`skgxp`), that isolates Oracle from the underlying transport mechanism. As Oracle conducts communication between processes, it does not need to know how data moves between systems.

Storage Foundation *for Oracle RAC* provides a shared library that is dynamically loaded by Oracle at run-time to implement the `skgxp` functionality. This module communicates with the LLT Multiplexer (LMX) via `ioctl` calls. The LMX module is a kernel module designed to receive communications from the `skgxp` module and pass them on to the appropriate process on other nodes. The LMX module “multiplexes” communications between multiple-related processes on the cluster nodes. LMX leverages all features of LLT, including load balancing and fault resilience.



## I/O Fencing

I/O fencing is a feature within a kernel module of Storage Foundation *for Oracle RAC* designed to guarantee data integrity. This feature works even in the case of faulty cluster communications causing a split-brain condition.

### Understanding Split Brain and the Need for I/O Fencing

Split brain is an issue faced by all cluster solutions. To provide high availability, the cluster must be capable of taking corrective action when a node fails. In this situation, Storage Foundation *for Oracle RAC* configures CVM, CFS, and RAC to reflect the altered membership.

Problems arise when the mechanism that detects the failure breaks down because symptoms appear identical to those of a failed node. For example, if a system in a two-node cluster fails, the system stops sending heartbeats over the private interconnects and the remaining node takes corrective action. However, the failure of private interconnects (instead of the actual nodes) would present identical symptoms and cause each node to determine its peer has departed. This situation typically results in data corruption because both nodes attempt to take control of data storage in an uncoordinated manner.

In addition to a broken set of private networks, other scenarios can cause this situation. If a system is so busy that it appears to stop responding or “hang,” the other nodes could declare it as dead. This declaration may also occur for nodes using hardware that supports a “break” and “resume” function. When a node drops to PROM level with a break and subsequently resumes operations, the other nodes may declare the system dead even though the system later returns and begins write operations.

Storage Foundation *for Oracle RAC* uses a technology called I/O fencing to remove the risk associated with split brain. I/O fencing blocks access to storage from specific nodes; even a node that is alive is unable to cause damage.

### SCSI-3 Persistent Reservations

Storage Foundation *for Oracle RAC* uses SCSI-3 Persistent Reservations (SCSI-3 PR). SCSI-3 PR is designed to resolve the issues of using SCSI reservations in a clustered SAN environment. SCSI-3 PR enables access by multiple nodes to a device and simultaneously blocks access for other nodes. SCSI-3 PR reservations are persistent across SCSI bus resets and SCSI-3 PR supports multiple paths from a host to a disk. In contrast, only one host can use SCSI-2 reservations with one path. If the need arises to block access because of data integrity concerns, only one host and one path remain active. The requirements for larger clusters, with multiple nodes reading and writing to storage in a controlled manner, make SCSI-2 reservations obsolete.



SCSI-3 PR uses a concept of registration and reservation. Each system registers its own “key” with a SCSI-3 device. Multiple systems registering keys form a membership and establish a reservation, typically set to “Write Exclusive Registrants Only” (WERO). This setting enables only registered systems to perform write operations. For a given disk, only one reservation can exist amidst numerous registrations.

With SCSI-3 PR technology, blocking write access is as simple as removing a registration from a device. Only registered members can “eject” the registration of another member. A member wishing to eject another member issues a “preempt and abort” command. Ejecting a node is final and “atomic;” an ejected node cannot eject another node.

In Storage Foundation *for Oracle RAC*, a node registers the same key for all paths to the device. A single preempt and abort command ejects a node from all paths to the storage device.

To summarize, the SCSI-3 PR specification describes the method to control access to disks with the registration and reservation mechanism. The method to determine who can register with a disk and when a registered member should eject another node is specific to the implementation. The following paragraphs describe I/O fencing concepts and implementation for Storage Foundation *for Oracle RAC*.

## I/O Fencing Components

I/O Fencing, or simply fencing, gives write access to members of the active cluster and blocks access to non-members. The physical components of I/O fencing in Storage Foundation *for Oracle RAC* are *coordinator* disks and *data* disks. Each component has a unique purpose and uses different physical disk devices.

### Data Disks

Data disks are standard disk devices for data storage and are either physical disks or RAID Logical Units (LUNs). These disks must support SCSI-3 PR and are part of standard VxVM/CVM disk groups.

CVM is responsible for fencing data disks on a disk group basis. VxVM enables I/O fencing and provides additional features. Disks added to a disk group are automatically fenced, as are new paths discovered to a device.



## Coordinator Disks

Coordinator disks are three (or an odd number greater than three) standard disks or LUNs set aside for I/O fencing during cluster reconfiguration. These disks provide a lock mechanism to determine which nodes get to fence off data drives from other nodes. A node must eject a peer from the coordinator disks before it can fence the peer from the data drives. This concept of racing for control of the coordinator disks to gain the ability to fence data disks is key to understanding prevention of split brain through fencing.

Coordinator disks do not serve any other purpose in the Storage Foundation *for Oracle RAC* configuration. Users cannot store data on these disks or include the disks in a disk group for user data. The coordinator disks can be any three disks that support SCSI-3 PR. VERITAS recommends using the smallest possible LUNs for coordinator disks. Since coordinator disks do not store any data, cluster nodes need only register with them and do not need to reserve them.

## I/O Fencing Operations

I/O fencing, provided by the kernel-based fencing module (VXFEN), performs identically on node failures and communications failures. When the fencing module on a node is informed of a change in cluster membership by the GAB module, it immediately begins the fencing operation. The node attempts to eject the key for departed node(s) from the coordinator disks using the preempt and abort command. When the node successfully ejects the departed nodes from the coordinator disks, it ejects the departed nodes from the data disks. In a split-brain scenario, both sides of the split would race for control of the coordinator disks. The side winning the majority of the coordinator disks wins the race and fences the loser. The loser then panics and reboots the system.





# Preparing to Install Storage Foundation for Oracle RAC

---

2

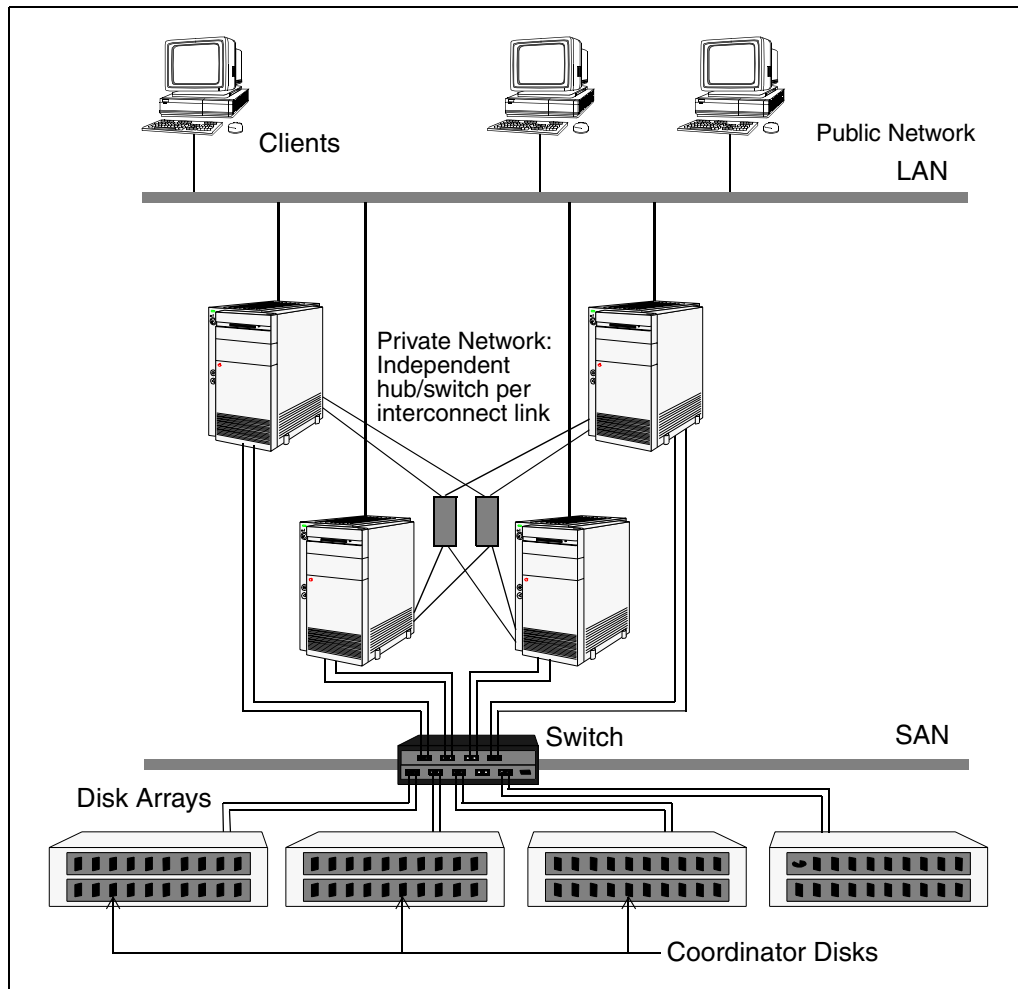
This chapter provides an overall view of the product and phases of the installation and configuration for VERITAS Storage Foundation *for Oracle RAC* (version 4.1). Refer to the information below for hardware and software requirements and information you need to have on hand when interacting with the installation utilities.

## High-Level View of Storage Foundation for Oracle RAC

This section introduces a high-level view of VERITAS Storage Foundation *for Oracle RAC* and tasks to install SFRAC. If you install SFRAC with Oracle9i and create a database, the cluster has the following characteristics:

- ◆ Nodes connected by at least two VCS private network links using 100 Base T or Gigabit Ethernet controllers on each system. For two-node clusters, cross-over Ethernet cables are acceptable. For maximum performance, VERITAS recommends the use of switches over hubs. In either case, use a minimum of two switches or hubs to provide necessary redundancy.
- ◆ Nodes connected to shared storage devices via Fibre Channel switch. VERITAS does not support the use of shared SCSI with this product. For a complete list of supported Fibre Channel storage devices, see the current hardware compatibility list on the VERITAS Support Web site: <http://support.veritas.com>.
- ◆ Nodes running VERITAS Cluster Server (VCS), VERITAS Volume Manager with cluster features (CVM), VERITAS File System with cluster features (CFS), and Storage Foundation *for Oracle RAC* agents and components, including I/O fencing.
- ◆ Oracle RAC software that resides either on the shared storage accessible to all nodes or locally on each node.
- ◆ Oracle RAC database configured on the shared storage (cluster file system or raw volumes) is available to each node.
- ◆ Resources required by Oracle RAC; these resources runs in parallel on each node.





# Phases of Installation and Configuration

The basic installation and configuration process for an SFRAC cluster involves five phases. In addition to these phases, refer to this guide for details on upgrading a cluster with SFRAC 3.5 to SFRAC 4.1.

## Phase One: Preparing and Configuring Hardware

Prior to installing *Storage Foundation for Oracle RAC*, set up the basic hardware. Details about supported hardware are on the VERITAS support web site: <http://support.veritas.com>.

Each cluster requires:

- ✓ Two or four nodes connected to the public network that are up and running the B.11.23.0409 HP-UX Base OS Auxiliary version of the HP operating system.
- ✓ Two or more 100BaseT or Gigabit Ethernet links directly linking each node to the other node to form a private network that handles direct system-to-system communication. VERITAS recommends switches for the private network links. These links must be of the same type; you cannot mix 100BaseT and Gigabit.
- ✓ Each node can access shared storage through a switch. This storage must support SCSI-3 persistent reservations.

## Phase Two: Installing SFRAC and Configuring its Components

“[Installing SFRAC 4.1 and Configuring Components](#)” on page 27 describes the procedures to install *Storage Foundation for Oracle RAC*. Verify system requirements for installation described in “[Requirements for Installing Components](#)” on page 21. Installing *Storage Foundation for Oracle RAC* involves:

- ✓ Running the `installer` utility or `installsfrac` and `installsfrac -configure` scripts to install and configure VCS, Volume Manager enabled for clusters (CVM), and VERITAS File System enabled for clusters (CFS). Included with this installation are other *Storage Foundation for Oracle RAC* modules and the VCS enterprise agent for Oracle. The script is interactive.
- ✓ Setting up I/O fencing. This includes:
  - ◆ Running the `vxfcntlshdw` script to verify shared storage can support I/O fencing.
  - ◆ Setting up coordinator disks for the I/O fencing feature into a disk group.
  - ◆ Editing the configuration file to set the `UseFence=SCSI3` attribute.
  - ◆ Shutting down and restarting the node.



## Phase Three: Installing Oracle9i

After installing and configuring components of Storage Foundation *for Oracle RAC*, install Oracle9i Release 2. Refer to [“Installing Oracle9i Software”](#) on page 65 for instructions on installing this software.

## Phase Four: Creating the Database

Numerous procedures exist for creating a database. This guide describes how to create a starter database on raw volumes within a VxVM disk group or on a VERITAS cluster file system using the Oracle `dbca` utility. These procedures are provided in case database creation tools are not available. Refer to [“Creating a Starter Database”](#) on page 265.

## Phase Five: Setting up VCS to Manage RAC Resources

Storage Foundation for Oracle RAC provides the capability to completely automate the RAC environment. This capability ranges from enabling automatic control of the entire database environment to having VCS mount cluster file systems or enable CVM and CFS daemons. The user or DBA is free to choose the level of control and automation.

VCS uses the `main.cf` configuration file (see [“Sample Configuration Files”](#) on page 229) to manage resources in the cluster. The SFRAC installation process creates a basic VCS configuration file. After installing Oracle and creating the database, you can modify the `main.cf` file on one of the cluster nodes to reflect the new resources and their configuration. The examples in [“Configuring VCS Service Groups for Oracle9i”](#) describe:

- ✓ Editing the CVM service group to define the location of the Oracle binaries, listener process for Oracle9i, and resources.
- ✓ Creating an Oracle service group to define the database and required resources.



## Requirements for Installing Components

This section describes the prerequisites to install VERITAS Storage Foundation *for Oracle RAC* software.

- ◆ For a list of supported hardware for Storage Foundation *for Oracle RAC*, see the VERITAS Technical Support website: <http://support.veritas.com>.
- ◆ For the latest information on updates, patches, and software issues regarding this release, see the following TechNote on the VERITAS Technical Support website: <http://support.veritas.com/docs/275787>.

Refer to the *VERITAS Storage Foundation 4.1 for Oracle RAC Release Notes* for additional details.

## License Keys

VERITAS Storage Foundation *for Oracle RAC* is a licensed software product. The installation program prompts you for a license key for each node. You cannot use this product until you have completed the licensing process. Use one of the following methods described in the upcoming sections to obtain a valid license key.

### Using the VERITAS vLicense™ Web Site to Obtain a License Key

You can obtain your license key most efficiently using the VERITAS vLicense Web site. The License Key Request Form has all the information to establish a user account on vLicense and generate your license key. The License Key Request Form is a one-page insert included with the disc in your product package. You must have this form to obtain a software license key for your VERITAS product.

---

**Note** *Do not discard the License Key Request Form. If you misplace the form, email [license@veritas.com](mailto:license@veritas.com).*

---

The License Key Request Form contains information unique to your VERITAS software purchase. To obtain your software license key, refer to the following information on the form:

- ◆ Your VERITAS customer number
- ◆ Your order number
- ◆ Your serial number

Follow the appropriate instructions on the vLicense Web site to obtain your license key depending on whether you are a new or previous user of vLicense. Access the site at <http://vlicense.veritas.com> and log in or create a new login. After following the instructions and receiving the generated license key, you can proceed with installation.



## Faxing the License Key Request Form to Obtain a License Key

If you do not have Internet access, fax the License Key Request Form to VERITAS. Be aware that faxing the form generally requires several business days to process in order to provide a license key. Before faxing the form, sign and date it in the appropriate spaces. Fax it to the number shown on the form.

## Oracle RAC Versions

Storage Foundation *for Oracle RAC* supports systems running Oracle9i Release 2.

## HP Operating Systems

All nodes in the cluster must run the same level and release of the HP-UX operating system. SFRAC 4.1 requires the September 2004 (also called AR0904) HP-UX 11i Version 2.0 (B11.23) release for HP 9000/Itanium servers, 64-bit. If necessary, upgrade the operating system to this version or a later version.

To identify the Operating Environment (OE) currently installed on your system, use the `swlist` command. For example:

```
# swlist | grep HPUX
HPUX11i-OE-MC                B.11.23.0409    HP-UX Mission
    Critical Operating Environment Component
HPUXBaseAux                  B.11.23.0409    HP-UX Base OS
    Auxiliary
HPUXBaseOS                   B.11.23         HP-UX Base OS
OnlineDiag                   B.11.23.03.16   HPUX 11.23 Support
    Tools Bundle, Sep 2004
```

The output of this command includes a line that identifies the installed OE. For example:

```
HPUX11i-OE-MC                B.11.23.0409    HP-UX Mission
    Critical Operating Environment Component
```





## Operating Environments

HP-UX 11i OEs are bundles of key applications available to users to help them install a complete operating system in a single-pass process. VERITAS supports SFRAC for HP-UX on the following OEs:

- ◆ HP-UX11i-OE
- ◆ HP-UX 11i v2 Foundation OE
- ◆ HP-UX11i-OE-Ent
- ◆ HP-UX 11i v2 Enterprise OE
- ◆ HP-UX11i-OE-MC
- ◆ HP-UX 11i v2 Mission Critical OE

## HP Patches

Install the following HP packages and patches on each node before installing SFRAC:

- ◆ DiskQuota-Enh
- ◆ FSCmdsEnh
- ◆ FSLibEnh
- ◆ PHCO\_32385
- ◆ PHCO\_32387
- ◆ PHCO\_32388
- ◆ PHCO\_32389
- ◆ PHCO\_32390
- ◆ PHCO\_32391
- ◆ PHCO\_32392
- ◆ PHCO\_32393
- ◆ PHCO\_32488
- ◆ PHCO\_32523
- ◆ PHCO\_32524
- ◆ PHCO\_32551
- ◆ PHCO\_32552
- ◆ PHCO\_32596



- ◆ PHCO\_32608
- ◆ PHCO\_32609
- ◆ PHCO\_32610
- ◆ PHCO\_32611
- ◆ PHKL\_32272
- ◆ PHKL\_32425
- ◆ PHKL\_32430
- ◆ PHKL\_32431

## Disk Space (Local)

Each node requires at least the following local disk space:

- ◆ For the installation of SFRAC without Oracle9i: 7 GB (recommended)
- ◆ For the installation of Oracle9i on the local disk: *approximately* 20 GB
- ◆ For the duration of the installation of SFRAC and Oracle9i: a minimum of 1 GB in the `/tmp` directory. This disk space is required only during the installation.

## Synchronized Cluster Nodes

VERITAS recommends all cluster nodes have the same time. If you do not run the Network Time Protocol (NTP) daemon, make sure to synchronize the time settings on each node.

## System-to-System Communication

If you configured `ssh` (SSH client) for the cluster nodes, the installation program can install SFRAC as long as `ssh` commands between nodes can execute without password prompting and confirmation.

If you did not configure `ssh`, enable each node to have remote `remsh` access to the other nodes during installation and disk verification. On each node, placing a `"++"` character in the first line of the `/.rhosts` file gives remote access to the system running the install program. You can limit the remote access to specific nodes. Refer to the manual page for the `/.rhosts` file for more information.

Remove the remote `remsh` access permissions after the installation and disk verification process.



## Removing Pre-existing License Keys

1. To view license key files currently installed on a node, type:

```
# cd /opt/VRTS/bin
# ./vxlicrep
```

The output lists the license keys and information about their respective products.

2. Navigate to the directory containing the license key files and list the files:

```
# cd /etc/vx/licenses/lic
# ls -a
```

3. Move the license key files to another location for future reference. For example, in the directory containing the license key files, create a sub-directory and move the files to that new location:

```
# mkdir OLD
# mv * OLD
```



## Information Required to Install SFRAC

The installation program prompts you for information about the cluster where you are installing VERITAS Storage Foundation *for Oracle RAC*. Keep the following information at hand:

### Cluster Configuration

- ✓ The name of the cluster (must begin with a letter of the alphabet (a-z, A-Z)).
- ✓ The ID number of the cluster from 0-255 (each cluster within a site requires a unique ID).
- ✓ The host names of the nodes in the cluster.
- ✓ The device names for the private network links.
- ✓ A valid license key for VERITAS Storage Foundation *for Oracle RAC*.

### Web-based Cluster Manager (Optional)

- ✓ The name of the public NIC for VCS Cluster Manager (Web Console).
- ✓ The virtual IP address of the NIC for Cluster Manager.
- ✓ The net mask for the virtual IP address.
- ✓ NetworkHosts IP addresses to check the connection.

### SMTP/SNMP (Optional)

- ✓ Domain-based address of the SMTP server.
- ✓ Email address of SMTP-notification recipients.
- ✓ Severity level of events for SMTP notification.
- ✓ System name for the SNMP console.
- ✓ Severity level of events for SNMP notification.

---

**Note** The installation of VCS using the `installer` utility is similar to the installation described in the *VERITAS Cluster Server Installation Guide*. Refer to that guide for more extensive details on installing VCS.

---



# Installing SFRAC 4.1 and Configuring Components

---

3

This chapter describes how to install and configure the components of the VERITAS Storage Foundation *for Oracle RAC* in a cluster. The main tasks include:

- ✓ Running the installation script to add packages to each cluster node
- ✓ Running the installation script to perform basic configuration.
- ✓ Verifying the disks intended for shared storage support SCSI-3 persistent reservations.
- ✓ Setting up the “coordinator” disk group.
- ✓ Reviewing the configuration of the software components.

## Installing SFRAC 4.1 Packages

Install VERITAS Storage Foundation *for Oracle RAC* on clusters of up to four nodes. The procedure shows a sample installation on four nodes, `slpas06`, `slpas07`, `slpas08`, and `slpas09`.

Installing SFRAC involves an installation phase and configuration phase. Use either of the following tools to start the installation phase:

- ◆ The VERITAS product installer (see [“Using the VERITAS Product Installer to Install SFRAC”](#) on page 30). This tool offers a high-level approach to installing multiple VERITAS products.
- ◆ The `installsfrac` script (see [“Using the installsfrac Script to Install SFRAC”](#) on page 31). This tool offers a direct approach to installing SFRAC.

---

**Note** If you obtained SFRAC from an electronic download site, you must use the `installsfrac` script instead of the VERITAS product installer.

---

The process installs the following components on each node: VERITAS Cluster Server (VCS), VERITAS Volume Manager (VxVM), and VERITAS File System (VxFS). Refer to [“Requirements for Installing Components”](#) on page 21 for general prerequisites.



## Setting the PATH Variable

The installation and other commands are located in various directories. If necessary, add these directories to your PATH environment variable on each system:

If you use the Bourne Shell (sh or ksh):

```
# PATH=/usr/sbin:/sbin:/usr/bin:/usr/lib/vxvm/bin:\
/opt/VRTSvxfs/cfs/bin:/opt/VRTSvcs/bin:/opt/VRTS/bin:\
/opt/VRTSvcs/rac/bin:/opt/VRTSob/bin:$PATH; export PATH
```

If you use the C Shell (csh or tcsh):

```
# setenv PATH /usr/sbin:/sbin:/usr/bin:/usr/lib/vxvm/bin:\
/opt/VRTSvxfs/cfs/bin:/opt/VRTSvcs/bin:/opt/VRTS/bin:\
/opt/VRTSvcs/rac/bin:/opt/VRTSob/bin:$PATH
```

---

**Note** Do not define paths for the root user to a cluster file system in the `LD_LIBRARY_PATH` variable. You can define `$ORACLE_HOME/lib` in `LD_LIBRARY_PATH` for the oracle user.

---

The path defined as `/opt/VRTSob/bin` is optional unless you choose to install VERITAS Enterprise Administrator.

## Setting the MANPATH Variable

Set the MANPATH variable to enable viewing manual pages:

If you use the Bourne Shell (sh or ksh):

```
# MANPATH=/usr/share/man:/opt/VRTS/man; export MANPATH
```

If you use the C Shell (csh or tcsh):

```
# setenv MANPATH /usr/share/man:/opt/VRTS/man
```

## Mounting the VERITAS SFRAC Disc

1. Log in as `root` user to a cluster node or a remote node in the same subnet as the cluster nodes.
2. Insert the VERITAS software disc into the CD drive and determine the appropriate CD-ROM device:

```
# ioscan -fn -C disk
```

For example, the listing may indicate the block device of the CD-ROM drive is `/dev/dsk/c0t0d0`; make a note of the device file as it applies to your system.

3. If necessary, create a directory under which to mount the software disc and mount the disc:

```
# mkdir -p /cdrom
# mount -F cdfs /dev/dsk/c0t0d0 /cdrom
```

---

**Note** VERITAS advises using the `-F cdfs` option for the mount command.

---

Alternately, you can mount the disc using Portable File System (PFS):

- ◆ Check whether the PFS daemons are running:

```
# ps -ef | grep pfs_mountd
```

- ◆ If the daemons are running, look for output resembling the example below:

```
# ps -ef | grep pfs
root 17796 17795 0 Mar 23 ? 5:51 pfsd.rpc
root 17794 17793 0 Mar 23 ? 0:00 pfs_mountd.rpc
root 17795      1 0 Mar 23 ? 0:00 pfsd
root 17793      1 0 Mar 23 ? 0:00 pfs_mountd
```

- ◆ If the daemons are not running, start them:

```
# nohup pfs_mountd &
# nohup pfsd &
```

- ◆ If necessary, create a directory under which to mount the software disc:

```
# mkdir -p /cdrom
```

- ◆ Mount the CD-ROM. For example, to mount the CD-ROM to the `/cdrom` mount point, type:

```
# /usr/sbin/pfs_mount -o iso /dev/dsk/c0t0d0 /cdrom
```

`/dev/dsk/c0t0d0` is the block device file of the drive.

4. Verify the mounting process:

```
# mount
```

In the output, look for a line that resembles the example below:

```
/cdrom on slias15:/cdrom ro,nointr,retrans=3,timeo=7,
  wsize=2048,proto=udp on Wed Mar 23 14:28:37 2005
```



## Using the VERITAS Product Installer to Install SFRAC

Review this section to install SFRAC using the VERITAS product installer. This tool offers a high-level approach to installing multiple VERITAS products. If you prefer a more direct approach to specifically installing SFRAC, skip this section and proceed to [“Using the installsfrac Script to Install SFRAC”](#) on page 31.

1. Navigate to the directory containing the installer program:

```
# # cd /cdrom
```

2. Start the product installer:

```
# ./installer
```

3. Enter "i" to select the "Install/Upgrade a Product" option.
4. Select "VERITAS Storage Foundation for Oracle RAC" in the product list.
5. After viewing the copyright message, proceed to [step 3 of "Verifying Installation Requirements"](#) on page 31 to continue the installation.





## Using the `installsfrac` Script to Install SFRAC

Review this section to install SFRAC using the `installsfrac` utility; you can also install SFRAC using the VERITAS product installer as described on [page 30](#).

### Verifying Installation Requirements

1. Navigate to the folder containing the `installsfrac` script:

```
# cd /cdrom/storage_foundation_for_oracle_rac
```

2. Start `installsfrac`:

```
# ./installsfrac
```

3. Enter the names of the nodes where you want to install the software:

```
VERITAS STORAGE FOUNDATION FOR ORACLE RAC 4.1 INSTALLATION
PROGRAM
```

```
Enter the system names separated by spaces on which to install
SFRAC: slpas06 slpas07 slpas08 slpas09
```

4. Review the output as the script checks the operating system and patches on each node and verifies that the local node running the script can communicate with remote nodes:

```
Verifying communication with slpas06 ..... ping successful
Attempting remsh with slpas06 ..... remsh successful
Attempting rcp with slpas06 ..... rcp successful
Checking OS version on slpas06 ..... HP-UX B.11.23
Checking VRTSdbac package ..... not installed
Checking for patch PHKL_31500 ..... installed
.
.
Checking for patch PHCO_32611 ..... installed
Checking for patch DiskQuota-Enh ..... installed
Checking for patch FSLibEnh ..... installed
Checking for patch FSCmdsEnh ..... installed
Creating log directory on slpas09 ..... Done
```

At the conclusion of these checks, the script:

- ◆ Creates a log called `installsfracdate_time` on each system in the directory `/var/tmp`. See [“Summary of Configuration”](#) on page 47 for information about this log and other files created during installation.



- ◆ Specifies the utilities it uses to communicate with remote systems (typically `remsh` and `rcp`).
5. Review the output as the utility checks whether the VERITAS infrastructure packages, `VRTSvlic` and `VRTSvlic`, are present on each node.

Installing VERITAS Infrastructure packages on `slpas06`:

```
Checking VRTSvlic package ..... not installed
Checking VRTSvlic package ..... not installed
Checking file system space ..... required space is available
Installing VRTSvlic 4.1 on slpas06 ..... Done
Installing VRTSvlic 3.02.006c on slpas06 ..... Done
.
```

- ◆ If the `VRTSvlic` package is not present, the utility installs it on each node after checking for sufficient disk space.
  - ◆ If the `VRTSvlic` package is not the current version, the utility prompts you to upgrade the package to the current version. The utility exits if you decline to upgrade the package.
  - ◆ If the `VRTSvlic` package is not present, the utility installs it on each node.
6. Enter the license key for VERITAS Storage Foundation *for Oracle RAC*. You can also enter keys for additional product features:

SFRAC Licensing Verification:

```
Checking SFRAC license key on slpas06 ..... not licensed
Enter a SFRAC license key for slpas06: [?]
XXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXX
Registering VERITAS Storage Foundation for Oracle RAC PERMANENT
key on slpas06
```

```
Do you want to enter another license key for slpas06? [y,n,q,?]
(n)
```

```
Checking SFRAC license key on slpas07 ..... not licensed
.
.
```

```
Do you want to enter another license key for slpas09? [y,n,q,?]
(n)
SFRAC licensing completed successfully.
```



## 7. View the list of packages to install on each node:

```

installer will install the following SFRAC packages:
VRTSperl      VERITAS Perl 5.8.0 Redistribution
VRTSob        VERITAS Enterprise Administrator Service
VRTSobgui     VERITAS Enterprise Administrator
VRTSat        VERITAS Authentication Service
VRTSvxvm      VERITAS Volume Manager Binaries
VRTSalloc     VERITAS Volume Manager Intelligent Storage
                Provisioning
VRTSvmpro     VERITAS Volume Manager Management Services Provider
VRTSddlpr     VERITAS Device Discovery Layer Services Provider
VRTSfspro     VERITAS File System Management Services Provider
VRTSvmdoc     VERITAS Volume Manager Documentation
VRTSvxfs      VERITAS File System
VRTSfsman     VERITAS File System Manual Pages
VRTSfsdoc     VERITAS File System Documentation
VRTSap        VERITAS Action Provider
VRTStep       VERITAS Task Provider
VRTSllt       VERITAS Low Latency Transport
VRTSgab       VERITAS Group Membership and Atomic Broadcast
VRTSvxfen     VERITAS I/O Fencing
VRTSvcs       VERITAS Cluster Server
VRTSvcsag     VERITAS Cluster Server Bundled Agents
VRTSvcsmg     VERITAS Cluster Server Message Catalogs
VRTSvcsmn     VERITAS Cluster Server Man Pages
VRTSvcsdc     VERITAS Cluster Server Documentation
VRTSjre       VERITAS Java Runtime Environment Redistribution
VRTScutil     VERITAS Cluster Utilities
VRTScscm      VERITAS Cluster Server Cluster Manager
VRTSweb       VERITAS Java Web Server
VRTSvcsw      VERITAS Cluster Manager (Web Console)
VRTScscw      VERITAS Cluster Server Configuration Wizards
VRTScssim     VERITAS Cluster Server Simulator
VRTSglm       VERITAS Group Lock Manager
VRTScavf      VERITAS Cluster Server Agents for Cluster File
                System
VRTSdbac      VERITAS Oracle Real Application Cluster Support
                Package
VRTSgms       VERITAS Group Messaging Services
VRTSodm       VERITAS Oracle Disk Manager
VRTSdbckp     VERITAS Storage Foundation Checkpoint Scripts
VRTSormp      VERITAS Storage Mapping Services for Oracle Database
VRTSvcsor     VERITAS Cluster Server Oracle Enterprise Extension
VRTScsocw     VERITAS Cluster Server Oracle 9i RAC Wizard
VRTSvxmsa     VERITAS Mapping Service, Application Libraries

```



8. Review and respond to the script as it verifies system requirements.
  - ◆ The utility determines if any of the packages to install are already installed.
  - ◆ The utility checks for the required file system space.
  - ◆ The utility checks for the presence of processes that could conflict with the installation.

If requirements for installation are not met, the utility stops and indicates the actions required to proceed with the process:

```
Checking system installation requirements:
```

```
Checking SFRAC installation requirements on slpas06:
```

```
Checking VRTSperl package ..... not installed
Checking VRTSob package ..... not installed
Checking VRTSobgui package ..... not installed
Checking VRTSvxvm package ..... not installed
Checking VRTSalloc package ..... not installed
.
.
Checking lmx driver ..... not running
Checking vxglm driver ..... not running
Checking vxfen driver ..... not running
Checking gab driver ..... not running
Checking llt driver ..... not running
```

```
Installation requirement checks completed successfully.
```

## installer: Installing Packages

1. After the installer checks for system and installation requirements, choose to install SFRAC packages simultaneously or consecutively on each node:

```
SFRAC can be installed on systems consecutively or
simultaneously. Installing on systems consecutively takes more
time but allows for better error handling.
```

```
Would you like to install Storage Foundation for Oracle RAC on
all systems simultaneously? [y,n,q,?] (y)
```

2. Verify the installation process completes successfully. This step usually takes about 20 minutes on a 2 GB machine with two CPUs:

```
Installing Storage Foundation for Oracle RAC 4.1 on all systems
simultaneously:
```



```

Installing VRTSperl 4.0.11 on slpas07 ..... Done 1 of 168 steps
Installing VRTSperl 4.0.12 on slpas06 .... Done 2 of 168 steps
Installing VRTSperl 4.0.12 on slpas09 ..... Done 3 of 168 steps
Installing VRTSperl 4.0.12 on slpas08 ..... Done 4 of 168 steps
Installing VRTSob 3.2.532.0 on slpas06 .... Done 5 of 168 steps
.
.
Installing VRTScsocw 4.1 on slpas09 ..... Done 165 of 168 steps
Installing VRTScsocw 4.1 on slpas07 ..... Done 166 of 168 steps
Installing VRTSvxmsa 4.2.1-REV=build218_2004.10.29 on slpas09
..... Done 167 of 168 steps
Installing VRTSvxmsa 4.2.1-REV=build218_2004.10.29 on slpas07
..... Done 168 of 168 steps

Storage Foundation for Oracle RAC installation completed
successfully.

```

## Summary of Installation

Review the output at the end of the installation:

Installation of Storage Foundation for Oracle RAC 4.1 has completed successfully.

The installation summary is saved at:

```
/opt/VRTS/install/logs/installsffrac104130927.summary
```

The installsfrac log is saved at:

```
/opt/VRTS/install/logs/installsffrac104130927.log
```

Note that the summary and log files contain the date as part of their names. The script also prompts you to reboot the nodes using the `/usr/sbin/shutdown -r now` command before proceeding to configure SFRAC.



## Configuring the Cluster and Optional Features

Configuring the cluster involves setting up optional features including VCS Cluster Manager (Web Console), SMTP notification, and SNMP trap notification. Use either of the following tools to start the configuration phase:

- ◆ The VERITAS product installer (see “[Using the VERITAS Product Installer to Configure SFRAC](#)” on page 38). This tool offers a high-level approach to configuring VERITAS products.
- ◆ The `installsfrac -configure` script (see “[Using the installsfrac -configure Script to Configure SFRAC](#)” on page 39). This tool offers a direct approach to configuring SFRAC.

## Mounting the VERITAS SFRAC Disc and Configuring VxSS

1. Log in as `root` user to a cluster node or a remote node in the same subnet as the cluster nodes.
2. Insert the VERITAS software disc into the CD drive and determine the appropriate CD-ROM device:

```
# ioscan -fn -C disk
```

For example, the listing may indicate the block device of the CD-ROM drive is `/dev/dsk/c0t0d0`; make a note of the device file as it applies to your system.

3. If necessary, create a directory under which to mount the software disc and mount the disc:

```
# mkdir -p /cdrom
# mount -F cdfs /dev/dsk/c0t0d0 /cdrom
```

---

**Note** VERITAS advises using the `-F cdfs` option for the `mount` command.

---

Alternately, you can mount the disc using Portable File System (PFS):

- ◆ Check whether the PFS daemons are running:

```
# ps -ef | grep pfs_mountd
```

- ◆ If the daemons are running, look for output resembling the example below:

```
# ps -ef | grep pfs
root 17796 17795 0 Mar 23 ? 5:51 pfsd.rpc
root 17794 17793 0 Mar 23 ? 0:00 pfs_mountd.rpc
root 17795      1 0 Mar 23 ? 0:00 pfsd
root 17793      1 0 Mar 23 ? 0:00 pfs_mountd
```

- ◆ If the daemons are not running, start them:

```
# nohup pfs_mountd &
# nohup pfsd &
```

- ◆ If necessary, create a directory under which to mount the software disc:

```
# mkdir -p /cdrom
```

- ◆ Mount the CD-ROM. For example, to mount the CD-ROM to the /cdrom mount point, type:

```
# /usr/sbin/pfs_mount -o iso /dev/dsk/c0t0d0 /cdrom
```

/dev/dsk/c0t0d0 is the block device file of the drive.

4. Verify the mounting process:

```
# mount
```

In the output, look for a line that resembles the example below:

```
/cdrom on slias15:/cdrom ro,nointr,retrans=3,timeo=7,
  wsize=2048,proto=udp on Wed Mar 23 14:28:37 2005
```

5. Install the root broker only if you plan on configuring VERITAS Security Services (VxSS); refer to the *VERITAS Cluster Server User's Guide* for details on the root broker and VxSS. VERITAS recommends that you install the root broker on a stable system outside the cluster:

- a. Navigate to the location of the script:

```
# cd cluster_server
# ./installvcs -security
```

- b. Enter "3" to install VxSS root broker.

- c. After reviewing the VxSS information, agree to install the VERITAS Authentication Service in R+AB mode.

- d. Enter the system on which to install the Authentication Service.

- e. Review the output as the script verifies system and installation requirements, and installs the Authentication Service.

- f. Review the output at the end of the installation and note the locations of the summary and log files that contain the date as part of their names.



## Using the VERITAS Product Installer to Configure SFRAC

Review this section to configure SFRAC using the VERITAS product installer. This tool offers a high-level approach to configuring VERITAS products. If you prefer a more direct approach to specifically configuring SFRAC, skip this section and proceed to [“Using the installsfrac Script to Install SFRAC”](#) on page 31.

1. Navigate to the directory containing the installer program:

```
# cd /cdrom
```

2. Start the product installer:

```
# ./installer
```

3. Enter "c" to select the “Configure an Installed Product” option.
4. Select “VERITAS Storage Foundation for Oracle RAC” in the product list.
5. After viewing the copyright message, proceed to [step 3 of “Verifying Installation Requirements”](#) on page 31 to continue the configuration.





## Using the `installsfrac -configure` Script to Configure SFRAC

Review this section to configure SFRAC using the `installsfrac -configure` utility; you can also configure SFRAC using the VERITAS product installer as described on [page 38](#).

### Checking Requirements for Configuring Cluster

1. Navigate to the folder containing the `installsfrac -configure` script:

```
# cd /cdrom/storage_foundation_for_oracle_rac
```

2. Start the script:

```
# ./installsfrac -configure
```

3. Enter the names of the nodes where you want to configure the software:

```
VERITAS STORAGE FOUNDATION FOR ORACLE RAC 4.1 CONFIGURATION
PROGRAM
```

Enter the system names separated by spaces on which to configure SFRAC: **slpas06 slpas07 slpas08 slpas09**

4. Review the output as the script checks the operating system and patches on each node and verifies that the local node running the script can communicate with remote nodes:

```
Verifying communication with slpas06 ..... ping successful
Attempting remsh with slpas06 ..... remsh successful
Attempting rcp with slpas06 ..... rcp successful
Checking OS version on slpas06 ..... HP-UX B.11.23
Checking VRTSdbac package ..... version 4.1 installed
Checking for patch PHKL_31500 ..... installed
.
.
Checking for patch PHCO_32611 ..... installed
Checking for patch DiskQuota-Enh ..... installed
Checking for patch FSLibEnh ..... installed
Checking for patch FSCmdsEnh ..... installed
Creating log directory on slpas09 ..... Done
```

At the conclusion of these checks, the script:

- ◆ Creates a log called `installsfracdate_time` on each system in the directory `/var/tmp`. See [“Summary of Configuration”](#) on page 47 for information about this log and other files created during installation.



- ◆ Specifies the utilities it uses to communicate with remote systems (typically `remsh` and `rcp`).
5. After verifying the license keys for SFRAC and additional features on each node, review the output as the script checks various response files and processes:

Checking SFRAC processes on slpas06:

```
Checking had process ..... not running
Checking hashadow process ..... not running
Checking CmdServer process ..... running
```

SFRAC processes must be stopped completely before it can be configured by `installsfrac`.

Do you want to stop SFRAC processes? [y,n,q] (y) y

```
Killing CmdServer ..... Done
Checking notifier process ..... not running
Checking odm driver ..... not running
Checking vcsmm driver ..... vcsmm module loaded
.
.
Stopping vxfen driver ..... Done
Checking gab driver ..... gab module loaded
Stopping gab driver ..... Done
Checking lltdriver ..... lltdriver module loaded
Stopping lltdriver ..... Done
```

SFRAC processes are stopped



## Configuring Heartbeat Links, VxSS, Cluster Manager, and SMTP/SNMP Notification

6. After reviewing the instructions on how to respond to the configuration questions, enter the unique cluster name and ID (see “[Cluster Configuration](#)” on page 26):

To configure VCS for SFRAC the following is required:

A unique Cluster name

A unique Cluster ID number between 0-255

Two or more NIC cards per system used for heartbeat links

All systems are being configured to create one cluster

Enter the unique cluster name: [?] racclus69

Enter the unique Cluster ID number between 0-255: [b,?] **169**

7. Enter the details for private heartbeat links:

- a. When specifying the private heartbeat NICs for cluster nodes, note lan0 is typically the network interface card for public network only:

```
Discovering NICs on slpas06 ... discovered lan0 lan1 lan2
lan3 lan4
```

```
Enter the NIC for the first private heartbeat NIC on slpas06:
[b,?] lan1
```

```
Would you like to configure a second private heartbeat link?
[y,n,q,b,?] (y)
```

```
Enter the NIC for the second private heartbeat link on
slpas06:
```

```
[b,?] lan2
```

```
Would you like to configure a third private heartbeat link?
[y,n,q,b,?] (n)
```

```
Are you using the same NICs for private heartbeat links on all
systems? [y,n,q,b,?] (y)
```

In this example,lan1 and lan2 are the private heartbeat NICs on all nodes. You may use NICs with different device names on some of the nodes. If necessary, indicate the NICs are different when prompted.

- b. Verify the information:

```
Cluster Name: racclus69
```

```
Cluster ID Number: 169
```

```
Private Heartbeat NICs for slpas06: link1=lan1 link2=lan2
```

```
Private Heartbeat NICs for slpas07: link1=lan1 link2=lan2
```

```
Private Heartbeat NICs for slpas08: link1=lan1 link2=lan2
```



```
Private Heartbeat NICs for slpas09: link1=lan1 link2=lan2
```

```
Is this information correct? [y,n,q] (y)
```

8. After reviewing the information on VERITAS Security Services (VxSS), indicate whether or not to configure these services.
  - ◆ If you configure VxSS, view the output as the script checks the VRTSAt package and enter the root broker when prompted.
  - ◆ If you decline to configure VxSS, enter the user name, password, and privilege information and confirm it.

9. Enter the details for Cluster Manager (Web Console):

- a. Specify whether you want to configure Cluster Manager (Web Console). This configuration requires the name of a NIC device, a virtual IP address, and a net mask for use by each node:

```
The following information is required to configure Cluster Manager:
```

```
A public NIC used by each system in the cluster  
A Virtual IP address and netmask for Cluster Manager  
One or more NetworkHosts IP addresses for connection checking
```

```
Do you want to configure Cluster Manager (Web Console) [y,n,q]  
(y)
```

If you decline to configure the Web Console, proceed to [step 10](#) to configure SMTP notification.

- b. Confirm whether you want to use the discovered public NIC on the first node:

```
Active NIC devices discovered on slpas06: lan0  
Enter the NIC for Cluster Manager (Web Console) to use on  
slpas06: [b,?](lan0)  
Is lan0 to be the public NIC used by all systems [y,n,q,b,?]  
(y)
```

- c. Enter the virtual IP address for the Web Console:

```
Enter the Virtual IP address for Cluster Manager: [b,?]  
10.180.88.188
```

- d. Confirm or enter the net mask:

```
Enter the netmask for IP 10.180.88.188: [b,?] (255.255.240.0)
```



**e.** Enter the NetworkHosts IP addresses:

```
Enter the NetworkHosts IP addresses, separated by spaces:
[b,?] 10.216.168.40 10.216.168.41 10.216.168.42
10.216.168.43
```

**f.** Verify the information on the Web Console:

```
Cluster Manager (Web Console) verification:
```

```
NIC: lan0
IP: 10.180.88.188
Netmask: 255.255.240.0
Netmask: 10.216.168.40 10.216.168.41 10.216.168.42
10.216.168.43
```

```
Is this information correct? [y,n,q] (y)
```

If the information is *not* correct, answer "n". The installer prompts you to enter the information again.

**10.** Enter the details for SMTP notification:**a.** Specify whether you want to configure SMTP notification:

```
The following information is required to configure SMTP
notification:
```

```
The domain-based hostname of the SMTP server
The email address of each SMTP recipient
A minimum severity level of messages to send to each
recipient
```

```
Do you want to configure SMTP notification? [y,n,q] (y)
```

If you decline to configure SMTP notification, proceed to [step 11](#) to configure SNMP notification.

**b.** Provide the details to configure SMTP notification:

```
Enter the domain-based hostname of the SMTP server
(example: smtp.yourcompany.com): [b,?] smtp.xyzstar.com
Enter the full email address of the SMTP recipient
(example: user@yourcompany.com): [b,?] ozzie@xyzstar.com
Enter the minimum severity of events for which mail should be
sent to ozzie@veritas.com [I=Information, W=Warning,
E=Error, S=SevereError]: [b,?] w
Would you like to add another SMTP recipient? [y,n,q,b] (n) n
```

**c.** Verify the summary information:

```
SMTP Address: smtp.xyzstar.com
Recipient: ozzie@xyzstar.com receives email for Warning or
higher events
```

Is this information correct? [y,n,q] (y)

If the information is *not* correct, answer "n" and enter the information again.

**11. Enter the details for SNMP trap notification:**

**a. Specify whether you want to configure SNMP notification:**

The following information is required to configure SNMP notification:

```
System names of SNMP consoles to receive VCS trap messages
SNMP trap daemon port numbers for each console
A minimum severity level of messages to send to each
console
```

Do you want to configure SNMP notification? [y,n,q] (y)

If you decline to configure SNMP notification, proceed to [step 13](#).

**b. Provide the details to configure SNMP trap notification:**

```
Enter the SNMP trap daemon port: [b,?] (162)
Enter the SNMP console system name: [b,?] saturn
Enter the minimum severity of events for which SNMP traps
should be sent to saturn [I=Information, W=Warning, E=Error,
S=SevereError]: [b,?] i
Would you like to add another SNMP console? [y,n,q,b] (n)
```

**12. Verify the summary information:**

```
SNMP Port: 162
Console: saturn receives SNMP traps for Information or
higher events
```

Is this information correct? [y,n,q] (y)

If the information is *not* correct, answer "n" and enter the information again.

**13. Enter the timeout interval for CVM:**

```
Enter Cluster Volume Manager cluster reconfiguration timeout
(sec): (200)
```

Refer to the VERITAS Volume Manager documentation for more information on CVM.

## Starting VxVM and Specifying VxVM Information

1. Disks on HP systems typically use device names, such as `/dev/rdsk/c0t1d1`, to identify disks on the system. You can choose to use the VM *enclosure-based* naming scheme, which makes disk arrays more readily recognizable. Most users do not choose this option.

The enclosure-based naming scheme is a feature of Volume Manager. It allows one to reference disks using a symbolic name that is more meaningful than the operating system's normal device access name. This symbolic name is typically derived from the array name.

```
Do you want to set up the enclosure-based naming scheme?
[y,n,q,?] (n)
```

If you want to set up the scheme, specify to do this for all eligible systems:

```
Do you want to use the enclosure-based naming scheme for all of
the eligible systems? [y,n,q,?] (y)
```

Refer to the VERITAS Volume Manager documentation for details on this scheme.

2. The script creates the Storage Foundation *for Oracle RAC* configuration files and copies them to each cluster node:

```
Configuring Storage Foundation for Oracle RAC:
```

```
Creating Cluster Server configuration files ..... Done
Copying configuration files to slpas06 ..... Done
Copying configuration files to slpas07 ..... Done
Copying configuration files to slpas08 ..... Done
Copying configuration files to slpas09 ..... Done
```

Storage Foundation for Oracle RAC configured successfully.

3. View the processes that Storage Foundation *for Oracle RAC* starts up:

```
Starting SFRAC:
```

```
Starting LLT on slpas06 ..... Started
Starting LLT on slpas06 ..... Started
Starting LLT on slpas08 ..... Started
Starting LLT on slpas09 ..... Started
Starting GAB on slpas06 ..... Started
.
.
Starting SFRAC drivers on slpas08 ..... Started
Starting Cluster Server on slpas09 ..... Started
Checking SFRAC drivers on slpas09 ..... Checked
```



```
Starting SFRAC drivers on slpas09 ..... Started
Confirming SFRAC startup ..... 4 systems RUNNING
```

Note: The vxconfigd daemon will be started, which can take a while depending upon the hardware configuration.

```
Disabling enclosure-based naming on slpas06 ..... Done
Starting vxconfigd for VxVM on slpas06 ..... Started
Disabling enclosure-based naming on slpas07 ..... Done
Starting vxconfigd for VxVM on slpas07 ..... Started
```

```
.
```

Starting vxfen in disabled mode on target systems...

```
Starting vxfen on slpas06 ..... Done
Starting vxfen on slpas07 ..... Done
Starting vxfen on slpas08 ..... Done
Starting vxfen on slpas09 ..... Done
```

Done with starting vxfen on target systems...

```
Starting glm on slpas06 ..... Starting Cluster and Agents on
Starting glm on slpas07 ..... Starting Cluster and Agents on
Starting glm on slpas08 ..... Starting Cluster and Agents on
Starting glm on slpas06 ..... Starting Cluster and Agents on
```

Configuring CFS agents for SFRAC:

```
Confirming VxVM configuration daemons startup
..... All systems Enabled
Starting CFS agents
..... Started
```

The script prompts you to reboot the nodes with the `/usr/sbin/shutdown -r now` command.

4. If applicable, set up the default disk group. Because some VxVM commands require that a disk group be specified, the installer enables you to register the name of a default VxVM disk group (which you can create later) on each eligible node:

Many Volume Manager commands affect the contents or configuration of a disk group. Such commands require that the user specify a disk group. This is accomplished by using the `-g` option of a command or setting the `VXVM_DEFAULTDG` environment variable. An alternative to these two methods is to configure the default disk group of a system.

```
Do you want to set up a system wide default disk group?
[y,n,q,?] (y) y
```





Which disk group? [<group>,list,q,?] **xyz\_dg**

5. If you specified setting up a default disk group, review the setup output:

Volume Manager default disk group setup and daemon startup

```
Setting default diskgroup to xyz_dg on slpas06 ..... Done
Starting vxrelocd on slpas06 ..... Started
Starting vxcached on slpas06 ..... Started
Starting vxconfigbackupd on slpas06 ..... Started
.
```

## Summary of Configuration

Review the final output:

Configuration of Storage Foundation for Oracle RAC 4.1 has completed successfully.

The configuration summary is saved at:

```
/opt/VRTS/install/logs/installsfrac104130927.summary
```

The installsfrac log is saved at:

```
/opt/VRTS/install/logs/installsfrac104130927.log
```

The summary and log files contain the date as part of their names. The script also notes that the VCS `cvm` service group remains in a `FAULTED` state until you configure the coordinator disks as described on [page 54](#).

At this point, running the `gabconfig -a` command on all nodes enables you see ports `a`, `d`, `h`, and `o` for GAB membership.

For example:

```
# gabconfig -a
GAB Port Memberships
=====
Port a gen  ada401  membership 0123
Port d gen  ada409  membership 0123
Port h gen  ada407  membership 0123
Port o gen  ada406  membership 0123
```



## Setting Up Shared Storage and I/O Fencing for SFRAC

The shared storage for VERITAS Storage Foundation *for Oracle RAC* must support SCSI-3 persistent reservations to enable I/O fencing (see “[I/O Fencing](#)” on page 13). SFRAC involves two types of shared storage: data disks to store shared data, and coordinator disks, which are small LUNs (typically three per cluster), to control access to data disks by the nodes.

Setting up shared storage and I/O fencing involve the following tasks:

- ◆ Adding disks. See “[Adding and Initializing Disks](#)” on page 48.
- ◆ Testing data disks. See “[Testing Disks Using vxfcntlshdw](#)” on page 51.
- ◆ Setting up coordinator disks. See “[Setting Up Coordinator Disks](#)” on page 54.
- ◆ Editing the VCS configuration file for I/O fencing. See “[Editing the UseFence Attribute in VCS Configuration](#)” on page 59.
- ◆ Stopping and starting each node to bring up the cluster configuration with I/O fencing enabled. See “[Starting VCS, CVM, and CFS on All Nodes](#)” on page 60.

### Adding and Initializing Disks

---

**Note** Install the driver and HBA card. Refer to the documentation from the vendor for instructions.

---

After you physically add shared disks to the nodes, you must initialize them as VxVM disks. Use the following examples; see the *VERITAS Volume Manager Administrator's Guide* for more information on adding and configuring disks.

1. Use the `ioscan` and `insf` commands on each node to make the new disks recognizable:

```
# ioscan -C disk
# insf -e
```

---

**Caution** The HP-UX man page for the `insf` command instructs you to run the command in single-user mode only. Running `insf -e` in multi-user mode is acceptable assuming no other user is accessing any of the device files. This command can change the mode, owner, or group of an existing special (device) file, or unlink and recreate a file. Special files that are currently open may be left in an indeterminate state.

---

2. Verify that the Array Support Library (ASL) for the disk array is installed on each of the nodes. Run the following command on each node and examine the output to verify the installation of ASL:

```
# vxddladm listsupport all
LIB_NAME                ARRAY_TYPE    VID          PID
=====
.
libv xemc.s1            A/A          EMC          SYMMETRIX
.
```

If the ASL for the array you are adding is not installed, obtain and install it on each node before proceeding. The ASL for the supported storage device you are adding is available from the disk array vendor or VERITAS technical support.

3. Scan all disk drives and their attributes, update the VxVM device list, and reconfigure DMP with the new devices. Type:

```
# vxdisk scandisks
```

4. To initialize the disks as VxVM disks, use one of the following methods:

- ◆ Use the interactive `vxdiskadm` utility to initialize the disks as VxVM disks.
- ◆ Use the `vxdisksetup` command to initialize a disk as a VxVM disk. The example that follows specifies the CDS format:

```
vxdisksetup -i device_name format=cdsdisk
```

For example:

```
# vxdisksetup -i c2t0d2 format=cdsdisk
```

## Verifying the Nodes See the Same Disk

To confirm whether a disk (or LUN) supports SCSI-3 persistent reservations, two nodes must simultaneously have access to the same disks. Because a shared disk is likely to have a different name on each node, check the serial number to verify the identity of the disk. Use the `vxfenadm` command with the `-i` option to verify that the same serial number for the LUN is returned on all paths to the LUN.

For example, an EMC array is accessible by the `/dev/rdisk/c2t13d0` path on node A and by the `/dev/rdisk/c2t11d0` path on node B. From node A, enter:

```
# vxfenadm -i /dev/rdisk/c2t13d0
Vendor id      : EMC
Product id    : SYMMETRIX
Revision      : 5567
Serial Number  : 42031000a
```



The same serial number information should appear when you enter the equivalent command on node B using the `/dev/rrdsk/c2t11d0` path.

On a disk from another manufacturer, Hitachi Data Systems, the output is different and may resemble:

```
# vxfenadm -i /dev/rrdsk/c2t0d2
Vendor id      : HITACHI
Product id     : OPEN-3      -HP
Revision       : 0117
Serial Number  : 0401EB6F0002
```

Refer to the `vxfenadm(1M)` manual page.



## Testing Disks Using vxfcntlsthdw

Use the `vxfcntlsthdw` utility to test the shared storage arrays support SCSI-3 persistent reservations and I/O fencing. Make sure to test disks serving as coordinator disks; see [“Setting Up Coordinator Disks”](#) on page 54.

### Reviewing Guidelines on vxfcntlsthdw

- ◆ Verify the connection of the shared storage for data to two of the nodes on which you installed Storage Foundation *for Oracle RAC*.

---

**Caution** The tests overwrite and destroy data on the disks unless you use the `-r` option.

---

- ◆ The two nodes must have `remsh` permission set to ensure each node has root user access to the other. Temporarily modify the `/.rhosts` file to enable cluster communications for the `vxfcntlsthdw` utility, placing a `++` character in the first line of the file. You can also limit the remote access to specific systems. Refer to the manual page for the `/.rhosts` file for more information. See [“Removing remsh Permissions and Restoring Public Network Connections”](#) on page 58 after completing the testing process.
- ◆ To ensure both nodes are connected to the same disk during the testing, use the `vxfenadm -i diskpath` command to verify the disk serial number. See [“Verifying the Nodes See the Same Disk”](#) on page 49.

### Running vxfcntlsthdw

This procedure uses the `/dev/rdisk/c4t8d0` disk in the steps.

1. Make sure system-to-system communication is functioning properly. See [“System-to-System Communication”](#) on page 24.

2. From one node, start the utility:

```
# /opt/VRTSvcS/vxfen/bin/vxfcntlsthdw
```

3. After reviewing the overview and warning that the tests overwrite data on the disks, confirm to continue the process and enter the node names.

```
***** WARNING!!!!!!!!!! *****
THIS UTILITY WILL DESTROY THE DATA ON THE DISK!!

Do you still want to continue : [y/n] (default: n)
y
Enter the first node of the cluster:
```



**slpas06**

Enter the second node of the cluster:

**slpas07**

4. Enter the names of the disks you are checking. For each node, the disk may be known by the same name:

Enter the disk name to be checked for SCSI-3 PGR on node slpas06  
in the format: /dev/rdisk/cxtxdx  
**/dev/rdisk/c4t8d0**

Enter the disk name to be checked for SCSI-3 PGR on node slpas07  
in the format: /dev/rdisk/cxtxdx  
Make sure it's the same disk as seen by nodes slpas06 and slpas07  
**/dev/rdisk/c4t8d0**

The disk names, regardless if they are identical, must refer to the same physical disk to facilitate the testing.

5. Review the output as the utility performs the checks and report its activities:

```
Evaluating the disk before testing 0 Pre-existing keys.....  
.....Passed.  
Registering keys on disk /dev/rdisk/c4t8d0 from node  
slpas06 ..... Passed.  
Verifying registrations for disk /dev/rdisk/c4t8d0 on node  
slpas06 ..... Passed.  
Reads from disk /dev/rdisk/c4t8d0 on node slpas06 ..... Passed.  
Writes to disk /dev/rdisk/c4t8d0 from node slpas06 ..... Passed.  
Reads from disk /dev/rdisk/c4t8d0 on node slpas07 ..... Passed.  
Writes to disk /dev/rdisk/c4t8d0 from node slpas07 ..... Passed.  
Reservations to disk /dev/rdisk/c4t8d0 from node slpas06 .....  
..... Passed.  
Verifying reservation for disk /dev/rdisk/c4t8d0s2 on node  
slpas06 ..... Passed.  
.  
.
```

If a disk is ready for I/O fencing on each node, the utility reports success:

```
ALL tests on the disk /dev/rdisk/c4t8d0 have PASSED.  
The disk is now ready to be configured for I/O Fencing on node  
slpas06.
```

```
ALL tests on the disk /dev/rdisk/c4t8d0 have PASSED.  
The disk is now ready to be configured for I/O Fencing on node  
slpas07.
```

```
Removing test keys and temporary files, if any ...  
.
```



6. Run the `vxfcntlshdw` utility for each disk you intend to verify.

---

**Note** The `vxfcntlshdw` utility has additional options suitable for testing many disks. The options for testing disk groups (`-g`) and disks listed in a file (`-f`) are described in detail in “[vxfcntlshdw Options](#)” on page 121. You can also test disks without destroying data using the `-r` option.

---



## Setting Up Coordinator Disks

I/O fencing requires coordinator disks that are configured in a disk group and accessible to each node in the cluster. These disks enables the `vxfen` driver to resolve potential split-brain conditions and prevent data corruption. See “[I/O Fencing](#)” on page 13 for a description of I/O fencing and the role of coordinator disks.

Because a coordinator disk does not apply to data storage, configure the disk as the smallest possible LUN on a disk array to avoid wasting space. Make sure you already added and initialized disks for use as coordinator disks, as described in the requirements below.

However, to use the `vxfcntlsthdw` utility to verify SCSI-3 persistent reservation support, use disks of at least 1 MB. Disks smaller than 1 MB can be tested manually; contact VERITAS support (<http://support.veritas.com>) for the procedure.)

### Requirements for Coordinator Disks

- ✓ You must have at least three coordinator disks and the total number of coordinator disks must be an odd number. This requirement ensures a majority of disks can be achieved.
- ✓ Each of the coordinator disks must use a physically separate disk or LUN.
- ✓ Each of the coordinator disks should exist on a different disk array, if possible.
- ✓ You must initialize each disk as a VxVM disk. VERITAS recommends the default (CDS) format.
- ✓ The coordinator disks must support SCSI-3 persistent reservations. See “[Requirements for Testing the Coordinator Disk Group](#)” on page 55.
- ✓ The coordinator disks must exist in a disk group (for example, `vxencoorddg`). See “[Creating the vxencoorddg Disk Group](#)” on page 55.

VERITAS recommends using hardware-based mirroring for coordinator disks.



## Creating the vxfencoorddg Disk Group

From one node, create a disk group named `vxfencoorddg`. This group must contain an odd number of disks or LUNs and a minimum of three disks. For example, assume the disks have the device names `c1t1d0`, `c2t1d0`, and `c3t1d0`. Refer to the *VERITAS Volume Manager Administrator's Guide* for details on creating disk groups.

1. On any node, create the disk group by specifying the device name of the disks:

```
# vxhdg init vxfencoorddg c1t1d0 c2t1d0 c3t1d0
```

Refer to the *VERITAS Volume Manager Administrator's Guide* for details on creating disk groups.

## Requirements for Testing the Coordinator Disk Group

- ◆ The utility requires that the coordinator disk group, `vxfencoorddg`, be accessible from two nodes.
- ◆ The two nodes must have `remsh` permission set such that each node has root user access to the other. Temporarily modify the `.rhosts` file to enable cluster communications for the `vxfentsthdw` utility, placing a `++` character in the first line of the file. You can also limit the remote access to specific systems. Refer to the manual page for the `.rhosts` file for more information. See [“Removing remsh Permissions and Restoring Public Network Connections”](#) on page 58 when you complete the testing process.
- ◆ To ensure both nodes are connected to the same disks during the testing process, use the `vxfenadm -i diskpath` command to verify the serial number. See [“Verifying the Nodes See the Same Disk”](#) on page 49.

## Using vxfentsthdw -c to Test the Coordinator Disk Group

Use the `vxfentsthdw` utility to verify disks are configured to support I/O fencing. In this procedure, the `vxfentsthdw` utility tests the three disks one disk at a time from each node. From the node `slpas06`, the disks are `/dev/rdisk/c1t1d0`, `/dev/rdisk/c2t1d0`, and `/dev/rdisk/c3t1d0`. From the node `slpas07`, the same disks are seen as `/dev/rdisk/c4t1d0`, `/dev/rdisk/c5t1d0`, and `/dev/rdisk/c6t1d0`.

1. Use the `vxfentsthdw` command with the `-c` option. For example:

```
# /opt/VRTSvcs/vxfen/bin/vxfentsthdw -c vxfencoorddg
```

2. Enter the nodes you are using to test the coordinator disks:

```
Enter the first node of the cluster:
slpas06
```



Enter the second node of the cluster:  
**slpas07**

**3. Review the output of the testing process:**

```
Testing slpas06 /dev/rdisk/clt1d0 slpas07 /dev/rdisk/c4t1d0

Evaluating the disk before testing 0 Pre-existing keys.....
.....Passed.
Registering keys on disk /dev/rdisk/clt1d0 from node
slpas06 ..... Passed.
Verifying registrations for disk /dev/rdisk/clt1d0 on node
slpas06 ..... Passed.
Registering keys on disk /dev/rdisk/c4t1d0 from node
slpas07 ..... Passed.
Verifying registrations for disk /dev/rdisk/clt1d0 on node
slpas06 ..... Passed.
Verifying registrations for disk /dev/rdisk/c4t1d0 on node
slpas07 ..... Passed.
Preempt and aborting key KeyA using key KeyB on node
slpas07 ..... Passed.
Verifying registrations for disk /dev/rdisk/clt1d0 on node
slpas06 ..... Passed.
Verifying registrations for disk /dev/rdisk/c4t1d0 on node
slpas07 ..... Passed.
Removing key KeyB on node slpas07 ..... Passed.
Check to verify there are no keys from node slpas06 .....
..... Passed.

ALL tests on the disk /dev/rdisk/clt1d0 have PASSED.
The disk is now ready to be configured for I/O Fencing on node
slpas06 as a COORDINATOR DISK.

ALL tests on the disk /dev/rdisk/c4t1d0 have PASSED.
The disk is now ready to be configured for I/O Fencing on node
slpas07 as a COORDINATOR DISK.

*****
Testing slpas06 /dev/rdisk/clt1d0 slpas07 /dev/rdisk/clt1d0
.
.
```

After you test all disks in the disk group, the vxfencoorddg disk group is ready for use.



## Removing and Replacing a Failed Disk

If a disk in the coordinator disk group fails verification, remove the failed disk or LUN from the `vxfencoorddg` disk group, replace it with another, and retest the disk group.

- ◆ Use the `vxdiskadm` utility to remove the failed disk from the disk group. Refer to the *VERITAS Volume Manager Administrator's Guide*.
- ◆ Add a new disk to the node, initialize it, and add it to the coordinator disk group. See “[Creating the vxfencoorddg Disk Group](#)” on page 55.
- ◆ Retest the disk group. See “[Requirements for Testing the Coordinator Disk Group](#)” on page 55.

---

**Note** If you need to replace a disk in an active coordinator disk group, refer to the troubleshooting procedure, “[Removing or Adding Coordinator Disks](#)” on page 226.

---

## Configuring /etc/vxfendg Disk Group for I/O Fencing

After setting up and testing the coordinator disk group, configure it for use.

1. Deport the disk group:
 

```
# vxdg deport vxfencoorddg
```
2. Import the disk group with the `-t` option to avoid automatically importing it when the nodes restart:
 

```
# vxdg -t import vxfencoorddg
```
3. Deport the disk group. Deporting the disk group prevents the coordinator disks from serving other purposes:
 

```
# vxdg deport vxfencoorddg
```
4. On all nodes, type:
 

```
# echo "vxfencoorddg" > /etc/vxfendg
```

No spaces should appear between the quotes in the "vxfencoorddg" text.

This command creates the `/etc/vxfendg` file, which includes the name of the coordinator disk group.

Based on the contents of the `/etc/vxfendg` file, the `rc` script creates the `/etc/vxfentab` file for use by the `vxfen` driver when the system starts. `/etc/vxfentab` invokes the `vxfenconfig` command, which configures the `vxfen` driver to start and use the coordinator disks listed in `/etc/vxfentab`. `/etc/vxfentab` is a generated file; do not modify this file.



### Example /etc/vxfentab File

The list of coordinator disks is in the `/etc/vxfentab` file on each node. The same disks may appear using different names on each node. For example, a device could appear as `c0t1d2` on one node and `c4t1d2` on another node. In this case, use the commands `vxfenadm -i /dev/rdisk/c0t1d2` on the first node and `vxfenadm -i /dev/rdisk/c4t1d2` on the second node to verify if the serial number is the same number.

An example of the `/etc/vxfentab` file on one node resembles:

```
/dev/rdisk/c1t1d0  
/dev/rdisk/c2t1d0  
/dev/rdisk/c3t1d0
```

If you must remove disks from or add disks to an existing coordinator disk group, refer to [“Removing or Adding Coordinator Disks”](#) on page 226.

## Starting I/O Fencing

On each node, start the I/O fencing driver:

```
# /sbin/init.d/vxfen start
```

## Removing remsh Permissions and Restoring Public Network Connections

After completing the installation of VERITAS *Storage Foundation for Oracle RAC* and verification of disk support for I/O fencing, remove the temporary `remsh` access permissions you set for the nodes and restore the connections to the public network.

---

**Note** If the nodes use `ssh` for secure communications, and you temporarily removed the connections to the public network, restore the connections at this time.

---

## Editing the UseFence Attribute in VCS Configuration

After adding coordinator disks and configuring I/O fencing, add the `UseFence = SCSI3` cluster attribute to the VCS configuration file, `/etc/VRTSvcs/conf/config/main.cf`. This entry gives the user the flexibility to disable or enable I/O fencing by modifying the `UseFence` attribute.

1. Save the existing configuration:

```
# haconf -dump -makero
```

2. Stop VCS on all nodes:

```
# hastop -all
```

3. Make a backup copy of the `main.cf` file:

```
# cd /etc/VRTSvcs/conf/config
# cp main.cf main.orig
```

4. On one node, use `vi` or another text editor to edit the `main.cf` file. Modify the list of cluster attributes by adding the `UseFence` attribute and assigning its value of `SCSI3`.

```
cluster rac_cluster1 (
    UserNames = { admin = "cDRpdxPmHpzS." }
    Administrators = { admin }
    HacliUserLevel = COMMANDROOT
    CounterInterval = 5
    UseFence = SCSI3
)
```

5. Save and close the file.

6. Verify the syntax of the file `/etc/VRTSvcs/conf/config/main.cf`:

```
# hacf -verify /etc/VRTSvcs/conf/config
```

7. Using `rcp` or another utility, copy the VCS configuration file from a node (for example, `slpas06`) to the remaining cluster nodes. On each remaining node, enter:

```
# rcp slpas06:/etc/VRTSvcs/conf/config/main.cf
   /etc/VRTSvcs/conf/config
```



## Starting VCS, CVM, and CFS on All Nodes

With the configuration file in place on each system, start VCS, CVM, and CFS:

```
# hastart
```

Make sure to run this command from each node.

## Running gabconfig -a to Verify GAB Port Membership

After setting up I/O fencing and starting VCS, CVM, and CFS on each node, verify GAB port membership by running the `gabconfig -a` command.

For example:

```
slpas06# gabconfig -a
GAB Port Memberships
=====
Port a gen    ada401 membership 0123
Port b gen    ada40d membership 0123
Port d gen    ada409 membership 0123
Port f gen    ada41c membership 0123
Port h gen    ada40f membership 0123
Port o gen    ada406 membership 0123
Port v gen    ada416 membership 0123
Port w gen    ada418 membership 0123
```

The output of the `gabconfig -a` command displays the nodes that have membership with the modules installed and configured in the installation. The first line indicates that each node (0, 1, 2, and 3) has membership with the GAB utility that uses "Port a." The software configures the ports in the list for the following functions:

<b>Port</b>	<b>Function</b>
a	GAB
b	I/O fencing
d	Oracle Disk Manager (ODM)
f	Cluster File System (CFS)
h	VERITAS Cluster Server (VCS: High Availability Daemon)
o	VCSMM driver
v	Cluster Volume Manager (CVM)
w	vxconfigd (module for CVM)



## Verifying the CVM Group is Online

On all nodes, type:

```
# hagrps -state cvm
```

Make sure the `cvm` group is ONLINE.

## Unmounting the Software Disc

After verifying the successful installation and configuration of all required packages, unmount the disc by using either the `pfs_umount` or standard `umount` command. For example:

```
# pfs_umount /cdrom
```

or

```
# umount /cdrom
```

`/cdrom` is the location of the disc mount point.

---

**Note** You may have to kill the `pfs` daemons before ejecting the disc.

---

## Rebooting Nodes (General)

When you need to reboot any of the nodes, make sure SFRAC is properly shut down by always using the `shutdown` command with the `-r` option instead of the `reboot` command.



## Example of VCS Configuration File After SFRAC Installation

To verify the installation, you can examine the VCS configuration file, `main.cf`, in the directory `/etc/VRTSvcs/conf/config`. See [“Sample main.cf: After SFRAC Installation and Before Oracle9i Installation”](#) on page 63, for an example of a configuration file.

Note the following information about the VCS configuration file after the SFRAC installation:

- ◆ The “include” statements list types files for VCS (`types.cf`), CFS (`CFSTypes.cf`), and CVM (`CVMTypes.cf`). These files are in the `/etc/VRTSvcs/conf/config` directory. The types file for the Oracle enterprise agent (`OracleTypes.cf`) is also located in `/etc/VRTSvcs/conf/config`. These files define the agents that control the resources in the cluster.
  - ◆ The VCS types include all agents bundled with VCS. Refer to the *VERITAS Bundled Agents Reference Guide* for information about VCS agents.
  - ◆ The CFS types include `CFSMount` and `CFSfsckd`.

The `CFSMount` agent mounts and unmounts the shared volume file systems.

The `CFSfsckd` types are defined for cluster file system daemons and do not require user configuration.
  - ◆ The CVM types include the `CVMCluster`, `CVMVxconfigd`, and `CVMVolDg`.

The `CVMCluster` agent, which is automatically configured during installation, starts CVM in the cluster by autoimporting shared disk groups, controls node membership in the cluster, and defines how nodes communicate the state of volumes. Refer to [“CVMCluster, CVMVxconfigd, CVMVolDg, and CFSMount Agents”](#) on page 243.

The `CVMVxconfigd` agent starts and monitors the `vxconfigd` daemon; this daemon maintains disk and disk group configurations, communicates configuration changes to the kernel, and modifies configuration information stored on disks.

The `CVMVolDg` agent starts the volumes in a shared disk group, sets activation modes, and monitors specified critical volumes.
  - ◆ The Oracle enterprise agent types includes definitions for the Oracle agent and the Netlsnr agent. The Oracle agent monitors the resources for an Oracle database, and the Netlsnr agent manages the resources for the listener process.



- ◆ The cluster definition, with the cluster name provided during installation (for example, `rac_cluster1`), includes the names of users and administrators of the cluster. The `UseFence = SCSI3` attribute is not automatically present; you must manually add it after the installation. See [“Editing the UseFence Attribute in VCS Configuration”](#) on page 59.
- ◆ The `main.cf` now includes the `cvm` service group. This group includes definitions for monitoring the CFS and CVM resources. The `CVMCluster` agent resource definition indicates the nodes use GAB for messaging operations.
- ◆ The `cvm` group has the `parallel` attribute set to 1. This value enables the resources to run in parallel on each node in the system list.

## Sample main.cf: After SFRAC Installation and Before Oracle9i Installation

The configuration file created on your node is located in `/etc/VRTSvcs/conf/config/main.cf`. Review the sample VCS configuration file below:

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "OracleTypes.cf"

cluster racclus69 (
    HacliUserLevel = COMMANDROOT
    CounterInterval = 5
)

system slpas06 (
)

system slpas07 (
)

system slpas08 (
)

system slpas09 (
)

group cvm (
    SystemList = { slpas06 = 0, slpas07 = 1, slpas08 = 2,
                  slpas09 = 3 }
    AutoFailOver = 0
)
```



```
Parallel = 1
AutoStartList = { slpas06, slpas07, slpas08, slpas09 }
)

CFSfsckd vxfsckd (
)

CVMCluster cvm_clus (
    CVMClustName = racclus69
    CVMNodeId = { slpas06 = 0, slpas07 = 1,
                 slpas08 = 2, slpas09 = 3 }
    CVMTransport = gab
    CVMTimeout = 200
)

CVMVxconfigd cvm_vxconfigd (
    Critical = 0
    CVMVxconfigdArgs = { syslog }
)

cvm_clus requires cvm_vxconfigd

// resource dependency tree
//
//     group cvm
//     {
//     CFSfsckd vxfsckd
//     {
//     CVMCluster cvm_clus
//     {
//     CVMVxconfigd cvm_vxconfigd
//     }
//     }
//     }
// }
```

# Installing Oracle9i Software

---

# 4

After setting up VERITAS Storage Foundation *for Oracle RAC*, proceed to install Oracle9i software. You can install the software on shared storage or locally on each node. Make sure to review the Oracle installation manuals before completing the procedures below.

When you use the Oracle installer, do not create a database or install a pre-created database. For example, when you choose to use the “Enterprise Edition” or the “Standard Edition” displayed by the Oracle installer, select the “Software only” option to install the binaries without creating a database. If you choose “Custom installation,” do *not* create a database when prompted to do so.

Create the database after installing Oracle and running the VERITAS script, `/opt/VRTS/bin/linkrac9i`; see [“Linking to VERITAS Libraries”](#) on page 81. Refer to [“Creating a Starter Database”](#) on page 265 for suggested methods to create a database.

## Prerequisite for Installing Oracle9i

Before installing the Oracle database, verify that the shared disk arrays support SCSI-3 persistent reservations and I/O fencing. Refer to [“Setting Up Shared Storage and I/O Fencing for SFRAC”](#) on page 48. I/O fencing is also described in [“I/O Fencing”](#) on page 13.



## Before the Installation: Reviewing Shared Disk Groups and Volumes

This section highlights general information to refer to when dealing with disk groups and volumes. Refer to the *VERITAS Volume Manager Administrator's Guide* for complete details on creating and managing shared disk groups.

### Viewing Information on a Disk Group

To display information about a specific disk group, type:

```
vxdbg list disk_group
```

### Checking the Connectivity Policy on a Shared Disk Group

By default, the connectivity policy for a shared disk group is set to “global.” This setting protects against possible data corruption and causes all nodes in the cluster to detach from the disk group when any node reports a disk failure.

The output of the `vxdbg list shared_disk_group` command includes the following line:

```
detach-policy: global
```

To change the connectivity policy for a disk group from “local” to “global,” type:

```
vxedit set diskdetpolicy=global shared_disk_group
```

### Determining Whether a Node is CVM Master or Slave

On one node (`nebula`), determine whether the node is the master or slave:

```
vxdctl -c mode
```

If `nebula` is the slave and `galaxy` is the master, the output shows:

```
mode: enabled: cluster active - SLAVE  
master: galaxy
```



## Enabling Write Access to Volumes in the Disk Groups

By default, the activation mode for shared disk groups is inactive (set to `off`). To create databases on the shared volumes, enable the write access to the volumes:

On the CVM master node, enter:

```
vxldg -s import shared_disk_group
vxvol -g shared_disk_group startall
vxldg -g shared_disk_group set activation=sw
```

On the slave nodes, enter:

```
vxldg -g shared_disk_group set activation=sw
```

Refer to the description of disk group activation modes in the *VERITAS Volume Manager Administrator's Guide* for more information.

## Deporting and Importing Shared Disk Groups

Shared disk groups in an SFRAC environment are configured for “Autoimport” at the time of CVM startup. If the user manually deports the shared disk group on the CVM master, the disk group is deported on all nodes. To reimport the disk group, the user must import the disk group as a shared group from the CVM master.

To deport a shared disk group, use the following command on the CVM master:

```
vxldg deport shared_disk_group
```

To import a shared disk group, use the following command on the CVM master:

```
vxldg -s import shared_disk_group
```

To import a disk group as a standalone disk group, deport it from the CVM master and use the following command on any node:

```
vxldg -C import shared_disk_group
```

To reimport a disk group as a shared disk group, deport it from the standalone node and use the following command on the CVM master node:

```
vxldg -C -s import shared_disk_group
```



## Creating the Oracle9i User and Group

Create a local group and local user for Oracle on each node. For example, create the `dba` group and the `oracle` user. Make sure to assign the same user ID, group ID, and home directory for the user on each node. Refer to the examples below:

1. On each node running Oracle9i, enter:

```
# groupadd -g 99 dba
# useradd -g dba -u 999 -d /oracle oracle
```

Set up your environment variables as described in the *Oracle Real Application Clusters Installation and Configuration Guide*. Refer to this guide for more information on the `dba` group and `oracle` user.

2. Create a password for the `oracle` user on each node:

```
# passwd oracle
```

3. Make sure that the user `oracle` can `remsh` between the nodes without having to use or confirm a password. If not, in the `oracle` user account directory (for example, `/oracle`), create a `.rhosts` file that includes plus characters ("`+ +`") in the first line.



## Installing and Configuring Oracle9i: Overview

The procedures in this section assume you have already installed Storage Foundation for Oracle RAC 4.1. Installing Oracle9i in a VERITAS Storage Foundation for Oracle RAC environment involves the following tasks:

- ✓ Creating a shared disk group and volume for the Oracle9i SRVM component.
- ✓ Installing Oracle9i Release 2 on either shared disks or a local disk.
- ✓ Installing the Patch 2 for Oracle9i Release 2.
- ✓ Manually link SFRAC libraries with Oracle.

### Creating Shared Storage Location for the SRVM

The Oracle database requires configuring the Oracle Server Management (SRVM) component on a shared storage device. The SRVM component device stores information about the database server configuration.

Use a raw volume or a cluster file system for SRVM. This volume requires at least 200 MB.

#### Method 1: Creating a Separate Location for SRVM

The following procedures give details on creating a separate raw volume and file system for SRVM. If you would rather create a file system in the raw volume designated only for the SRVM component, refer to [“Method 2: Creating a Cluster File System for SRVM \(Optional\)”](#) below. For example, to create a 300 MB `srvm_vol` volume in the `orasrv_dg` shared disk group on the `c2t3d1` disk, complete the following steps:

1. From the CVM master node, create the shared disk group on the shared disk:

```
# vxdg -s init orasrv_dg c2t3d1
```

2. Create the volume in the shared disk group:

```
# vxassist -g orasrv_dg make srvm_vol 300M
```

3. Set the ownership and permissions for the volume:

```
# vxcedit -g orasrv_dg set user=oracle group=dba mode=660 srvm_vol
```

4. Deport and import the disk group, and set the activation mode:

```
# vxdg deport orasrv_dg
# vxdg -s import orasrv_dg
# vxvol -g orasrv_dg startall
# vxdg -g orasrv_dg set activation=sw
```



5. On the other nodes, enter:

```
# vxvg -g orasrv_dg set activation=sw
```

When you install Oracle9i, specify the name of the raw volume (`/dev/vx/rdisk/orasrv_dg/srvm_vol`). If you want to create a cluster file system for the SRVM component, proceed to the next section.

## Method 2: Creating a Cluster File System for SRVM (Optional)

If desired, create a cluster file system for the component in the `srvm_vol` volume. To create the file system:

1. On the CVM master node, create a VxFS file system for SRVM on the shared volume created for this purpose (see previous section). For example, create the file system on `srvm_vol`:

```
# mkfs -F vxfs /dev/vx/rdisk/orasrv_dg/srvm_vol
```

2. On each node, create the identical mount point for the file system:

```
# mkdir /orasrv
```

3. On each node, mount the file system using the device file for the block device:

```
# mount -F vxfs -o cluster,largefiles  
/dev/vx/dsk/orasrv_dg/srvm_vol /orasrv
```

4. On one of the nodes, create a file for `ora_srvm` that is at least 200 MB. For example, to create a file of 200 MB, enter:

```
# cd /orasrv  
# prealloc ora_srvm 200000000
```

5. After mounting the file system, change the ownership again on each of the nodes:

```
# chown -R oracle:dba /orasrv
```



## Installing Oracle9i Release 2 Software

Install Oracle9i Release 2 on a shared disk or local hard disk on each node. After installing the Release 2 software, install the appropriate patches described later in the chapter

### Method 1: Installing Oracle9i Release 2 on Cluster File System

The following procedure describes installing the Oracle9i Release 2 software on shared storage in a RAC environment. To install the Oracle9i Release 2 software locally on each node, see [“Method 2: Installing Oracle9i Release 2 on Local File System”](#) on page 75.

1. Log in as root user.
2. On the CVM master node, create a shared disk group:
 

```
# vxdg -s init orabinvol_dg c2t3d1
```
3. Create the volume in the shared group. For the Oracle9i Release2 binaries, make the volume 5000 MB:
 

```
# vxassist -g orabinvol_dg make orabinvol 5000M
```
4. Deport and import the disk group, and set the activation mode:
 

```
# vxdg deport orabinvol_dg
# vxdg -s import orabinvol_dg
# vxvol -g orabinvol_dg startall
# vxdg -g orabinvol_dg set activation=sw
```
5. On the other nodes, enter:
 

```
# vxdg -g orabinvol_dg set activation=sw
```
6. On the CVM master node, create a VxFS file system on the shared volume on which to install the Oracle9i binaries. For example, create the file system on orabinvol:
 

```
# mkfs -F vxfs -o largefiles /dev/vx/rdsk/orabinvol_dg/orabinvol
```
7. On each node, create the mount point for the file system:
 

```
# mkdir /oracle
```
8. On each node, mount the file system using the device file for the block device:
 

```
# mount -F vxfs -o cluster,largefiles
/dev/vx/dsk/orabinvol_dg/orabinvol /oracle
```



9. On each node, assign ownership of the Oracle directory to `oracle`:

```
# chown -R oracle:dba /oracle
# chmod 775 /oracle
```

10. On the first node, insert Disk1 of the Oracle9i disks in the CD-ROM drive.

- a. Log in as `root` user.

- b. Determine the appropriate CD-ROM device:

```
# ioscan -fn -C disk
```

For example, the listing may indicate the block device of the CD-ROM drive is `/dev/dsk/c1t2d0`; make a note of the device file as it applies to your system.

- c. Check whether the PFS (Portable File System) daemon is running:

```
# ps -ef | grep pfs_mountd
```

If the daemon is running, look for output resembling the following example:

```
root 2853 1825 0 14:49:31 pts/0 0:02 pfs_mountd
```

- d. If the daemon is not running, start it:

```
# nohup pfs_mountd &
# nohup pfsd &
```

- e. If necessary, create a directory under which to mount the software disc:

```
# mkdir -p /cdrom
```

- f. Mount the CD-ROM. For example, to mount the CD-ROM to the mount point `/cdrom`, enter:

```
# /usr/sbin/pfs_mount -t rrip /dev/dsk/c1t2d0 /cdrom
```

`/dev/dsk/c1t2d0` is the block device file of the drive.

- g. Verify the CD-ROM is mounted:

```
# mount
```

In the output, look for a line that resembles the following example:

```
/cdrom on /dev/dsk/c1t2d0 ro on Wed Feb 26 15:11:32 2003
```

11. Log in as `oracle` user on each node.



12. On each node, edit the `.rhosts` file to provide the other nodes access to the local system during the installation. Place a "+" character in the first line of the file. You can remove this permission after the installation is complete.

13. On each node, create a directory for the installation of the Oracle9i binaries. For example:

```
$ mkdir /oracle/VRT
```

14. On each node, set the following environment variables (the examples are for ksh):

```
$ export ORACLE_BASE=/oracle
$ export ORACLE_HOME=/oracle/VRT
$ export LD_LIBRARY_PATH=$ORACLE_HOME/lib
$ export PATH=$PATH:$ORACLE_HOME/bin
$ export CLASSPATH=$CLASSPATH:$ORACLE_HOME/JRE:$ORACLE_HOME\
  /jlib:$ORACLE_HOME/rdbms/jlib:$ORACLE_HOME/network/jlib
```

15. On the first node, set the `DISPLAY` variable.

If you use the Bourne Shell (`sh` or `ksh`):

```
$ DISPLAY=host:0.0;export DISPLAY
```

If you use the C Shell (`csh` or `tcsh`):

```
$ setenv DISPLAY host:0.0
```

16. On the first node, run the Oracle9i `runInstaller` utility and follow the wizard:

```
$ /cdrom/Disk1/runInstaller
```

- a. When `runInstaller` prompts you to run the `/tmp/orainstRoot.sh` script, make sure the script exists *on each node* before proceeding. If so, skip to [step c](#).
- b. If the `/tmp/orainstRoot.sh` script does not exist on each node, copy it from the first node to each of the other cluster nodes.
- c. Run the `/tmp/orainstRoot.sh` script on each node.
- d. When the **Cluster Node Selection** dialog box appears, highlight the local node highlighted for installation.

---

**Note** If the **Node Selection** dialog box does not appear, refer to [“Missing Dialog Box During Installation of Oracle9i Release 2”](#) on page 216.

---



Since the installation of Oracle binaries takes place on CFS, make sure to select *only one node* in the **Cluster Node Selection** dialog box. This node is typically the node on which you are running the installer. Entering more than one node creates unnecessary additional installations. If you inadvertently enter more than one node in the **Node Selection** dialog box, allow the installations to complete rather than interrupt them.

- e. In the **Available Products** dialog box, select **Oracle 9i Database 9.2.0.1.0**.
- f. In the **Installation Types** dialog box:
  - ◆ If you select **Enterprise Edition** or **Standard Edition**, select the **Software Only** installation option.
  - ◆ If you select **Custom Installation**, make sure to decline to build a database in the **Create Database** dialog box.

Refer to the *Oracle9i Installation Guide* for more information on using the utility.

- g. The installer prompts you to designate the location for the Oracle SRVM component. Refer to one of the following options:
  - ◆ Enter the name of a file, such as `ora_srvm`, located on the cluster file system. See “[Method 2: Creating a Cluster File System for SRVM \(Optional\)](#)” on page 70. For example:

```
/orasrv/ora_srvm
```

- ◆ If you used an alternate volume, enter the location here. See “[Method 1: Creating a Separate Location for SRVM](#)” on page 69. For example:

```
/dev/vx/rdisk/orasrv_dg/srvm_vol
```

- h. When prompted to specify the JDK home directory, enter the location of the `jre` binary. For example:

```
/opt/java1.3/jre
```

The location for other versions of the binary may be different.

- 17. When prompted by the installer, unmount the currently mounted disc.

- a. Log in as `root` user.
- b. Use the `pfs_umount` command. For example:

```
# pfs_umount /cdrom
```

where `/cdrom` is the CD-ROM mount point. Remove the disc.



18. If you must mount another disc to continue the installation of Oracle9i, proceed to [step a](#) below. Otherwise, skip to [step 19](#).
  - a. Insert the next disc in the CD-ROM drive.
  - b. Mount the CD. For example, to mount the CD to the `/cdrom` mount point, type:
 

```
# /usr/sbin/pfs_mount -t rrip /dev/dsk/c1t2d0 /cdrom
```
  - c. Proceed with the Oracle9i installation.
  - d. To unmount the disc, go back to [step 17](#).
19. When you complete installing Oracle9i Release 2 software, install the Oracle9i Release 2 patches; see “[Adding Oracle9i Release 2 Patches](#)” on page 80. If you are not installing patches, proceed to “[Linking to VERITAS Libraries](#)” on page 81.

## Method 2: Installing Oracle9i Release 2 on Local File System

Use this procedure to install Oracle9i Release 2 on each node locally in a VERITAS Storage Foundation *for Oracle RAC* environment:

1. Log in as `root` user on one node.
2. On one node, create a disk group:
 

```
# vxdg init or_dg c0t1d1
```
3. Create the volume in the group:
 

```
# vxassist -g or_dg make or_vol 5000M
```

For the Oracle9i Release 2 binaries, make the volume 5,000 MB.
4. Create a VxFS file system on `or_vol` to install the Oracle9i binaries. For example:
 

```
# mkfs -F vxfs -o largefiles /dev/vx/rdisk/or_dg/or_vol
```
5. Create the mount point for the file system:
 

```
# mkdir /oracle
```
6. Mount the file system using the device file for the block device:
 

```
# mount -F vxfs /dev/vx/dsk/or_dg/or_vol /oracle
```



7. Edit the `/etc/fstab` file and list the new file system. For example:

```
/dev/vx/dsk/or_dg/or_vol /oracle vxfs delaylog 0 1
```

8. Assign ownership of the Oracle directory to `oracle`:

```
# chown -R oracle:dba /oracle  
# chmod 775 /oracle
```

9. Repeat [step 1](#) through [step 8](#) on the other node.

If the `preinstallora` script fails, error messages inform you of necessary corrective actions.

10. On the first node, insert Disk1 of the Oracle9i discs in the CD-ROM drive.

- a. Log in as `root` user.

- b. Determine the appropriate CD-ROM device:

```
# ioscan -fn -C disk
```

For example, the listing may indicate the block device for the CD-ROM drive is `/dev/dsk/c1t2d0`; make a note of the device file as it applies to your system.

- c. Check whether the PFS (Portable File System) daemon is running:

```
# ps -ef | grep pfs_mountd
```

If the daemon is running, look for output resembling the following example:

```
root 2853 1825 0 14:49:31 pts/0 0:02 pfs_mountd
```

- d. If the daemon is not running, start it:

```
# nohup pfs_mountd &  
# nohup pfsd &
```

- e. If necessary, create a directory under which to mount the software disc:

```
# mkdir -p /cdrom
```

- f. Mount the CD-ROM. For example, to mount the CD-ROM to the mount point `/cdrom`, type:

```
# /usr/sbin/pfs_mount -t rrip /dev/dsk/c1t2d0 /cdrom
```

`/dev/dsk/c1t2d0` is the block device file for the drive.

- g. Verify the CD-ROM is mounted:

```
# mount
```



In the output, look for a line that resembles:

```
/cdrom on /dev/dsk/c1t2d0 ro on Wed Feb 26 15:11:32 2003
```

11. Log in as `oracle` user on each node.
12. On each node, edit the `.rhosts` file to provide the other node access to the local system during the installation. Place a "+" character in the first line of the file. You can remove this permission after the installation is complete.
13. On each node, create a directory for the installation of the Oracle9i binaries:

```
$ mkdir /oracle/VRT
```

14. On each node, set the following environment variables:

```
$ export ORACLE_BASE=/oracle
$ export ORACLE_HOME=/oracle/VRT
$ export LD_LIBRARY_PATH=$ORACLE_HOME/lib
$ export PATH=$PATH:$ORACLE_HOME/bin
$ export CLASSPATH=$CLASSPATH:$ORACLE_HOME/JRE:$ORACLE_HOME\
  /jlib:$ORACLE_HOME/rdbms/jlib:$ORACLE_HOME/network/jlib
```

15. On the first node, set the `DISPLAY` variable.

If you use the Bourne Shell (`sh` or `ksh`):

```
$ DISPLAY=host:0.0;export DISPLAY
```

If you use the C Shell (`csh` or `tcsh`):

```
$ setenv DISPLAY host:0.0
```

16. On the first node, run the Oracle9i `runInstaller` utility:

```
$ /cdrom/Disk1/runInstaller
```

- a. When `runInstaller` prompts you to run the `/tmp/orainstRoot.sh` script, make sure the script exists *on each node* before proceeding. If it does, proceed to [step c](#).
- b. If the `/tmp/orainstRoot.sh` script does not exist on each node, copy the script from the first node to each of the other cluster nodes.
- c. Run the `/tmp/orainstRoot.sh` script on each node.
- d. When the **Node Selection** dialog box appears, select all nodes for installation.



**Note** If the **Node Selection** dialog box does not appear, refer to “[Missing Dialog Box During Installation of Oracle9i Release 2](#)” on page 216.

---

- e. In the **Available Products** dialog box, select **Oracle 9i Database 9.2.0.1.0**.
  - f. In the **Installation Types** dialog box:
    - ◆ If you select **Enterprise Edition** or **Standard Edition**, make sure to select the **Software Only** installation option.
    - ◆ If you select **Custom Installation**, decline to build a database in the **Create Database** dialog box.
  - g. As the installer runs, it prompts you to designate the location for the Oracle SRVM component. Complete one of the following options:
    - ◆ Enter the name of a file, such as `ora_srvm`, located on the cluster file system. See “[Method 2: Creating a Cluster File System for SRVM \(Optional\)](#)” on page 70. For example:

```
    /orasrv/ora_srvm
```
    - ◆ To use the raw volume, enter `/dev/vx/rdisk/orasrv_dg/srvm_vol`. See “[Creating the Oracle9i User and Group](#)” on page 68.
  - h. When prompted to specify the **JDK Home Directory**, enter the location of the `jre` binary. For example:

```
    /opt/java1.3/jre.
```

The location for other versions of the binary may be different.
17. When prompted by the installer, unmount the currently mounted disc:
- a. Log in as `root` user.
  - b. Use the `pfs_umount` command. For example:

```
    # pfs_umount /cdrom
```

where `/cdrom` is the CD-ROM mount point. Remove the disc.
18. If you must mount another disc to continue the installation of Oracle9i, proceed to [step a](#). Otherwise, skip to [step 19](#).
- a. Insert the next disc in the CD-ROM drive.



- b.** Mount the disc. For example, to mount the disc to the `/cdrom` mount point, enter:

```
# /usr/sbin/pfs_mount -t rrip /dev/dsk/clt2d0 /cdrom
```

- c.** Continue with the Oracle9i installation.

- 19.** After installing Oracle9i Release 2 software, you can install the Oracle9i Release 2 patches; see “[Adding Oracle9i Release 2 Patches](#).” If you do not need to install patches, proceed to “[Linking to VERITAS Libraries](#)” on page 81.



## Adding Oracle9i Release 2 Patches

1. Log in as `oracle` user.
2. On one node, create the directory where you intend to copy the Oracle9i patch software. For example:

```
$ mkdir /oracle/patch
$ cd /oracle/patch
```
3. Copy all files included with the downloaded Oracle9i patch software to `/oracle/patch`. When you uncompress and untar the downloaded ZIP file, the software resides in a `Disk1` directory.
4. On each node, edit the `.rhosts` file to provide the other node access to the local system during the installation. Place a "+" character in the first line of the file. You can remove this permission after installation is complete.

5. Set the `DISPLAY` variable.

If you use the Bourne Shell (`sh` or `ksh`):

```
$ DISPLAY=host:0.0;export DISPLAY
```

If you use the C Shell (`csh` or `tcsh`):

```
$ setenv DISPLAY host:0.0
```

6. Run the Oracle9i `runInstaller` utility:

```
$ $ORACLE/patch/Disk1/runInstaller
```

7. As the `runInstaller` utility runs, note the following points:
  - ◆ Select `products.xml` from the `/oracle/patch/Disk1/stage` directory.
  - ◆ In the **Node Selection** dialog box:
    - ◆ When installing an Oracle9i Release 2 patch on a cluster file system, select the default node for installation.
    - ◆ When installing an Oracle9i Release 2 patch on a local file system, select all nodes for installation.

If you are installing Oracle9i Release 2 Patch 2 (9.2.0.2) on a local file system, install the software by invoking the installer on each node, one node at a time. Refer to the following document: *Oracle Metalink document ID: Note: 224598.1*. Search on: 224598.1 ALERT: Secondary 9.2.0.2 RAC Instances Will Not Mount After Relinking.

8. Proceed with the installation.



---

## Linking to VERITAS Libraries

1. At the completion of the patch installation, run the following command as `oracle` user to link Oracle with the VERITAS IPC library and enable Oracle to use the VERITAS ODM and cluster membership (VCSMM) libraries:

```
$ /opt/VRTSvcs/rac/bin/linkrac9i
```

---

**Note** If the Oracle binaries are on a cluster file system, perform this step on only one node. If the Oracle binaries are on a local file system of each cluster node, run the command to link Oracle with VERITAS libraries on each cluster node.

---

If your system uses a bundled C compiler or a compiler other than the ANSI C compiler, you can safely ignore such warnings as:

```
(Bundled) cc: warning 922: "+Oshortdata=8" is unsupported in the  
bundled compiler, ignored.
```

Refer to *Oracle Metalink Document ID 66442.1* for more information.  
Search on: 66442.1 FAQ about 'C' Compiler Issues on HP-UX.

2. After starting Oracle instances, confirm Oracle uses the VERITAS libraries. Examine the Oracle alert file, `alert_$(ORACLE_SID).log`, for the following lines:

```
Oracle instance running with ODM: VERITAS 4.1 ODM Library,  
Version 1.0  
cluster interconnect IPC version:VERITAS IPC 4.1 08:44:27 Apr 20  
2005
```

---

**Note** If Oracle binaries are on a local file system of each cluster node, examine the Oracle alert file on each of the cluster nodes.

---

To perform post-installation tasks, refer to the Oracle Patch Set Notes that accompany the patch.



## Disabling ODM Library (Optional) for Databases Using Raw Devices

By default, the `linkrac9i` command links all libraries, including the ODM library, with Oracle. You must use the ODM library for a database using a cluster file system; a VERITAS cluster file system requires the ODM library. If the database uses a raw device rather than a cluster file system, ODM is optional.

ODM supports kernel asynchronous I/O. If you choose to disable the ODM library, VERITAS recommends enabling the kernel asynchronous I/O driver of HP-UX by following the instructions in the *Oracle Administrator's Reference Release 2 for UNIX Systems* or the following document:

*Oracle Metalink Document Id. Note:139272.1*

Search on: 139272.1 HP-UX: Asynchronous i/o

The Oracle `DISK_ASYNC_IO` configuration parameter, which configures the Oracle database to use asynchronous I/O to reduce CPU utilization and improve transaction throughput, is set to `true` by default.

To disable the VERITAS ODM library:

1. Log in as `oracle` user.

2. Change directories:

```
$ cd $ORACLE_HOME/lib
```

3. Remove the file linked to the VERITAS ODM library.

For PA systems:

```
$ rm libodmd9.sl
```

For IA systems:

```
$ rm libodmd9.so
```

4. Copy the `$ORACLE_HOME/lib/libodmd9.sl` file to disable the ODM library.

For PA systems:

```
$ cp $ORACLE_HOME/lib/libodmd9.sl.oracle
   $ORACLE_HOME/lib/libodmd9.sl
```

For IA systems

```
$ cp $ORACLE_HOME/lib/libodmd9.so.oracle
   $ORACLE_HOME/lib/libodmd9.so
```

## Linking to VERITAS Libraries When Upgrading Oracle Patch Level

See [Appendix F. “Linking VERITAS Libraries to Oracle” on page 269](#) for details.

## Creating the Databases

Create the Oracle database on shared storage. Refer to “[Creating a Starter Database](#)” on page 265.

## Upgrading the Databases

If you currently have Oracle databases running and want to migrate them to the latest Oracle patch level, refer to the `README.html` file downloaded with the patch. The file is located in the `/oracle/patch` directory.

## Stopping Applications Not Under VCS Control

When running applications that are not under VCS control but use cluster file system mount points under VCS control, stop those applications any time you plan to stop VCS or shut down the nodes.

### Stopping gsd

Stop the `gsd` process if it starts. As `oracle` user, enter the command:

```
$ $ORACLE_HOME/bin/gsdctl stop
```

### Stopping Applications using CFS (Outside of VCS Control)

Stop all other applications using the CFS mounts outside of VCS control. Ensure that no processes use the CFS mount point:

```
# fuser -c mount_point
```

To stop processes using the CFS mount point, type:

```
# fuser -ck mount_point
```



## Unmounting CFS File Systems Outside of VCS Control

Unmount CFS file systems outside of VCS control:

1. Determine the file systems to unmount by checking the `/etc/mnttab` file. For example:

```
# cat /etc/mnttab | grep vxfs | grep cluster
```

The output shows each line of the `/etc/mnttab` file that contains an entry for a VxFS file system mounted in the cluster mode.

2. By specifying its mount point, unmount each of the file systems listed in the output:

```
# umount mount_point
```

# Configuring VCS Service Groups for Oracle9i

---

5

This chapter describes setting up VCS to automate the Oracle RAC environment. Below is a conceptual overview of how VCS manages resources within a cluster.

## Service Groups in SFRAC: Overview

The VERITAS Cluster Server (VCS) package, provided as part of the installation of *Storage Foundation for Oracle RAC*, provides the ability to automate the entire RAC environment. For example, VCS can be used to automatically start the Cluster Volume Manager and Cluster File System resources within the cluster, bring up IP addresses and the Oracle Listener, mount the file systems with the Oracle binaries, mount the storage for the database instances, and actually start the database instance. Placing the database under VCS control in no way removes the DBA's capability for full control. It simply automates actions to enable the cluster to start up after any outage.

In a *Storage Foundation for Oracle RAC* cluster, the administrative staff is free to choose how much, or how little automated control they desire. Less automation means more traditional hands-on interaction, but also requires the administrator take corrective action in more circumstances. A better idea may be to allow VCS complete startup control to take care of system and power failures and restarts, while still allowing manual control if necessary.

VCS uses installed agents to manage the resources in the *Storage Foundation for Oracle RAC* environment. Each type of resource has an agent; for example, VCS uses a CFSSMount agent for mounting shared file systems, the CVMVolDg agent for activating shared disk groups and monitoring shared volumes, an Oracle agent for starting the Oracle database, an IP agent for setting up and monitoring an IP address, and so on.

The VCS configuration file (`/etc/VRTSvcs/conf/config/main.cf`) contains the information the agents require to manage each resource in the environment. In addition, the configuration file specifies the dependencies between resources; the dependencies that one resource has upon another sets the order in which the resources are started or stopped by the agents.

Within VCS configurations, resources exist in *service groups*. A service group consists of resources that must function together in support of a service. For example, an Oracle service group can consist of the Oracle database, a CFSSMount resource for the shared file



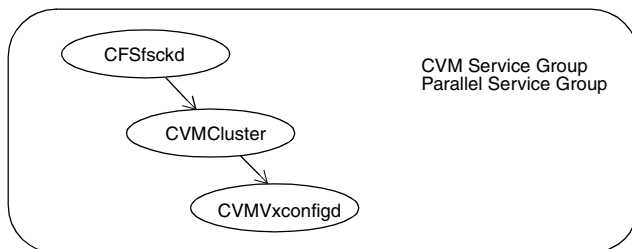
system containing the database, and a CVMVolDg resource for the shared disk group that contains the file system. Just as resources have dependencies upon other resources, service groups can have dependencies on other service groups.

## The CVM Service Group

After the installation of *Storage Foundation for Oracle RAC*, a basic CVM service group is created automatically. The CVM service group consists of resources for the CVM and CFS daemons (CVMVxconfigd and CFSfsckd) and the CVMCluster resource. If for any reason this group has not been created, you may create it at any time by running the command:

```
/opt/VRTSvxfs/cfs/bin/cfscluster config
```

The CVM service group must exist in the `/etc/VRTSvcs/conf/config/main.cf` file to enable CVM administrative commands (such as `vxclustadm`) to function. Do not delete or rename this service group. Refer to the [“Sample main.cf: After SFRAC Installation and Before Oracle9i Installation”](#) on page 63.



When you have installed Oracle RAC and created a database, you may modify the CVM service group to implement control of other core cluster infrastructure components, such as the Oracle listener and Oracle binaries. Most of these choices are covered in the rest of this chapter.



---

## Configuring Service Groups to Automate the RAC Environment

VCS can be configured to automate the Oracle RAC environment as well. Depending on installed configuration and level of control desired, the implementer can create service groups to bring up the ORACLE\_HOME and listener, mount database storage, as well as start and stop Oracle database instances.

There are a number of configuration choices that affect the overall cluster design. These choices will be discussed briefly in the next several paragraphs.

### Is Oracle Installed on Shared or Local Storage?

You can install the Oracle binary on shared storage, that is, in a cluster file system, where it is shared by cluster systems, or you can install it on each system in a local file system.

When the Oracle binaries are installed locally, the file system for ORACLE\_HOME may be mounted as part of the normal boot process. If this is the case, no VCS involvement is necessary.

When Oracle binaries are installed on a cluster file system, CVM and CFS need to be online before ORACLE\_HOME is mounted. Placing the ORACLE\_HOME resource under VCS control allows proper dependency relationships to be built for proper start up sequencing. The CFSMount agent and CVMVolDg agents are used for control.

---

**Note** Make sure that if the Oracle is installed on shared disk (CFS), that you use a different file system for the database.

---

### Is SRVM Located in ORACLE\_HOME or a Separate Disk Group?

The SRVM component must be installed on shared storage if the user wishes to use any of the Oracle supplied tools to manage the database. The SRVM configuration files can be placed on either a cluster file system or a shared raw volume. The preferred method is to configure SRVM as a directory on the shared ORACLE\_HOME. Alternate locations for the SRVM component include a shared volume or shared file system in a separate disk group.

### Does the Listener Use a Virtual IP Address?

The Listener may be configured to use a virtual IP address or host IP address.



## Applications to Address RAC Databases on the Same Server?

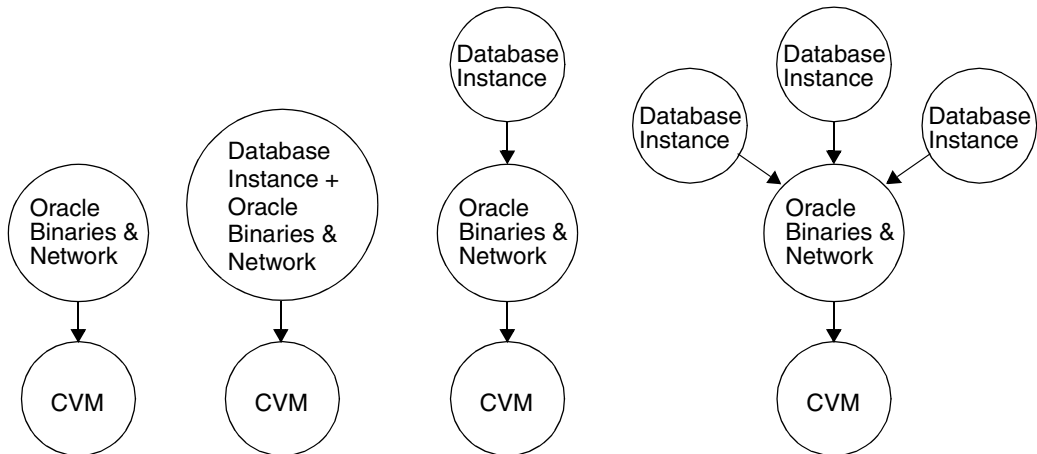
If the user intends to create service groups to manage applications accessing the RAC instances in the cluster, special consideration may have to be given to service group layout.

## Understanding VCS Service Group Dependency Capabilities and Limitations

VCS supports dependencies between service groups. For example, a service group containing a RAC instance may require another service group containing the Oracle binaries and listener. This group may in turn require the CVM group be online to provide CVM and CFS support.

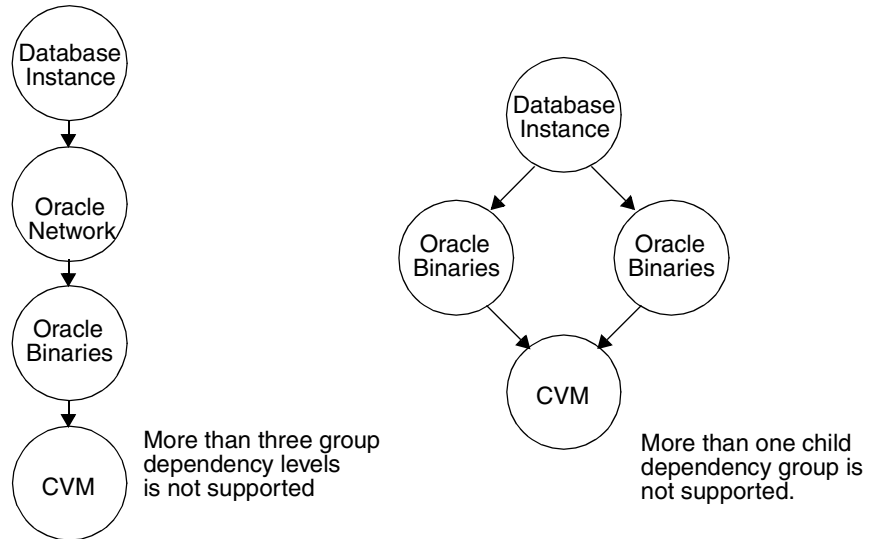
The following illustration shows various supported service group configurations. The circles represent parallel service groups running on *each* RAC node. For example, the left most configuration has the Oracle binaries and network component in a service group that depends on the CVM group for necessary daemon support for CVM and CFS. The next configuration shows a database added to the service group. The two example configurations to the right show better arrangements, because they allow for multiple service groups containing individual databases to be independently managed.

Supported Service Group Configurations



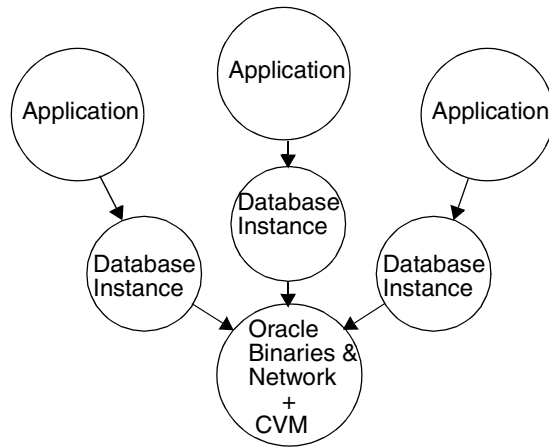
VCS supports a maximum of three dependency levels deep. It also supports only one child dependency group; that is, one service group cannot depend upon two others. The following diagram shows unsupported service group dependency configurations.

Service Group Configurations Not Supported by VCS



For users wishing to create additional service groups to manage applications and stay within the dependency limitations of VCS, VERITAS recommends moving all cluster infrastructure components, including ORACLE\_HOME and the listener to the CVM service group.

Supported Configuration: Infrastructure Components in the CVM Service Group



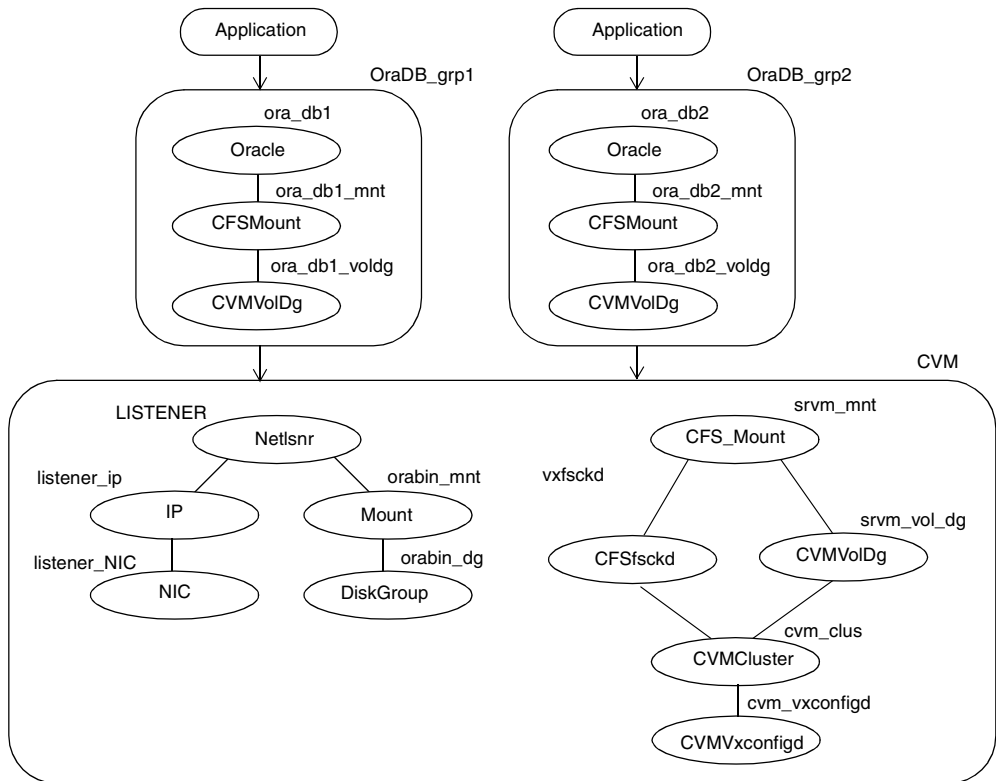
## Typical Configurations for Oracle and CVM Service Groups

The following table shows typical configurations for the Oracle and CVM service groups. In the example RAC environments that follow, the Oracle service groups are identical. The variations in the configuration, described in the following table, occur in the CVM group. The illustrations that follow show the dependencies among the service groups and the components.

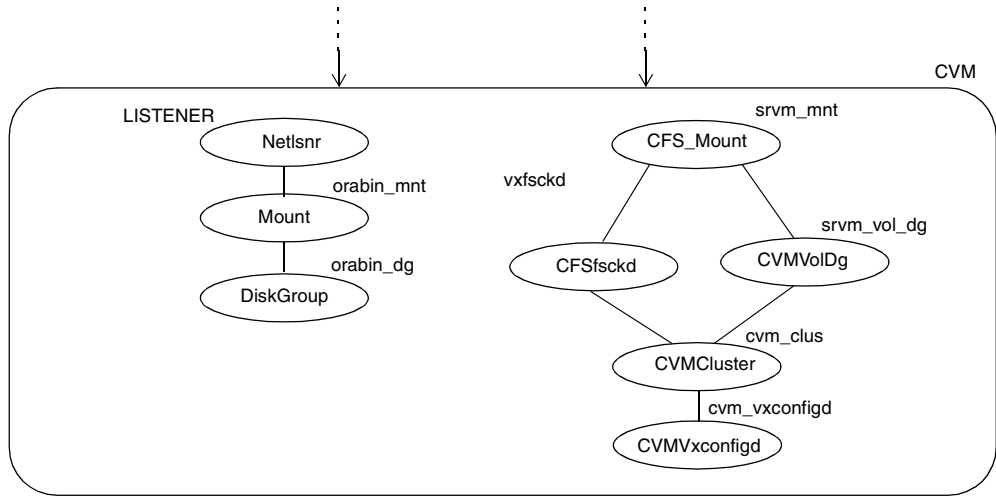
<b>Oracle Binary Installed: Locally or on CFS</b>	<b>SRVM Location on Shared Storage: Own Disk Group or With Oracle</b>	<b>IP for Listener: Host or Virtual</b>
Locally	Shared File System	Virtual IP
Locally	Shared File System	Host IP
Locally	Shared Volume	Host IP
CFS	ORACLE_HOME subdirectory	Host IP
CFS	Separate volume/file system	Host IP
CFS	ORACLE_HOME subdirectory	Virtual IP
CFS	Separate volume/file system	Virtual IP



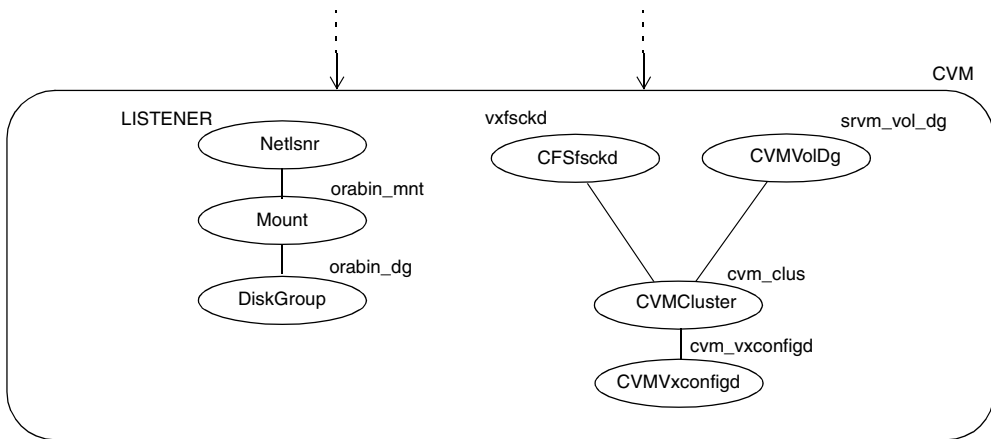
Typical Service Groups: Oracle Installed Locally; SRVM on Shared File System; Virtual IP Used



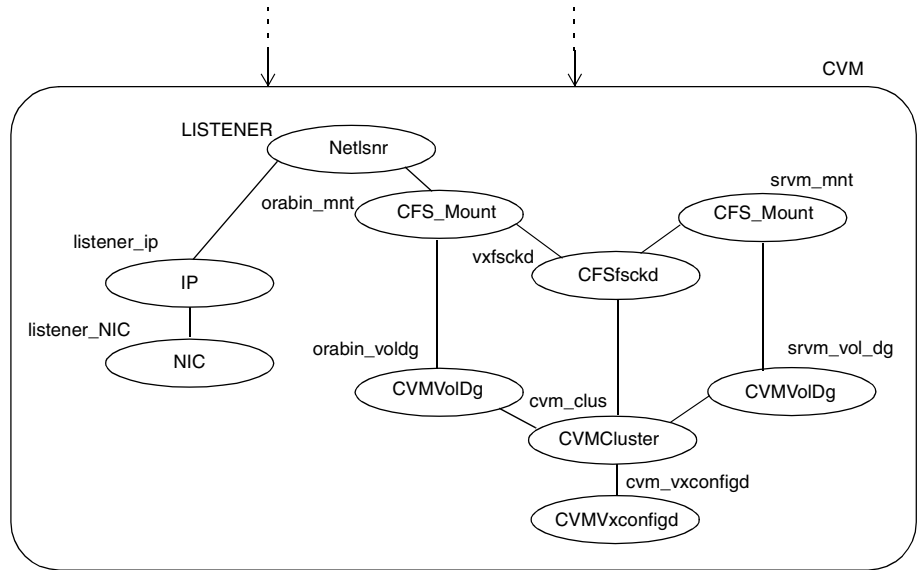
CVM Group with Oracle Installed Locally; SRVM on Shared File System; Host IP Used



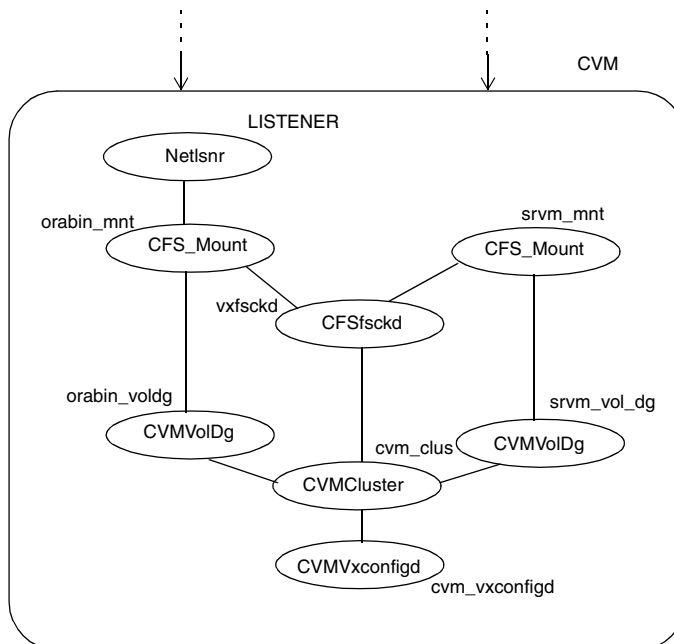
CVM Group with Oracle Installed Locally; SRVM on Shared Volume; Host IP Used



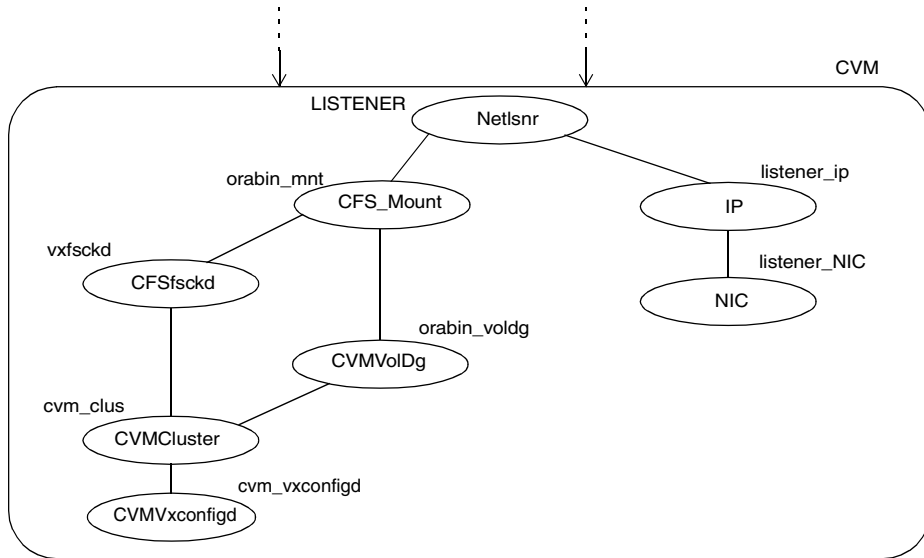
CVM Group with Oracle Installed on CFS; SRVM Configured in Separate File System; Virtual IP



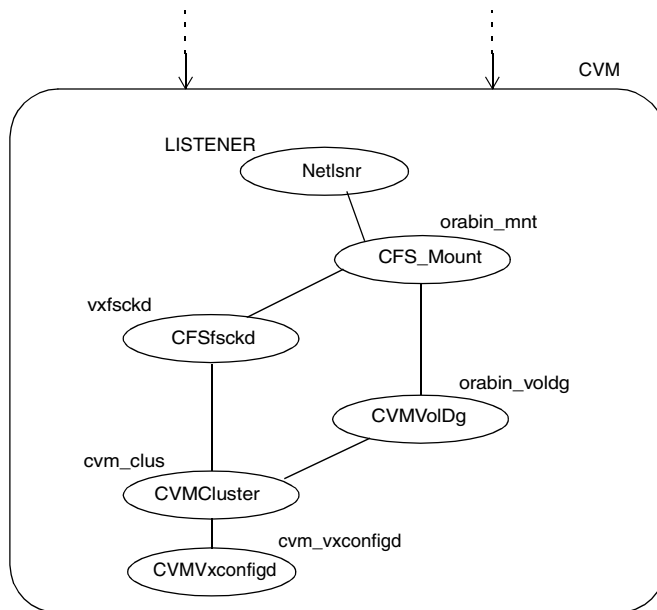
CVM Group with Oracle Installed on CFS; SRVM Configured in Separate File System; Host IP



CVM Group with Oracle Binaries and SRVM on CFS; Virtual IP



CVM Group with Oracle Binaries and SRVM on CFS; Host IP





## Configuring CVM and Oracle Service Groups

The CVM and Oracle service groups can be configured using the following two methods:

- ◆ By editing the VCS configuration file, `main.cf`, to define the service groups; see [“Configuring CVM and Oracle Service Groups Manually”](#) on page 95.
- ◆ By using a configuration wizard for Oracle RAC; see [“Configuring the Service Groups Using the Wizard”](#) on page 105.

### Configuring CVM and Oracle Service Groups Manually

You can edit the `main.cf` file directly to configure the CVM and Oracle service groups. If you would prefer to use the VCS RAC wizard to perform the configuration, see [“Configuring the Service Groups Using the Wizard”](#) on page 105.

1. Log in to one system as `root`.
2. Save your existing configuration to prevent any changes while you modify `main.cf`:

```
# haconf -dump -makero
```

3. Ensure VCS is not running while you edit `main.cf` by using the `hastop` command to stop the VCS engine on all systems and leave the resources available:

```
# hastop -all -force
```

4. Make a backup copy of the `main.cf` file:

```
# cd /etc/VRTSvcs/conf/config
# cp main.cf main.orig
```

Using `vi` or another text editor, edit the `main.cf` file and modify the `cvm` service group and create Oracle service groups using the guidelines in the following sections.

Two example configuration are shown for modifying the CVM service group: [“Example 1: CVM Group with Oracle Binaries on CFS,”](#) shown below, and [“Example 2: CVM Group with Oracle Binaries Installed Locally”](#) on page 98. Adding the Oracle database service group is the same for either configuration. See [“Creating Oracle Service Group in the main.cf File”](#) on page 101

#### Example 1: CVM Group with Oracle Binaries on CFS

For this example, you can refer to the illustration shown in [“CVM Group with Oracle Binaries and SRVM on CFS; Virtual IP”](#) on page 94.



## Modifying the CVM Service Group in the main.cf File

The `cvm` service group is created during the installation of Storage Foundation for Oracle RAC. After installation, the `main.cf` file resembles the example shown in [“Sample main.cf: After SFRAC Installation and Before Oracle9i Installation”](#) on page 63. Because Oracle had not been installed, the `cvm` service group includes only resources for the `CFSfsckd` daemon and the `CVMCluster` resource.

You must modify the `cvm` service group to add the `Netlsnr`, `NIC`, `IP`, `CVMVolDg`, and `CFSMount` resources to the configuration. You can refer to [“Sample Configuration Files”](#) on page 229 to see a complete example of how a `cvm` group is configured.

1. Make sure the `cvm` group has the group `Parallel` attribute set to 1. Typically this is already done during installation.

```
.
group cvm (
    SystemList = { galaxy = 0, nebula = 1 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { galaxy, nebula }
)
.
```

2. Define the `NIC` and `IP` resources. The VCS bundled `NIC` and `IP` agents are described in *VERITAS Cluster Server Bundled Agents Reference Guide*. The device name and the `IP` addresses are required by the listener for public network communication. Note that for the `IP` resource, the `Address` attribute is localized for each node (see [“Attributes of CVM, Oracle Groups Defined as Local”](#) on page 104).

```
.
    NIC listener_NIC (
        Device = lan0
        NetworkType = ether
    )

    IP listener_ip (
        Device = lan0
        Address @galaxy = "192.2.40.21"
        Address @nebula = "192.2.40.22"
    )
.
```

3. Define the `Netlsnr` resource. The `Netlsnr` listener agent is described in detail in the *VERITAS Cluster Server Enterprise Agent for Oracle, Installation and Configuration Guide*. Note that the `listener` attribute is localized for each node.

```
.
```



```

Netlsnr LISTENER (
  Owner = oracle
  Home = "/oracle/orahome"
  TnsAdmin = "/oracle/orahome/network/admin"
  MonScript = "./bin/Netlsnr/LsnrTest.pl"
  Listener @galaxy = LISTENER_a
  Listener @nebula = LISTENER_b
  EnvFile = "/opt/VRTSvcs/bin/Netlsnr/envfile"
)

```

4. You must configure the CVMVolDg and CFSSMount resources in the `cvm` group for the Oracle binaries installed on shared storage. Refer to the appendix “[CVMCluster, CVMVxconfigd, CVMVolDg, and CFSSMount Agents](#)” on page 243 for description of CVMVolDg and CFSSMount agents.

```

CVMVolDg orabin_voldg (
  CVMDiskGroup = orabindg
  CVMVolume = { "orabinvol", "srvmvol" }
  CVMActivation = sw
)

CFSSMount orabin_mnt (
  Critical = 0
  MountPoint = "/oracle"
  BlockDevice = "/dev/vx/dsk/orabindg/orabinvol"
  Primary = galaxy
)

```

5. Define the dependencies of resources in the group. The dependencies are specified such that the Netlsnr resource requires the IP resource that, in turn, depends on the NIC resource. The Netlsnr resource also requires the CFSSMount resource. The CFSSMount resource requires the daemons and vxfsckd used by the cluster file system. The CFSSMount resource also depends on the CVMVolDg resource that, in turn, requires the CVMCluster resource, which autoimports the shared storage and establishes GAB communications within the cluster. The CVMCluster resource depends on the CVMVxconfigd daemon to maintain the disk and disk group configuration.

```

cvm_clus require cvm_vxconfigd
orabin_voldg requires cvm_clus
orabin_mnt requires vxfsckd

```



```

orabin_mnt requires orabin_voldg
listener_ip requires listener_NIC
LISTENER requires listener_ip
LISTENER requires orabin_mnt

```

```

.
.

```

### Creating the Oracle Service Group and Adding Resources

The procedure for adding Oracle database resources is identical for the two example configurations shown. See [“Creating Oracle Service Group in the main.cf File”](#) on page 101.

### Example 2: CVM Group with Oracle Binaries Installed Locally

For this example, you can refer to the illustration shown in [“CVM Group with Oracle Installed Locally; SRVM on Shared File System; Host IP Used”](#) on page 92.

### Modifying the CVM Service Group in the main.cf File

The `cvm` service group is created during the installation of Storage Foundation for Oracle RAC. After installation, the `main.cf` file resembles the example shown in [“Sample main.cf: After SFRAC Installation and Before Oracle9i Installation”](#) on page 63. Because Oracle had not been installed, the `cvm` service group includes only resources for the `CFSfsckd` daemon and `CVMCluster` resource.

You must modify the `cvm` service group to add the `Netlsnr`, `Mount`, `Disk Group`, `CVMVolDg`, and `CFSMount` resources to the configuration. You can refer to [Appendix A, “Sample Configuration Files”](#) on page 229, to see a complete example of how a `cvm` group is configured.

1. Make sure the `cvm` group has the group `Parallel` attribute set to 1. Typically this is already done during installation.

```

.
group cvm (
    SystemList = { galaxy = 0, nebula = 1 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { galaxy, nebula }
)
.

```

2. Define the `Mount` and `DiskGroup` resources for use by the Oracle binaries. The VCS bundled `Mount` and `DiskGroup` agents are described in *VERITAS Cluster Server Bundled Agents Reference Guide*.



```

Mount orabin_mnt (
    MountPoint = "/prod/u01"
    BlockDevice = "/dev/vx/dsk/ora_bin_dg/ora_bin"
    FSType = vxfs FsckOpt = "-n"
)

DiskGroup orabin_dg (
    DiskGroup = ora_bin_dg
    StartVolumes = 0
    StopVolumes = 0
)

```

3. Define the Netlsnr resource. The Netlsnr listener agent is described in detail in the *VERITAS Cluster Server Enterprise Agent for Oracle, Installation and Configuration Guide*. Note that the Listener attribute is localized for each node.

```

Netlsnr LISTENER (
    Owner = oracle
    Home = "/oracle/orahome"
    TnsAdmin = "/oracle/orahome/network/admin"
    MonScript = "./bin/Netlsnr/LsnrTest.pl"
    Listener @galaxy = LISTENER_a
    Listener @nebula = LISTENER_b
    EnvFile = "/opt/VRTSvcs/bin/Netlsnr/envfile"
)

```

4. You must configure the CVMVolDg and CFSSMount resources in the cvm group for the SRVM disk group configured on shared storage. Refer to the appendix “[CVMCluster, CVMVxconfig, CVMVolDg, and CFSSMount Agents](#)” on page 243 for description of CVMVolDg and CFSSMount agents.

```

CFSSMount ora_srv_mnt (
    Critical = 0
    MountPoint = "/orasrv"
    BlockDevice = "/dev/vx/dsk/orasrv_dg/srvm_vol"
    Primary = galaxy
)

CVMVolDg srvm_vol_dg (
    CVMDiskGroup = orasrv_dg
)

```



```
        CVMVolume = { "srvmvol" }
        CVMActivation = sw
    )
.
.
```

5. Define the dependencies of resources in the group. The Netlsnr resource requires the Mount resource. The Mount resource requires DiskGroup resource. The CFSMount resource depends on the CVMVolDg resource, which, in turn, requires the CVMCluster resource, which autoimports the shared storage and establishes GAB communications within the cluster. The CFSMount resource also requires the CFSfsckd resource daemon, vxfsckd, used by the cluster file system. The CVMCluster resource depends on the CVMVxconfigd daemon to maintain the disk and disk group configuration.

```
.
.
LISTENER requires orabin_mnt
orabin_mnt requires orabin_dg
srvm_mnt requires srvm_vol_dg
srvm_mnt requires vxfsckd
srvm_vol_dg requires cvm_clus
cvm_clus requires cvm_vxconfigd
.
.
```

### Creating the Oracle Service Group and Adding Resources

The procedure for adding Oracle database resources is identical for the two example configurations shown. See the next section, [“Creating Oracle Service Group in the main.cf File.”](#)

## Creating Oracle Service Group in the main.cf File

For a complete description of the VCS Oracle enterprise agent, refer to the document, *VERITAS Cluster Server Enterprise Agent for Oracle, Installation and Configuration Guide*. That document includes instructions for configuring the Oracle and Netlsnr agents.

The VCS Enterprise Agent for Oracle version 4.1 is installed when you run the installer script. When you refer to the *VERITAS Cluster Server Enterprise Agent for Oracle Installation and Configuration Guide*, ignore the steps described in the section “Installing the Agent Software.”

1. Using the “[Sample Configuration Files](#)” on page 229 as an example, add a service group to contain the resources for an Oracle database. For example, add the group `oradb1_grp`. Make sure you assign the `Parallel` attribute a value of 1.

```
.
group oradb1_grp (
    SystemList = { galaxy = 0, nebula = 1 }
    AutoFailOver = 1
    Parallel = 1
    AutoStartList = { galaxy, nebula }
)
.
```

2. Create the `CVMVolDg` and `CFSMount` resource definitions. See “[CVMCluster, CVMVxconfigd, CVMVolDg, and CFSMount Agents](#)” on page 243 for a description of these agents and their attributes.

```
.
.
    CVMVolDg ora_db1_voldg (
        CVMDiskGroup = oradb1dg
        CVMVolume = { "oradb1vol" }
        CVMActivation = sw
    )

    CFSMount ora_db1_mnt (
        MountPoint = "/oradb1"
        BlockDevice = "/dev/vx/dsk/oradb1dg/oradb1vol"
        Primary = galaxy
    )
.
```



3. Define the Oracle database resource. Refer to the *VERITAS Cluster Server Enterprise Agent for Oracle, Installation and Configuration Guide* for information on the VCS enterprise agent for Oracle. Note that the Oracle attributes `Sid`, `Pfile`, and `Table` attributes must be set locally, that is, they must be defined for each cluster system.

```

.
.
    Oracle VRT (
        Sid @galaxy = VRT1
        Sid @nebula = VRT2
        Owner = oracle
        Home = "/oracle/orahome"
        Pfile @galaxy = "/oracle/orahome/dbs/initVRT1.ora"
        Pfile @nebula = "/oracle/orahome/dbs/initVRT2.ora"
        User = scott
        Pword = tiger
        Table @galaxy = vcstable_galaxy
        Table @nebula = vcstable_nebula
        MonScript = "./bin/Oracle/SqlTest.pl"
        AutoEndBkup = 1
        EnvFile = "/opt/VRTSvcs/bin/Oracle/envfile"
    )
.
.

```

4. Define the dependencies for the Oracle service group. Note that the Oracle database group is specified to require the `cvm` group, and that the required dependency is defined as “online local firm,” meaning that the `cvm` group must be online and remain online on a system before the Oracle group can come online on the same system. Refer to the *VERITAS Cluster Server User’s Guide* for a description of group dependencies.

Refer to the illustration in [“Understanding VCS Service Group Dependency Capabilities and Limitations”](#) on page 88.

```

.
.
    requires group cvm online local firm
    oradb1_mnt requires oradb1_voldg
    VRT requires oradb1_mnt
.
.

```

See the [“Sample Configuration Files”](#) on page 229 for a complete example. You can also find the complete file in `/etc/VRTSvcs/conf/sample_rac/main.cf`.

When you complete editing the `main.cf` file, see the next section, [“Saving and Checking the Configuration.”](#)





## Saving and Checking the Configuration

When you complete editing the `main.cf` file to configure the CVM and Oracle service groups, verify the new configuration.

1. Save and close the `main.cf` file.
2. Verify the syntax of the file `/etc/VRTSvcs/conf/config/main.cf`:  

```
# hacf -verify /etc/VRTSvcs/conf/config
```
3. Start the VCS engine on one system:  

```
# hastart
```
4. Type the command `hastatus`:  

```
# hastatus
```
5. When “LOCAL\_BUILD” is listed in the message column, start VCS on the other system:  

```
# hastart
```
6. Verify that the service group resources are brought online. On one system, enter:  

```
# hagrps -display
```

## Additional RAC Processes Monitored by the VCS Oracle Agent

For shallow monitoring, the VCS Oracle agent monitors the Oracle9i RAC processes `ora_lmon` and `ora_lmd`, in addition to the processes `ora_dbw`, `ora_smon`, `ora_pmon`, and `ora_lgwr`.



## Attributes of CVM, Oracle Groups Defined as Local

The following table lists attributes that must be defined as local for the CVM and Oracle service groups (note that each attribute has string-scalar as the type and dimension).

Resource	Attribute	Definition
IP	Address	The virtual IP address (not the base IP address) associated with the interface. For example: <pre>Address @sysa = "192.2.40.21" Address @sysb = "192.2.40.22"</pre>
Netlsnr	Listener	The name of the Listener. For example: <pre>Listener @sysa = LISTENER_a Listener @sysb = LISTENER_b</pre>
Oracle	Sid	The variable \$ORACLE_SID represents the Oracle system ID. For example, if the SIDs for two systems, <i>sysa</i> and <i>sysb</i> , are VRT1 and VRT2 respectively, their definitions would be: <pre>Sid @sysa = VRT1 Sid @sysb = VRT2</pre>
Oracle	Pfile	The parameter file: \$ORACLE_HOME/dbs/init\$ORACLE_SID.ora. For example: <pre>Pfile @sysa = "/oracle/VRT/dbs/initVRT1.ora" Pfile @sysb = "/oracle/VRT/dbs/initVRT2.ora"</pre>
Oracle	Table	The table used for in-depth monitoring by User/PWord on each cluster node. For example: <pre>Table @sysa = vcstable_sysa Table @sysb = vcstable_sysb</pre> <p>Using the same table on all Oracle instances is not recommended. Using the same table generates VCSIPC traffic and could cause conflicts between the Oracle recovery processes and the agent monitoring processes in accessing the table.</p> <p><b>Note</b> Table is only required if in-depth monitoring is used. If the PWord varies by RAC instance, it must also be defined as local.</p>

If other attributes for the Oracle resource differ for various RAC instances, define them locally as well. These other attributes may include the Oracle resource attributes *User*, *PWord*, the CVMVolDg resource attribute *CVMActivation*, and others.



## Modifying the VCS Configuration

For additional information and instructions on modifying the VCS configuration, refer to the *VERITAS Cluster Server User's Guide*.

## Location of VCS and Oracle Agent Log Files

On all cluster nodes, look at the following log files for any errors or status messages:

```
/var/VRTSvcs/log/engine_A.log  
/var/VRTSvcs/log/Oracle_A.log  
/var/VRTSvcs/log/Netlsnr_A.log
```

When large amounts of data are written, multiple log files may be required. For example, `engine_B.log`, `engine_C.log`, and so on, may be required. The `engine_A.log` contains the most recent data.

## Configuring the Service Groups Using the Wizard

You can use a configuration wizard to configure the VCS service groups for Storage Foundation for Oracle RAC environment. The wizard enables you to create the service group for Oracle and modify the CVM service group.

### Creating Oracle Service Group - Overview

The Oracle9i RAC configuration wizard guides you through the creation of an Oracle service group and the definition of the Oracle, CFSMount, and CVMVolDg resources. It adds the Netlsnr resources to the existing CVM group. If the listeners use the virtual IP, the wizard also adds the IP and NIC resources to the CVM group.

The wizard configures the Oracle service group to depend on CVM group with an online-local-firm dependency.

### Before Starting the Wizard

Before starting the Wizard, you can verify that your Oracle installation can be configured. Review the requirements listed below. Also, you need to provide the wizard information as it proceeds. Make sure you have that information at hand.

#### Prerequisites

- ✓ Oracle RAC instances and listeners must be running on all cluster nodes.
- ✓ The database files of all instances must be on a cluster file system.



---

**Note** The Wizard does not support using the same file system for the Oracle binary and Oracle datafiles.

---

- ✓ The SRVM location must be on a raw volume or a cluster file system.
- ✓ Each Oracle instance must be associated with a listener. The listener may be configured to listen to either the base IP or a virtual IP.

---

**Note** The RAC configuration wizard requires that for the default listener, the listener parameter file, `listener.ora`, must reside in `$ORACLE_HOME/network/admin`. No such restriction applies for non-default listeners.

---

- ✓ The IP addresses and host names specified in the files `listener.ora` and `tnsnames.ora` must be the same.
- ✓ If detail monitoring is to be used for a database instance, the table used for detail monitoring must be set up, with user and password assigned.

#### Information Required From the User

- ✓ The names of the database instances to be configured
- ✓ The information to required for the detail monitoring configuration
- ✓ The location of the Pfile for each instance

## Establishing Graphical Access for the Wizard

The configuration wizard requires graphical access to the VCS systems where you want to configure service groups. If your VCS systems do not have monitors, or if you want to run the wizards from a remote HP system, do the following:

### Establishing Graphical Access from a Remote System

1. From the remote system, (`jupiter`, for example), run `xhost +`  

```
# xhost +
```
2. Complete one of the following operations (depending on your shell):
  - ◆ If you are running `ksh`, run this step on one of the systems where the wizard will run (for example, `jupiter`):  

```
# export DISPLAY=jupiter:0.0
```
  - ◆ If you are running `csh`, run this step  

```
# setenv DISPLAY jupiter:0.0
```



3. Verify the DISPLAY environment variable is updated:

```
# echo $DISPLAY
jupiter:0.0
```

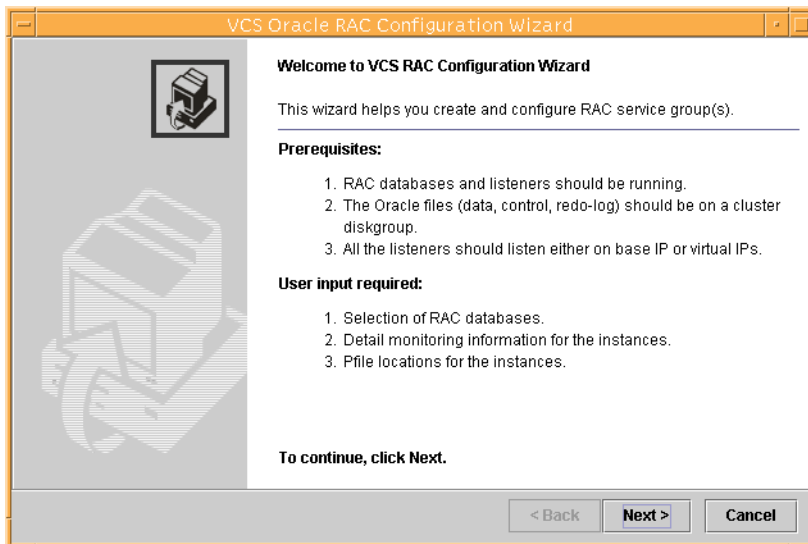
4. Make sure to set the JRE\_HOME variable to /opt/VRTSjre/jre1.4. If VRTSjre1.4 is not installed, the hawizard exits after displaying an error message.

## Starting the Configuration Wizard

The configuration wizard for Oracle 9i RAC is started at the command line.

1. Log on to one of your VCS systems as root.
2. Start the configuration wizard.

```
# /opt/VRTSvcs/bin/hawizard rac
```



## The Welcome Window

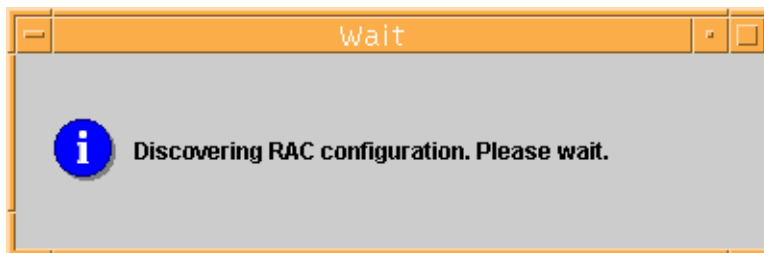
The RAC wizard starts with a Welcome window that highlights the prerequisites for configuration and the information required of the user to complete the configuration.

If your configuration does not meet the configuration requirements, you can stop the wizard by pressing Cancel. Take the necessary steps to meet the requirements and start the wizard again (see [“Starting the Configuration Wizard”](#) above).



## The Wizard Discovers the RAC Configuration

If you are ready to configure Oracle service group, press Next on the Welcome screen. The wizard begins discovering the current Oracle RAC information before proceeding with the next screen.



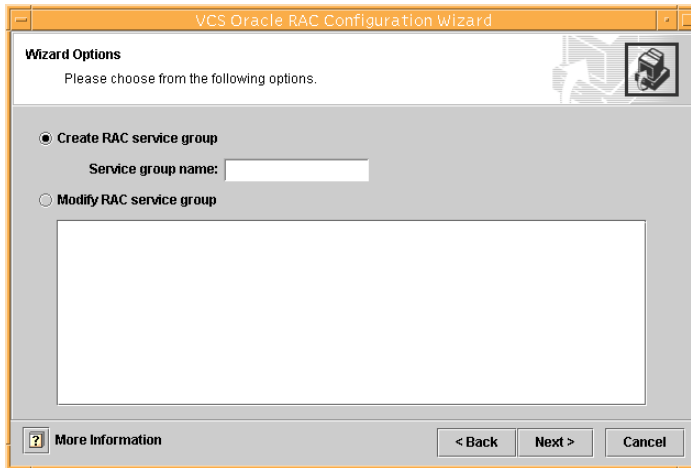
If the wizard does not find all databases and listeners running on all systems in the cluster, it halts with an error, indicating the problem. Press Cancel, and start the wizard again after you correct the problem.

## The Wizard Options Screen

The first configuration screen presents options to “Create RAC service group” or “Modify RAC service group.” If you are using the Wizard to modify an existing service group, see [“Modifying the Oracle 9i RAC Service Group Configuration”](#) on page 118.

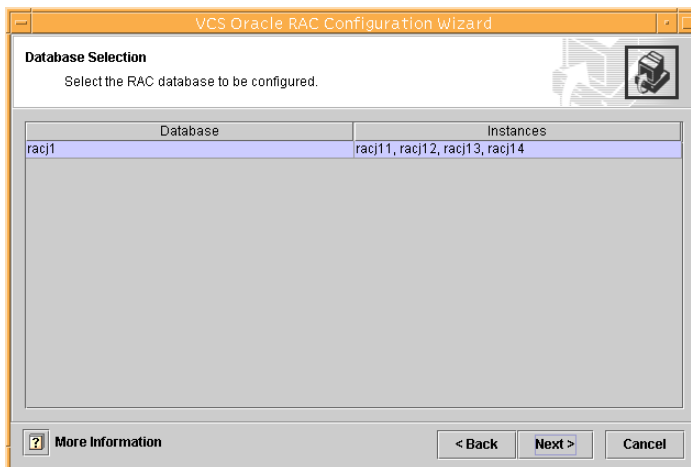
## Creating a RAC Service Group

To create an Oracle RAC service group, click the corresponding button and provide a name for the Oracle service group. Guidelines for naming an Oracle service group are available by clicking Help (“?”). After entering a service group name, click Next.



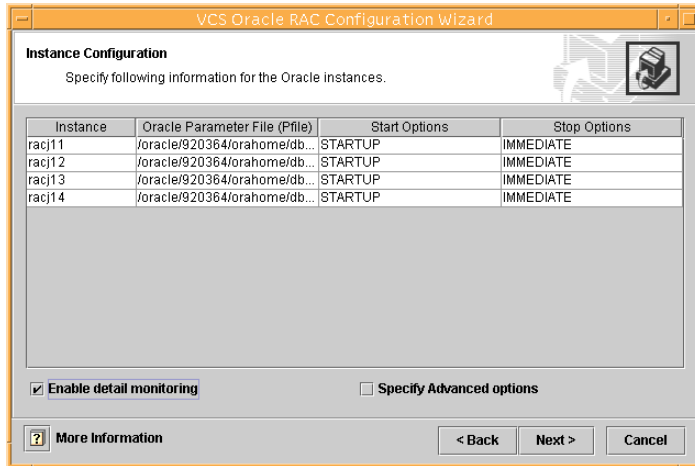
## Database Selection Screen

The databases and their instances running on the cluster are listed on this screen. Highlight only one of the databases if more than one are listed. Click Next.



## Instance Configuration Screen

Configure the basic database instance information on the Instance Configuration screen.



### Confirming Basic Database Instance Information

For each database instance discovered, basic configuration information is displayed. If necessary, double click in a field to select and edit its contents.

- ◆ Instance name: Each instance is listed in the left hand column.
- ◆ The Oracle Parameter File (Pfile): The file that is used to start Oracle. The default location for a given instance is listed. Edit the information if necessary.
- ◆ Startup Option:
 

Accept the displayed STARTUP option, or select an option from the drop-down menu. The startup options include starting in RESTRICTED, RECOVERDB, SRVCTLSTART, or CUSTOM modes.
- ◆ Stop Option:
 

Accept the displayed IMMEDIATE option, or select an option from the drop-down menu. The stop options also include TRANSACTIONAL, SRVCTLSTOP, and CUSTOM.

### Check box for Detailed Monitoring

On the Instance Configuration Screen, you can choose to enable Detail Monitoring. If you check Enable Detail Monitoring, be sure you have previously set up the database table, user, and password for the agent to use during monitoring. See [“Detail Monitoring Screen - Oracle Resource.”](#) If you are not set up for detail monitoring, do not select it.





## Check box for Advanced Options

On the Instance Configuration Screen, you can choose to Specify Advanced Options. The advanced options include setting up an `EnvFile` (to define environment variables), Encoding, and an `AutoEndBkup` parameter. See [“Oracle Advanced Configuration Screen.”](#)

## Detail Monitoring Screen - Oracle Resource

This screen is displayed if you have checked Detail Monitoring at the bottom of the Instance Configuration Screen.

Select	Sid	User	Password	Table
<input checked="" type="checkbox"/>	racj11	oracle	*****	tb1
<input checked="" type="checkbox"/>	racj12	oracle	*****	tb1
<input checked="" type="checkbox"/>	racj13	oracle	*****	tb1
<input checked="" type="checkbox"/>	racj14	oracle	*****	tb1

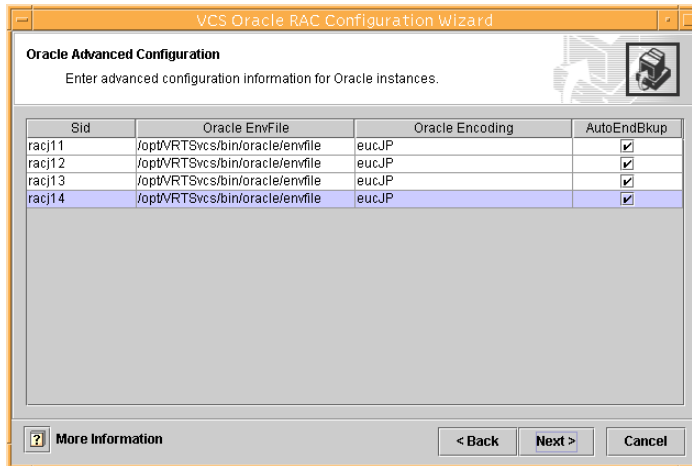
For each database instance identified by its Sid, this screen displays fields for defining the attributes that enable detail monitoring of the Oracle database resource. You do not have to enable detail monitoring on all instances, but for each instance you check, all of the fields are required:

- ◆ User: Oracle user, which the Oracle agent uses to log on to monitor the health of the database.
- ◆ Password: Password for the Oracle user.
- ◆ Table: Name of the database table to be used by the Oracle agent monitor.



## Oracle Advanced Configuration Screen

This screen is displayed if you have checked Specify Advance Options at the bottom of the Instance Configuration Screen.



For each database instance identified by its Sid, this screen displays fields for configuring the advanced attributes of the Oracle service group. You may select which database instance you want to configure advance attributes for, and which attributes you want to define.

The advance attributes include:

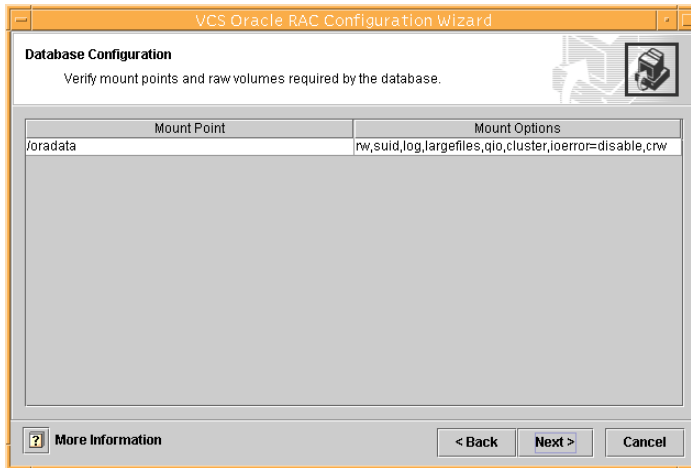
- ◆ EnvFile: the source file used by the agent entry point scripts
- ◆ Encoding: the operating system encoding that corresponds to Oracle encoding for the displayed Oracle output; the encoding value must match the encoding value used by the Netlsnr configuration.
- ◆ AutoEndBkup: specifies that datafile in the database are taken out of the backup mode when instance is onlined.

See the *VERITAS Cluster Server Enterprise Agent for Oracle Installation and Configuration Guide* for a complete description of the EnvFile, Encoding, and AutoEndBkup attributes.



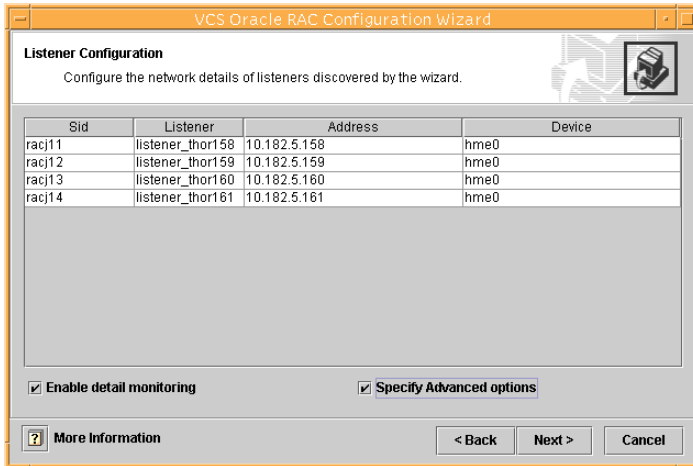
## Database Configuration Screen

If you have installed the database on a cluster file system, the wizard discovers the mount point and displays it on the Database Configuration screen. You can confirm the mount options displayed, or you can modify them. If the database exists on raw volumes, the wizard discovers the volumes.



## Listener Configuration Screen

The Listener Configuration screen displays the name of the listener corresponding to each database instance, as well as the IP address and device name used by each listener. Typically, you cannot change this information, only verify it.



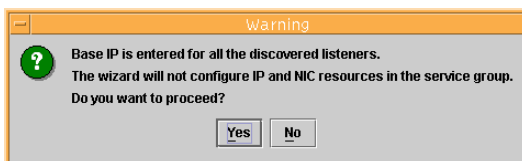
### Check boxes: Detail Monitoring, Advanced Listener Options

You can choose to configure detail monitoring for the Netlsnr agent by clicking the “Enable detail monitoring” check box. The wizard uses the monitor script `/opt/VRTSvcs/bin/Netlsnr/LsnrTest.pl` to monitor the listeners in detail.

You can also choose to “Specify Advanced options,” which include setting up an EnvFile (to define environment variables), Encoding, and LsnrPwd parameters. See “[Listener Advanced Configuration Screen](#)” on page 115 that follows.

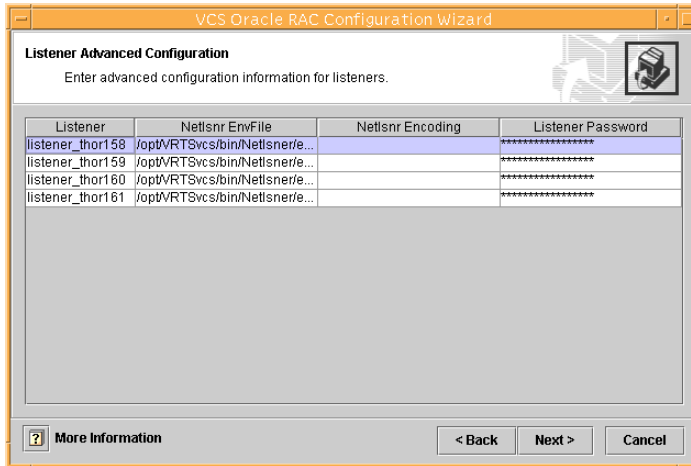
### Using the Host IP Address

If you have set up the listener to use the base, or host, IP address, the wizard displays the following message when you press Next on the Listener Configuration screen:



## Listener Advanced Configuration Screen

This screen displays if you have checked “Specify Advanced options at the bottom of the Listener Configuration Screen.



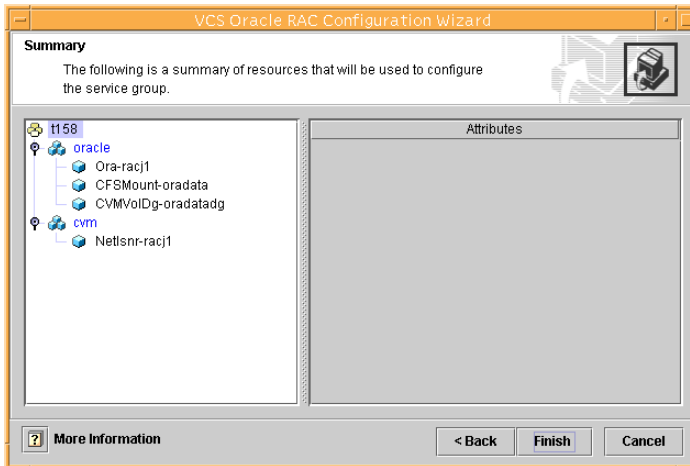
For each listener identified by name, this screen displays fields for defining the advanced attributes of the Netlsnr resource.

- ◆ The Netlsnr EnvFile: the name of the source file used by the agent entry point scripts; this file must exist
- ◆ Netlsnr Encoding: the operating system encoding that corresponds to Oracle encoding for the displayed Oracle output; the encoding value must match the encoding value used by the Oracle configuration (see “[Oracle Advanced Configuration Screen](#)” on page 112).
- ◆ Listener Password: the password used for Netlsnr; must specify the password as it appears in the listener.ora file.

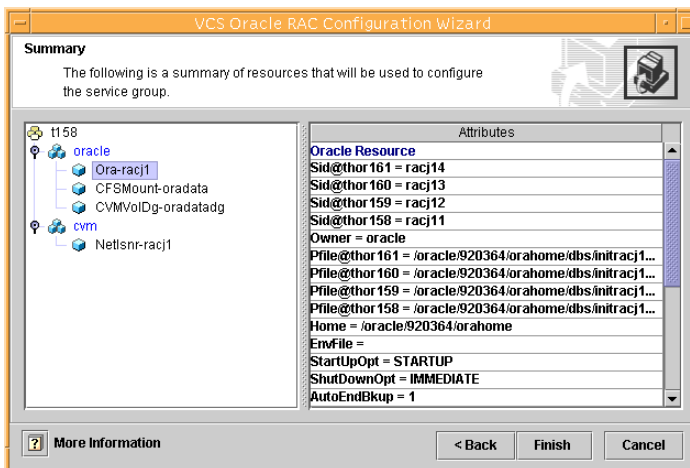


## Service Group Summary Screens

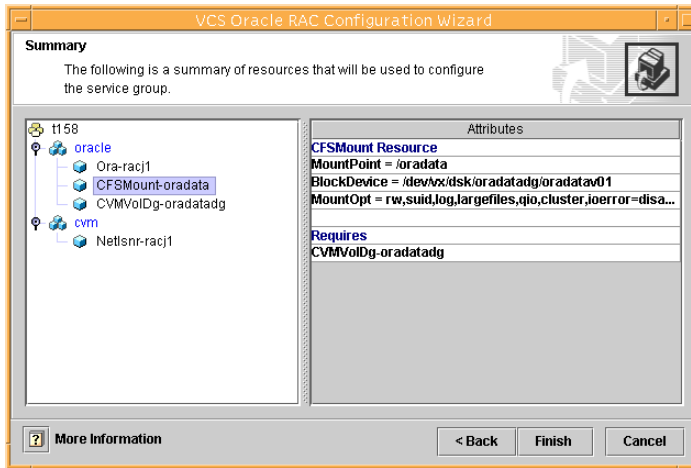
After you have configured the database and listener resources, the wizard displays the configuration on a “Summary” screen.



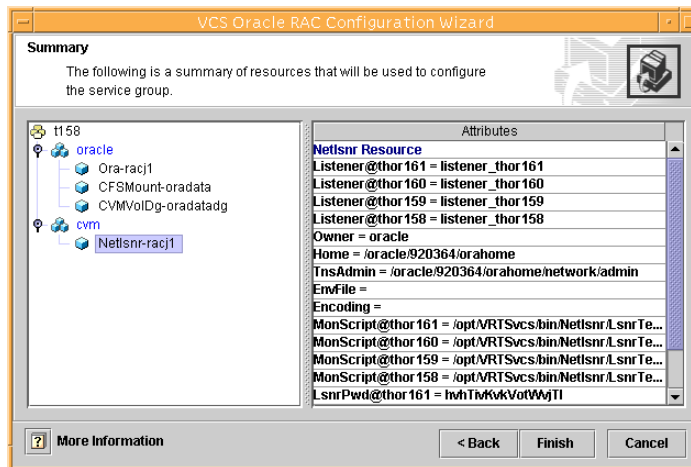
You can click on a resource within the service group to highlight it and display its attributes and their values. For example, if you click on the name of the Oracle resource, `Ora-racj1`, the wizard displays details of the Oracle resource, as the following illustration shows.



The next illustration shows the attributes for the CFSSMount resource. Note the dependencies listed at the bottom of the Attributes screen.



The NetLsnr resource is configured as part of the CVM service group. The CVM service group also contains other resources, but the wizard does not show them because the wizard does not control them.



### Implementing the Configuration

The wizard implements the configuration changes when you click Finish. The wizard creates the Oracle service group, adds the NetLsnr resource to the CVM the configuration.



## Modifying Oracle RAC Service Groups Using the Wizard

Once an Oracle RAC service group is created on a system, the configuration wizard can be used to modify the service group's Oracle, Netlsnr, and CVM components.

---

**Note** If modification of underlying mount point or volume information is necessary, the mount points or volumes must be deleted and added in the Oracle database before the wizard is started. Then, the wizard discovers the new information.

---

### Prerequisites

- ✓ To modify some resources, you must make the changes in the Oracle database before you start the RAC configuration wizard. When you start the wizard, it discovers the the new information. This applies to:
  - ◆ Adding or removing the database mount points
  - ◆ Adding or removing shared volumes
- ✓ To modify network resources, make sure that the service group is offline.
- ✓ To add or remove database files from your configuration, make sure that the service group is online.

### Modifying the Oracle 9i RAC Service Group Configuration

1. Start the RAC configuration wizard as root on the VCS system:  

```
# /opt/VRTSvcs/bin/hawizard rac
```
2. On the Welcome window, click Next.
3. In the Wizard Options window, select the "Modify RAC service group" option, select the service group to be modified, and click Next.
4. Follow the wizard instructions and make modifications as per your configuration. See ["Configuring the Service Groups Using the Wizard"](#) on page 105 for more information about the configuration wizard.





This chapter discusses I/O fencing with respect to various event scenarios.

## I/O Fencing of Shared Storage

When multiple systems have access to the data on shared storage, the integrity of the data depends on the systems communicating with each other so that each is aware when the other is writing data. Usually this communication occurs in the form of heartbeats through the private networks between the systems. If the private links are lost, or even if one of the systems is hung or too busy to send or receive heartbeats, each system could be unaware of the other's activities with respect to writing data. This is a *split brain* condition and can lead to data corruption.

The I/O fencing capability of the Storage Foundation *for Oracle RAC*, which is managed by VERITAS Volume Manager, prevents data corruption in the event of a split brain condition by using SCSI-3 persistent reservations for disks. This allows a set of systems to have registrations with the disk and a write-exclusive registrants-only reservation with the disk containing the data. This means that only these systems can read and write to the disk, while any other system can only read the disk. The I/O fencing feature fences out a system that no longer sends heartbeats to the other system by preventing it from writing data to the disk.

VxVM manages all shared storage subject to I/O fencing. It assigns the keys that systems use for registrations and reservations for the disks—including all paths—in the specified disk groups. The `vxfsen` driver is aware of which systems have registrations and reservations with specific disks. To protect the data on shared disks, each system in the cluster *must* be configured to use I/O fencing.



## Verifying Data Storage Arrays Using the vxfcntlshdw Utility

You can use the `vxfcntlshdw` utility to verify that shared storage arrays to be used for data support SCSI-3 persistent reservations and I/O fencing. The description in Chapter 3 shows how to use the testing utility to test a single disk. The utility has other options that may be more suitable for testing storage devices in other configurations.

---

**Note** Disks used as coordinator disks must also be tested. See [“Setting Up Coordinator Disks”](#) on page 54.

---

The utility, which you can run from one system in the cluster, tests the storage used for data by setting and verifying SCSI-3 registrations on the disk or disks you specify, setting and verifying persistent reservations on the disks, writing data to the disks and reading it, and removing the registrations from the disks. Refer also to the `vxfcntlshdw(1M)` manual page.

### General Guidelines for Using vxfcntlshdw

- ◆ The utility requires two systems connected to the shared storage.

---

**Caution** The tests overwrite and destroy data on the disks, unless you use the `-r` option.

---

- ◆ The two systems must have `remsh` permission set so that each node has root user access to the other. Temporarily modify the `/.rhosts` file to enable cluster communications for the `vxfcntlshdw` utility, placing a `++` character in the first line of the file. You can also limit the remote access to specific systems. Refer to the manual page for the `/.rhosts` file for more information. See [“Removing remsh Permissions and Restoring Public Network Connections”](#) on page 58 when you complete testing.
- ◆ To ensure both systems are connected to the same disk during the testing, you can use the `vxfenadm -i diskpath` command to verify a disk’s serial number. See [“Verifying the Nodes See the Same Disk”](#) on page 49.
- ◆ For disk arrays with many disks, use the `-m` option to sample a few disks before creating a disk group and using the `-g` option to test them all.
- ◆ When testing many disks with the `-f` or `-g` option, you can review results by redirecting the command output to a file.
- ◆ The utility indicates a disk can be used for I/O fencing with a message resembling:

```
The disk /dev/rdisk/c4t8d0s2 is ready to be configured for I/O
Fencing on node nebula
```

If the utility does not show a message stating a disk is ready, verification has failed.

- ◆ If the disk you intend to test has existing SCSI-3 registration keys, the test issues a warning before proceeding.

## vxfcntlshdw Options

The table below describes three methods the utility provides to test storage devices.

<b>vxfcntlshdw option</b>	<b>Description</b>	<b>When to Use</b>
<code>-m</code>	Utility runs manually, in interactive mode, prompting for systems and devices, and reporting success or failure. May be used with <code>-r</code> and <code>-t</code> options. <code>-m</code> is the default option.	For testing a few disks or for sampling disks in larger arrays.
<code>-f filename</code>	Utility tests system/device combinations listed in a text file. May be used with <code>-r</code> and <code>-t</code> options.	For testing several disks.
<code>-g disk_group</code>	Utility tests all disk devices in a specified disk group. May be used with <code>-r</code> and <code>-t</code> options.	For testing many disks and arrays of disks. Disk groups may be temporarily created for testing purposes and destroyed (ungrouped) after testing.

### Using the `-r` Option for Non-destructive Testing

To test disk devices containing data you want to preserve, you can use the `-r` option with the `-m`, `-f`, or `-g` options, which are described in the following sections. For example, to use the `-m` option and the `-r` option, you can run the utility by entering:

```
# /opt/VRTSvcs/vxfen/bin/vxfcntlshdw -rm
```

When invoked with the `-r` option, the utility does not use tests that write to the disks. Therefore, it does not test the disks for all of the usual conditions of use.

### Using the `-m` Option

The `-m` option for vxfcntlshdw is described in detail in chapter 3.



## Using the -f Option

Use the `-f` option to test disks that are listed in a text file. For example, you can create a file to test two disks shared by systems `galaxy` and `nebula` that might resemble:

```
galaxy /dev/rdisk/c2t2d1s2 nebula /dev/rdisk/c3t2d1s2
galaxy /dev/rdisk/c2t2d2s2 nebula /dev/rdisk/c3t2d2s2
```

where the first disk is listed in the first line and is seen by `galaxy` as `/dev/rdisk/c2t2d1s2` and by `nebula` as `/dev/rdisk/c3t2d1s2`. The other disk, in the second line, is seen as `/dev/rdisk/c2t2d2s2` from `galaxy` and `/dev/rdisk/c3t2d2s2` from `nebula`. Typically, the list of disks could be extensive.

Suppose you created the file named `disks_blue`. To test the disks, you would enter:

```
# /opt/VRTSvcs/vxfen/bin/vxfentshdw -f disks_blue
```

The utility reports the test results one disk at a time, just as for the `-m` option.

You can redirect the test results to a text file. Precede the command with “`yes`” to acknowledge that the testing destroys any data on the disks to be tested.

---

**Caution** Be advised that by redirecting the command’s output to a file, a warning that the testing destroys data on the disks cannot be seen until the testing is done.

---

For example:

```
# yes | /opt/VRTSvcs/vxfen/bin/vxfentshdw -f disks_blue >
blue_test.txt
```

## Using the -g Option

Use the `-g` option to test all disks within a disk group. For example, you create a temporary disk group consisting of all disks in a disk array and test the group.

---

**Note** Do not import the test disk group as shared; that is, do not use the `-s` option.

---

The utility reports the test results one disk at a time. You can redirect the test results to a text file for review.

```
# /opt/VRTSvcs/vxfen/bin/vxfentshdw -g red_disks_dg > redtest.txt
```

After testing, destroy the disk group and put the disks into disk groups as you need.

## Testing a Disk with Existing Keys

If the utility detects that a coordinator disk has existing keys, you see a message that resembles:

```
There are VERITAS I/O Fencing keys on the disk. Please make sure
that I/O Fencing is shut down on all nodes of the cluster before
continuing.
```

```
***** WARNING!!!!!!!!!! *****
```

```
THIS SCRIPT CAN ONLY BE USED IF THERE ARE NO OTHER ACTIVE NODES IN
THE CLUSTER! VERIFY ALL OTHER NODES ARE POWERED OFF OR INCAPABLE
OF ACCESSING SHARED STORAGE.
```

If this is not the case, data corruption will result.

Do you still want to continue : [y/n] (default: n) **y**

The utility prompts you with a warning before proceeding. You may continue as long as I/O fencing is not yet configured.



## How I/O Fencing Works in Different Event Scenarios

The following table describes how I/O fencing works to prevent data corruption in different failure event scenarios. For each event, corrective operator actions are indicated.

<b>Event</b>	<b>Node A: What Happens?</b>	<b>Node B: What Happens?</b>	<b>Operator Action</b>
Both private networks fail.	Node A races for majority of coordinator disks. If Node A wins race for coordinator disks, Node A ejects Node B from the shared disks and continues.	Node B races for majority of coordinator disks. If Node B loses the race for the coordinator disks, Node B removes itself from the cluster.	When Node B is ejected from cluster, repair the private networks before attempting to bring Node B back.
Both private networks function again after event above.	Node A continues to work.	Node B has crashed. It cannot start the database since it is unable to write to the data disks.	Restart Node B after private networks are restored.
One private network fails.	Node A prints message about an IOFENCE on the console but continues.	Node B prints message about an IOFENCE on the console but continues.	Repair private network. After network is repaired, both nodes automatically use it.
Node A hangs.	Node A is extremely busy for some reason or is in the kernel debugger.  When Node A is no longer hung or in the kernel debugger, any queued writes to the data disks fail because Node A is ejected. When Node A receives message from GAB about being ejected, it removes itself from the cluster.	Node B loses heartbeats with Node A, and races for a majority of coordinator disks.  Node B wins race for coordinator disks and ejects Node A from shared data disks.	Verify private networks function and restart Node A.



Event	Node A: What Happens?	Node B: What Happens?	Operator Action
<p>Nodes A and B and private networks lose power. Coordinator and data disks retain power. Power returns to nodes and they restart, but private networks still have no power.</p>	<p>Node A restarts and I/O fencing driver (vxfen) detects Node B is registered with coordinator disks. The driver does not see Node B listed as member of cluster because private networks are down. This causes the I/O fencing device driver to prevent Node A from joining the cluster. Node A console displays:</p> <p style="padding-left: 40px;">Potentially a preexisting split brain. Dropping out of the cluster. Refer to the user documentation for steps required to clear preexisting split brain.</p>	<p>Node B restarts and I/O fencing driver (vxfen) detects Node A is registered with coordinator disks. The driver does not see Node A listed as member of cluster because private networks are down. This causes the I/O fencing device driver to prevent Node B from joining the cluster. Node B console displays:</p> <p style="padding-left: 40px;">Potentially a preexisting split brain. Dropping out of the cluster. Refer to the user documentation for steps required to clear preexisting split brain.</p>	<p>Refer to section in Troubleshooting chapter for instructions on resolving preexisting split brain condition.</p>



Event	Node A: What Happens?	Node B: What Happens?	Operator Action
<p>Node A crashes while Node B is down. Node B comes up and Node A is still down.</p>	<p>Node A is crashed.</p>	<p>Node B restarts and detects Node A is registered with the coordinator disks. The driver does not see Node A listed as member of the cluster. The I/O fencing device driver prints message on console:</p> <pre>Potentially a preexisting split brain. Dropping out of the cluster. Refer to the user documentation for steps required to clear preexisting split brain.</pre>	<p>Refer to section in Troubleshooting chapter for instructions on resolving preexisting split brain condition.</p>





Event	Node A: What Happens?	Node B: What Happens?	Operator Action
The disk array containing two of the three coordinator disks is powered off.	Node A continues to operate as long as no nodes leave the cluster.	Node B continues to operate as long as no nodes leave the cluster.	
Node B leaves the cluster and the disk array is still powered off.	Node A races for a majority of coordinator disks. Node A fails because only one of three coordinator disks is available. Node A removes itself from the cluster.	Node B leaves the cluster.	Power on failed disk array and restart I/O fencing driver to enable Node A to register with all coordinator disks.

## The vxfenadm Utility

Administrators can use the `vxfenadm` command to troubleshoot and test fencing configurations. The command's options for use by administrators are:

- g - read and display keys
- i - read SCSI inquiry information from device
- m - register with disks
- n - make a reservation with disks
- p - remove registrations made by other systems
- r - read reservations
- x - remove registrations



## Registration Key Formatting

The key defined by VxVM associated with a disk group consists of seven bytes maximum. This key becomes unique among the systems when the VxVM prefixes it with the ID of the system. The key used for I/O fencing, therefore, consists of eight bytes.

0							7
Node ID	VxVM Defined	VxVM Defined	VxVM Defined	VxVM Defined	VxVM Defined	VxVM Defined	VxVM Defined

The keys currently assigned to disks can be displayed by using the `vxfenadm` command. For example, from the system with node ID 1, display the key for the disk `/dev/rdisk/c2t1d0s2` by entering:

```
# vxfenadm -g /dev/rdisk/c2t1d0s2
Reading SCSI Registration Keys...
Device Name: /dev/rdisk/c2t1d0s2
Total Number of Keys: 1
key[0]:
    Key Value [Numeric Format]: 65,80,71,82,48,48,48,48
    Key Value [Character Format]: APGR0000
```

The `-g` option of `vxfenadm` displays all eight bytes of a key value in two formats. In the numeric format, the first byte, representing the Node ID, contains the system ID plus 65. The remaining bytes contain the ASCII values of the letters of the key, in this case, "PGR0000." In the next line, the node ID 0 is expressed as "A;" node ID 1 would be "B."

# Using Storage Checkpoints and Storage Rollback

---

7

VERITAS Storage Checkpoints enable efficient backup and recovery of Oracle databases. This chapter describes Storage Checkpoints and Storage Rollback and how to use these technologies through Storage Foundation *for Oracle RAC*.

Topics covered in this chapter include:

- ◆ [“Using Storage Checkpoints and Storage Rollback for Backup and Restore”](#) on page 129
- ◆ [“Determining Space Requirements for Storage Checkpoints”](#) on page 131
- ◆ [“Performance of Storage Checkpoints”](#) on page 132
- ◆ [“Backing Up and Recovering the Database Using Storage Checkpoints”](#) on page 135
- ◆ [“Guidelines for Oracle Recovery”](#) on page 139
- ◆ [“Using the Storage Checkpoint Command Line Interface \(CLI\)”](#) on page 141

## Using Storage Checkpoints and Storage Rollback for Backup and Restore

VERITAS Storage Foundation *for Oracle RAC* provides a Storage Checkpoint facility that is similar to the snapshot file system mechanism. However, a Storage Checkpoint persists after a system restart. A Storage Checkpoint creates an exact image of a database instantly and provides a consistent image of the database from the time the Storage Checkpoint is created. The Storage Checkpoint image is managed and available only through the Storage Foundation *for Oracle RAC* command line interface (CLI). VERITAS NetBackup also makes use of Storage Checkpoints to provide a very efficient Oracle backup mechanism.

---

**Note** For more information on creating Storage Checkpoints with the CLI, see [“Creating Storage Checkpoints Using sfrac\\_ckptcreate”](#) on page 144.

---



A direct application of the Storage Checkpoint facility is Storage Rollback. Because each Storage Checkpoint is a consistent, point-in-time image of a file system, Storage Rollback is the restore facility for these on-disk backups. Storage rollback rolls back blocks contained in a Storage Checkpoint into the primary file system for faster database recovery. For more information on Storage Checkpoints and Storage Rollback, see the *VERITAS File System Administrator's Guide*.

## Understanding Storage Checkpoints and Storage Rollback

A Storage Checkpoint is a disk and I/O efficient snapshot technology for creating a “clone” of a currently mounted file system (the *primary* file system). Like a snapshot file system, a Storage Checkpoint appears as an exact image of the snapped file system at the time the Storage Checkpoint is made. However, unlike a snapshot file system that uses separate disk space, all Storage Checkpoints share the same free space pool where the primary file system resides. A Storage Checkpoint can be mounted as read-only or read-write, allowing access to the files as if it were a regular file system.

Initially, a Storage Checkpoint contains no data—it contains only the inode list and the block map of the primary fileset. This block map points to the actual data on the primary file system. Because only the inode list and block map are needed and no data is copied, creating a Storage Checkpoint takes only a few seconds and very little space.

A Storage Checkpoint initially satisfies read requests by finding the data on the primary file system, using its block map copy, and returning the data to the requesting process. When a write operation changes a data block *n* in the primary file system, the old data is first copied to the Storage Checkpoint, and then the primary file system is updated with the new data. The Storage Checkpoint maintains the exact view of the primary file system at the time the Storage Checkpoint had been taken. Subsequent writes to block *n* on the primary file system do not result in additional copies to the Storage Checkpoint because the old data only needs to be saved once. As data blocks are changed on the primary file system, the Storage Checkpoint gradually fills with the original data copied from the primary file system. Less of the block map in the Storage Checkpoint points back to blocks on the primary file system.

Storage rollback restores a database, a tablespace, or datafiles in the primary file systems to the point-in-time image created during a Storage Checkpoint. Storage rollback is accomplished by copying the “before” images from the appropriate Storage Checkpoint back to the primary file system. As with Storage Checkpoints, Storage Rollback restores at the block level, rather than at the file level.

If you mount a Storage Checkpoint as read-write, the CLI utility mounts the Storage Checkpoint as read-only. It then creates a shadow Storage Checkpoint of the mounted read-only checkpoint and mounts the shadow Storage Checkpoint instead. This ensures that any Storage Checkpoint data that has been modified incorrectly cannot be a source of any database corruption.



Mountable Storage Checkpoints can be used for a wide range of application solutions, including backup, investigations into data integrity, staging upgrades or database modifications, and data replication solutions.

---

**Note** For more information on mountable Storage Checkpoints, see “[Mounting Storage Checkpoints Using sfrac\\_ckptmount](#)” on page 148.

---

## Determining Space Requirements for Storage Checkpoints

To support Block-level Incremental (BLI) Backup and Storage Rollback, the file systems need extra disk space to store the Storage Checkpoints. The extra space needed depends on how the Storage Checkpoints are used. Storage Checkpoints that are used to keep track of the block changes contain only file system block maps, and therefore require very little additional space (less than 1 percent of the file system size).

If the database is online while the backup is running, the additional space required by each file system for Storage Checkpoints depends on the duration of the backup and the database workload. If workload is light during the backup or the backup window is relatively short (for example, for incremental backups), for most database configurations, an additional 10 percent of the file system size will be sufficient. If the database has a busy workload while a full backup is running, the file systems may require more space.

To support Storage Checkpoints and Storage Rollback, VxFS needs to keep track of the original block contents when the Storage Checkpoints were created. The additional space needed is proportional to the number of blocks that have been changed since a Storage Checkpoint was taken. The number of blocks changed may not be identical to the number of changes. For example, if a data block has been changed many times, only the first change requires a new block to be allocated to store the original block content. Subsequent changes to the same block require no overhead or block allocation.

If a file system that has Storage Checkpoints runs out of space, by default VxFS removes the oldest Storage Checkpoint automatically instead of returning an `ENOSPC` error code (UNIX `errno 28- No space left on device`), which can cause the Oracle instance to fail. Removing Storage Checkpoints automatically ensures the expected I/O semantics, but at the same time, eliminates a key recovery mechanism.

When restoring a file system that has data-full Storage Checkpoints from tape or other offline media, you need extra free space on the file system. The extra space is needed to accommodate the copy-on-write algorithm needed for preserving the consistent image of the Storage Checkpoints. The amount of free space required depends on the size of the restore and the number of Storage Checkpoints on the file system.

If you are restoring the entire file system, in most cases, you no longer need the existing Storage Checkpoint. You can simply re-make the file system using the `mkfs` command, and then restore the file system from tape or other offline media.



If you are restoring some of the files in the file system, you should first remove the data-full Storage Checkpoints that are no longer needed. If you have very limited free space on the file system, you may have to remove all data-full Storage Checkpoints in order for the restore to succeed.

Always reserve free disk space for growing volumes and file systems. You can also preallocate sufficient space for each file system when the file system is first created or manually grow the file system and logical volume where the file system resides. See the `vxassist(1)` and `fsadm_vxfs(1)` manual pages for more information.

## Performance of Storage Checkpoints

VxFS attempts to optimize the read and write access performance on both the Storage Checkpoint and the primary file system. Reads from a Storage Checkpoint typically perform at nearly the throughput of reads from a normal VxFS file system, allowing backups to proceed at the full speed of the VxFS file system.

Writes to the primary file system are typically affected by the Storage Checkpoints because the initial write to a data block requires a read of the old data, a write of the data to the Storage Checkpoint, and finally, the write of the new data to the primary file system. Having multiple Storage Checkpoints on the same file system, however, will not make writes slower. Only the initial write to a block suffers this penalty, allowing operations like writes to the intent log or inode updates to proceed at normal speed after the initial write.

Reads from the Storage Checkpoint are impacted if the primary file system is busy, because the reads on the Storage Checkpoint are slowed by all of the disk I/O associated with the primary file system. Therefore, performing database backup when the database is less active is recommended.

For more information on Storage Checkpoints and Storage Rollback, see the *VERITAS File System Administrator's Guide*.

## Setting Up the SFRAC Storage Checkpoint Feature for Use

To enable use of the Storage Foundation *for Oracle RAC* Storage Checkpoint feature:

- ✓ Set the ownership and permissions for the Storage Checkpoint utility.
- ✓ Choose only one instance from the database for which you want to create and manage a Storage Checkpoint.
- ✓ From the appropriate system, create a repository directory. This directory may be created on a cluster file system, on private shared storage, or within a local file system. The procedure that follows describes creating the repository on a cluster file system. A repository is required for storing information about the database elements.
- ✓ Set the ownership and permissions on the repository directory.
- ✓ Create a locks directory in the repository directory.
- ✓ Create the repository.
- ✓ Create the Storage Checkpoint for the database.

Perform the following steps on the node where the instance is running. In these examples, the owner is `oracle` and the group is `dba`.

### ▼ To enable Oracle user and dba group access to VRTSdbckp directory

1. Log in as root user and move to the `/opt` directory:

```
# cd /opt
```

2. Set the ownership and permissions of the `VRTSdbckp` directory:

```
# chown oracle /opt/VRTSdbckp
# chmod 500 /opt/VRTSdbckp
# chgrp dba /opt/VRTSdbckp
# chmod 550 /opt/VRTSdbckp
```

3. Perform the preceding steps on all nodes in the cluster.

### ▼ To choose a database instance and create a repository directory

1. Log in to the system running an instance of the database for which you want to create and manage a Storage Checkpoint.



2. Create and mount a file system for the database repository. In this example, the shared file system `ckptrepvol1` with a size of 300 Megabytes is created and mounted at the mount point `ckptrep`. The repository directory is `/ckptrep/test`.

```
# vxldg -s init ckptrepdg ckptrepdgd1=c2t3d0
# vxassist -g ckptrepdg make ckptrepvol1 300m
# mkfs -F vxfs /dev/vx/rdisk/ckptrepdg/ckptrepvol1
# mount -F vxfs -o cluster /dev/vx/dsk/ckptrepdg/ckptrepvol1 \
/ckptrep
# mkdir /ckptrep/test
```

#### ▼ To set ownership of repository directory

On the system where the instance is running, set the ownership of the repository directory:

```
# chown oracle:dba /ckptrep/test
```

#### ▼ To set the permissions of the repository directory and create a locks directory

1. Log in as `oracle` user.
2. As user `oracle`, set the permissions for the repository directory:

```
$ chmod 777 /ckptrep/test
```

3. Create a `locks` directory in the repository:

```
$ mkdir /ckptrep/test/locks
```

#### ▼ To create Storage Checkpoints

1. Making sure that the instance is running, set the `PATH` and `VXDBA_DBPATH` environment variables:

```
$ export PATH=$PATH:/opt/VRTSdbckp/bin
$ export VXDBA_DBPATH=/ckptrep/test
```

2. Run the `sfrac_update` command to create the repository:

```
$ sfrac_update -S db12 -H /oracle/92064/orahome
```

See [“Creating or Updating The Repository Using `sfrac\_update`”](#) on page 143 for more information about the `sfrac_update` command.

3. Create the checkpoint:

```
$ sfrac_ckptcreate -S db12 -H /oracle/92064/orahome -o online
```

See also [“Creating Storage Checkpoints Using `sfrac\_ckptcreate`”](#) on page 144.





## Backing Up and Recovering the Database Using Storage Checkpoints

Storage Checkpoints can be created by specifying one of the following options: online, offline, or instant. To create a Storage Checkpoint with the online option, the database should be online and you must enable ARCHIVELOG mode for the database. During the creation of the Storage Checkpoint, the tablespaces are placed in backup mode. Because it only takes a few seconds to take a Storage Checkpoint, the extra redo logs generated while the tablespaces are in online-backup mode are very small. You can roll back the entire database or individual tablespaces or datafiles to an online or offline Storage Checkpoint. After the rollback is complete, rolling the database forward is supported with online Storage Checkpoints. For the instant option, the database should be online and it can be running in either ARCHIVELOG or NOARCHIVELOG mode. You can only roll back the entire database to an instant Storage Checkpoint. After the rollback is complete, you need to perform Oracle database storage rollback instance recovery. Rolling the database forward is not supported; that is, you cannot apply archived redo logs.

---

**Note** For best recoverability, always keep ARCHIVELOG mode enabled, regardless of whether the database is online or offline when you create Storage Checkpoints.

---

Since the Storage Checkpoints record the before images of blocks that have changed, you can use them to do a disk-based or file-system-based storage rollback to the exact time when the Storage Checkpoint was taken. You can consider Storage Checkpoints as backups that are online, and you can use them to roll back an entire database tablespace, or a single database file. Rolling back to or restoring from any Storage Checkpoint is generally very fast because only the changed data blocks need to be restored.

---

**Note** Some database changes made after a Storage Checkpoint was taken may make it impossible to recover the database after storage rollback. For example, you may not be able to recover the database after a successful rollback if the control files for the database have recorded the addition or removal of datafiles. To provide recovery options, a backup copy of the control file for the database is saved under the `$VXDBA_DBPATH/$ORACLE_SID/checkpoint_dir/CKPT_NAME` directory immediately after a Storage Checkpoint is created. You can use this file to assist with database recovery, if necessary. If possible, both ASCII and binary versions of the control file will be left under the `$VXDBA_DBPATH/$ORACLE_SID/checkpoint_dir/CKPT_NAME` directory. The binary version will be compressed to conserve space. Use extreme caution when recovering your database using alternate control files.

---

Storage Checkpoints can only be used to restore from logical errors (for example, a human error). Because all the data blocks are on the same physical device, Storage Checkpoints cannot be used to restore files due to a media failure. A media failure requires a database restore from a tape backup or a copy of the database files kept on a separate medium. The



combination of data redundancy (disk mirroring) and Storage Checkpoints is recommended for highly critical data to protect them from both physical media failure and logical errors.

## Backing Up Using the Command Line

You can back up a database by creating a Storage Checkpoint using the `sfrac_ckptcreate` command, mounting the Storage Checkpoint as read-only using the `sfrac_ckptmount` command, and then backing it up using tools such as `tar` or `cpio`.

### Usage Notes

- ◆ See the `sfrac_ckptcreate(1M)`, `sfrac_ckptmount(1M)`, `tar(1)`, and `cpio(1)` manual pages for more information.
- ◆ Also refer to the examples, “[Creating Storage Checkpoints Using `sfrac\_ckptcreate`](#)” on page 144 and “[Mounting Storage Checkpoints Using `sfrac\_ckptmount`](#)” on page 148.

### ▼ To back up a frozen database image using the command line

---

**Note** In this example, all the database datafiles reside on one VxFS file system named `/db01`.

---

1. Create a Storage Checkpoint using the `sfrac_ckptcreate` command:

```
$ /opt/VRTS/bin/sfrac_ckptcreate -S DB12 -H $ORACLE_HOME \-o
online
NOTICE: Creating online Storage Checkpoint of database DB12.
Storage Checkpoint Checkpoint_1068762379 created.
```

2. Mount the Storage Checkpoint using the `sfrac_ckptmount` command:

```
$ /opt/VRTS/bin/sfrac_ckptmount-S DB12 -c \
Checkpoint_1068762379 -m /ckptmnt
```

3. Use `tar` to back up the Storage Checkpoint:

```
$ cd /ckptmnt
$ ls
db01
$ tar cvf /tmp/DB12_db1_1068762379.tar ./db01
```



## Verifying a Storage Checkpoint Using the Command Line

After creating a Storage Checkpoint, you may verify that it is free of errors using the procedure below.

### Usage Notes

See the `sfrac_ckptcreate(1M)` and `sfrac_ckptmount(1M)` manual pages for more information. Also refer to the examples, “[Creating Storage Checkpoints Using `sfrac\_ckptcreate`](#)” on page 144 and “[Mounting Storage Checkpoints Using `sfrac\_ckptmount`](#)” on page 148.

#### ▼ To verify that a Storage Checkpoint is error-free using the command line

1. Create and mount a Storage Checkpoint:

```
$ sfrac_ckptcreate -S DB12 -H $ORACLE_HOME
Storage Checkpoint Checkpoint_903937870 created.
$ sfrac_ckptmount -S DB12 -c Checkpoint_903937870 \
-m /ckptmnt
```

2. Examine the content of the Storage Checkpoint:

```
$ ls -l /ckptmnt/db01/DB12
-rw-r----- 1 oracle dba 12247040 Nov 13 15:28 control01ctl
-rw-r----- 1 oracle dba 12247040 Nov 13 15:28 control02ctl
-rw-r----- 1 oracle dba 12247040 Nov 13 15:28 control03ctl
.
.
-rw-r----- 1 oracle dba 41951232 Oct 29 17:30 temp01.dbf
-rw-r----- 1 oracle dba 10493952 Nov 13 15:28 tools01.dbf
-rw-r----- 1 oracle dba 209723392 Nov 13 15:28 undotbs01.dbf
-rw-r----- 1 oracle dba 209723392 Nov 13 15:28 undotbs02.dbf
-rw-r----- 1 oracle dba 26222592 Nov 13 15:28 users01.dbf
```

3. Run the `dbv` tool against the datafile (for example, `undotbs01.dbf`):

```
$ dbv file=/ckptmnt/db01/db12/undotbs01.dbf blocksize=8192
DBVERIFY: Release 9.2.0.4.1 - Production on Thu Nov 13 15:46:10
2003

Copyright (c) 1982, 2002, Oracle Corporation. All rights
reserved.

DBVERIFY - Verification starting : FILE =
/ckptmnt/db1/db1/undotbs01.dbf
```



DBVERIFY - Verification complete

Total Pages Examined : 25600  
Total Pages Processed (Data) : 0  
Total Pages Failing (Data) : 0  
Total Pages Processed (Index): 0  
Total Pages Failing (Index): 0  
Total Pages Processed (Other): 8880  
Total Pages Processed (Seg) : 0  
Total Pages Failing (Seg) : 0  
Total Pages Empty : 16720  
Total Pages Marked Corrupt : 0  
Total Pages Influx : 0



## Guidelines for Oracle Recovery

For optimal Oracle recovery, follow these guidelines:

- ◆ Back up all control files before storage rollback in case the subsequent Oracle recovery is not successful. Oracle recommends that you keep at least two copies of the control files for each Oracle database and that you store the copies on different disks. It is also a good idea to back up the control files before and after making structural changes to databases.

---

**Note** The `sfrac_ckptcreate` command automatically saves control file and log information when you create a Storage Checkpoint. See “[Creating Storage Checkpoints Using `sfrac\_ckptcreate`](#)” on page 144 for more information.

---

- ◆ Make sure that the control files are *not* rolled back.

A control file is a small binary file that describes the structure of the database and must be available to mount, open, and maintain the database. The control file stores all necessary database file information, log file information, the name of the database, the timestamp of database creation, and synchronization information, such as the Storage Checkpoint and log-sequence information needed for recovery. Rolling back the control file will result in an inconsistency between the physical database structure and the control file.

---

**Note** If your intention is to roll back the database to recover from structural changes that you do not want to maintain, you may want to use the backup control file that was created by the `sfrac_ckptcreate` command. The backup control file is located in the directory `$(VXDBA_DBPATH)/$(ORACLE_SID)/checkpoint_dir/CKPT_NAME`.

---

- ◆ Make sure that all archived redo logs are available.

A database backup with online and archived logs is required for a complete database recovery. Query `V$ARCHIVED_LOG` to list all the archived log information and `V$ARCHIVE_DEST` to list the location of archive destinations.

To restore the necessary archived redo log files, you can query `V$LOG_HISTORY` to list all the archived redo log history or query `V$RECOVERY_LOG` to list only the archived redo logs needed for recovery. The required archived redo log files can be restored to the destination specified in the `LOG_ARCHIVE_DEST` parameter or to an alternate location. If the archived redo logs were restored to an alternate location, use the `ALTER DATABASE RECOVER . . . FROM` statement during media recovery.

- ◆ After storage rollback, perform Oracle recovery, applying some or all of the archived redo logs.



**Note** After rolling back the database (including control files and redo logs) to an instant Storage Checkpoint, you need to recover the Oracle database instance. Rolling the database forward is not supported; that is, you cannot apply archived redo logs.

---

- ◆ To perform a complete media recovery:

```
SET AUTORECOVERY ON;  
RECOVER DATABASE;
```

- ◆ To perform an incomplete media recovery, use one of the following:

```
RECOVER DATABASE UNTIL CANCEL;
```

or

```
RECOVER DATABASE UNTIL TIME 'yyyy-mm-dd:hh:mm:ss';
```

(You can confirm the time of error by checking the `../bdump/alert*.log` file.)

or

```
RECOVER DATABASE UNTIL TIME 'yyyy-mm-dd:hh:mm:ss' using \  
backup controlfile;
```

or

```
RECOVER DATABASE UNTIL CHANGE scn;
```

- ◆ To open the database after an incomplete media recovery, use the following:

```
ALTER DATABASE OPEN RESETLOGS;
```

RESETLOGS resets the log sequence. The RESETLOGS option is required after an incomplete media recovery. After opening the database with the RESETLOGS option, remove the Storage Checkpoint you just rolled back to as well as any Storage Checkpoints that were taken before that one. These earlier Storage Checkpoints can no longer be used for storage rollback. After removing these Storage Checkpoints, be sure to create a new Storage Checkpoint.

---

**Caution** After rolling back, be sure to delete the Storage Checkpoint that you rolled back to and then create a new one.

---

See your Oracle documentation for complete information on recovery.



# Using the Storage Checkpoint Command Line Interface (CLI)

VERITAS Storage Foundation for Oracle RAC provides a command line interface to many key operations. The command line interface lets you incorporate command operations into scripts and other administrative processes.

## Overview of Commands

VERITAS Storage Foundation for Oracle RAC commands supported in the command line interface are located in the `/opt/VRTS/bin` directory. Online manual pages for these commands are located in the `/opt/VRTS/man` directory.

The following table summarizes the commands available to you from the command line:

VERITAS Storage Foundation for Oracle RAC Commands

Command	Description
<code>sfrac_update</code>	Creates or updates the VxDBA repository in VERITAS Storage Foundation for Oracle RAC
<code>sfrac_ckptcreate</code>	Creates a Storage Checkpoint for an Oracle database
<code>sfrac_ckptdisplay</code>	Displays Storage Checkpoints associated with an Oracle instance
<code>sfrac_ckptmount</code>	Mounts a Storage Checkpoint for an Oracle instance
<code>sfrac_ckptumount</code>	Unmounts a Storage Checkpoint for an Oracle instance
<code>sfrac_ckptrollback</code>	Rolls back an Oracle instance to a Storage Checkpoint point-in-time image
<code>sfrac_ckptremove</code>	Removes a Storage Checkpoint for an Oracle instance



## Examples of Using the Command Line Interface

This section provides examples for using the VERITAS Storage Foundation for Oracle RAC command line interface to perform administrative operations. For more detailed information about the commands and their syntax and available options, see the individual manual pages.

### Prerequisites

- ◆ You must log in as the database administrator to use the following CLI commands.
  - ◆ `sfrac_update`
  - ◆ `sfrac_ckptcreate`
  - ◆ `sfrac_ckptrollback`
- ◆ You can log in as the database administrator (typically, the user ID `oracle`) or `root` to use the following CLI commands.
  - ◆ `sfrac_ckptdisplay`
  - ◆ `sfrac_ckptmount`
  - ◆ `sfrac_ckptumount`
  - ◆ `sfrac_ckptremove`





## Creating or Updating The Repository Using `sfrac_update`

You can use the VERITAS Storage Foundation for Oracle RAC `sfrac_update` command to create or update the repository for VxDBA.

### Prerequisites

You must be logged on as the database administrator (typically, the user ID `oracle`).

### Usage Notes

- ◆ The `sfrac_update` command creates a repository in the `$VXDBA_DBPATH/ORACLE_SID` directory where information used by VERITAS Storage Foundation for Oracle RAC is kept. If the repository already exists, the command will refresh the information.
- ◆ The database must be up and running, and the `ORACLE_SID` and the `ORACLE_HOME` variable arguments must be specified with the `-S` and `-H` options, respectively.
- ◆ See the `sfrac_update(1M)` manual page for more information.

#### ▼ To update the VxDBA repository

Use the `sfrac_update` command as follows:

```
$ /opt/VRTS/bin/sfrac_update -S DB12 -H $ORACLE_HOME
```

#### ▼ To view the status of the VxDBA repository

Use the `sfrac_update` command with the `-n` option as follows:

```
$ /opt/VRTS/bin/sfrac_update -S DB12 -H $ORACLE_HOME -n
VxDBA repository is up to date.
```



## Creating Storage Checkpoints Using `sfrac_ckptcreate`

- ◆ You can use the VERITAS Storage Foundation *for Oracle RAC* `sfrac_ckptcreate` command to create a Storage Checkpoint from the command line. Storage Checkpoints can be either `online`, `offline`, or `instant`. See the `sfrac_ckptcreate(1M)` manual page for more information. If `online` is specified, the database is put into hot-backup mode when the Storage Checkpoint is created.
- ◆ If `offline` is specified, the database is expected to be down; this is the default.
- ◆ If `instant` is specified, the database is expected to be online; a Storage Checkpoint is taken for a “crash recovery”-type storage rollback.

### Usage Notes

In addition to creating the Storage Checkpoint, `sfrac_ckptcreate` also backs up the Oracle control file in:

```
$VXDBA_DBPATH/$ORACLE_SID/checkpoint_dir/CKPT_NAME
```

#### ▼ To create Storage Checkpoints while the database is online

Use the `sfrac_ckptcreate` command as in the following example:

```
$ /opt/VRTS/bin/sfrac_ckptcreate -S DB12 \  
-H $ORACLE_HOME -o online  
NOTICE: Creating online Storage Checkpoint of database DB12.  
Storage Checkpoint Checkpoint_1068762379 created.
```

#### ▼ To create Storage Checkpoints without updating the VxDBA repository while the database is online

Use the `sfrac_ckptcreate` command as in the following example:

```
$ /opt/VRTS/bin/sfrac_ckptcreate -S DB12 \  
-H $ORACLE_HOME -o online -n  
Storage Checkpoint Checkpoint_1068762379 created.
```

#### ▼ To create Storage Checkpoints while the database is offline

Use the `sfrac_ckptcreate` command as in the following example:

```
$ /opt/VRTS/bin/sfrac_ckptcreate -S DB12 \  
-H $ORACLE_HOME -o offline  
Storage Checkpoint Checkpoint_1068762379 created.
```

---

**Note** The default option is `offline`.

---

## Displaying Storage Checkpoints Using `sfrac_ckptdisplay`

You can use the VERITAS Storage Foundation for Oracle RAC `sfrac_ckptdisplay` command from the command line to display the Storage Checkpoints associated with a running Oracle database.

### Usage Notes

- ◆ See the `sfrac_ckptdisplay(1M)` manual page for more information.
- ◆ The default ordering for sorting Storage Checkpoint names is “-r” (most to least recent). By setting this variable to another sort option, the Status field identifies if the Storage Checkpoint is partial (P), complete (C), invalid (I), mounted (M), read-only (R), writable (W), or of type online (ON), offline (OFF), instant (IN), or unknown (UN).

#### ▼ To display VxDBA Storage Checkpoints without updating the VxDBA repository

Use the `sfrac_ckptdisplay` command as in the following example:

```
$ /opt/VRTS/bin/sfrac_ckptdisplay -S DB12 \
-H $ORACLE_HOME -o dbed -n
Checkpoint_971672042          Sun Oct 15 13:55:53 2000  C+R+IN
Checkpoint_903937870        Fri Oct 13 22:51:10 2000  C+R+ON
Checkpoint_901426272        Wed Oct 11 16:17:52 2000  P+R+ON
```

#### ▼ To display VxDBA Storage Checkpoints

Use the `sfrac_ckptdisplay` command as in the following example:

```
$ /opt/VRTS/bin/sfrac_ckptdisplay -S DB12 \
-H $ORACLE_HOME -o dbed
Checkpoint_971672042          Sun Oct 15 13:55:53 2000  C+R+IN
Checkpoint_903937870        Fri Oct 13 22:51:10 2000  C+R+ON
Checkpoint_901426272        Wed Oct 11 16:17:52 2000  P+R+ON
```



## Scheduling Storage Checkpoints Using `sfrac_ckptcreate` and `cron`

You can use the VERITAS Storage Foundation for Oracle RAC `sfrac_ckptcreate` command to schedule Storage Checkpoint creation in a `cron` job or other administrative script.

### Usage Notes

- ◆ Create a new `crontab` file or edit an existing `crontab` file to include a Storage Checkpoint creation entry with the following space-delimited fields:

```
minute hour day_of_month month_of_year day_of_week \  
/opt/VRTS/bin/sfrac_ckptcreate
```

where:

- ◆ *minute* - numeric values from 0-59 or \*
- ◆ *hour* - numeric values from 0-23 or \*
- ◆ *day\_of\_month* - numeric values from 1-31 or \*
- ◆ *month\_of\_year* - numeric values from 1-12 or \*
- ◆ *day\_of\_week* - numeric values from 0-6, with 0=Sunday or \*

Each of these variables can either be an asterisk (meaning all legal values) or a list of elements separated by commas. An element is either a number or two numbers separated by a hyphen (meaning an inclusive range).

- ◆ See the `sfrac_ckptcreate(1M)`, `cron(1M)`, and `crontab(1)` manual pages for more information.



## Scheduling Storage Checkpoint creation in a cron Job

- ◆ To create a Storage Checkpoint twice a day, at 5:00 a.m. and 7:00 p.m., every Monday through Friday, include the following entry in your crontab file:

```
0 5,19 * * 1-5 /opt/VRTS/bin/sfrac_ckptcreate -S DB12 \  
-H $ORACLE_HOME -o online
```

- ◆ To create a Storage Checkpoint at 11:30 p.m., on the 1st and 15th day of each month, include the following entry in your crontab file:

```
30 23 1,15 * * /opt/VRTS/bin/sfrac_ckptcreate -S DB12 \  
-H $ORACLE_HOME -o online
```

- ◆ To create a Storage Checkpoint at 1:00 a.m. every Sunday, with a prefix name of cronckpt\_, and while the database is offline, include the following entry in your crontab file:

```
0 1 * * 0 /opt/VRTS/bin/sfrac_ckptcreate -S DB12 \  
-H $ORACLE_HOME -o offline
```



## Mounting Storage Checkpoints Using `sfrac_ckptmount`

You can use the VERITAS Storage Foundation for Oracle RAC `sfrac_ckptmount` command to mount a Storage Checkpoint for an Oracle database from the command line.

### Usage Notes

- ◆ The `sfrac_ckptmount` command is used to mount a Storage Checkpoint into the file system namespace. Mounted Storage Checkpoints appear as any other file system on the machine and can be accessed using all normal file system based commands.
- ◆ See the `sfrac_ckptmount(1M)` manual page for more information.

#### ▼ To mount Storage Checkpoints with the read/write option

Use the `sfrac_ckptmount` command as in the following example:

```
$ /opt/VRTS/bin/sfrac_ckptmount -S DB12 -c Checkpoint_971672042 \  
-m /tmp/ckpt_rw -o rw  
Creating Storage Checkpoint on /tmp/ckpt_rw/share/oradata \  
with name Checkpoint_971672042_wr001
```

#### ▼ To mount Storage Checkpoints with the read-only option

Use the `sfrac_ckptmount` command as in the following example:

```
$ /opt/VRTS/bin/sfrac_ckptmount -S DB12 -c Checkpoint_971672042 \  
-m /ckptmnt -o ro
```

## Unmounting Storage Checkpoints Using `sfrac_ckptumount`

You can use the VERITAS Storage Foundation for Oracle RAC `sfrac_ckptumount` command from the command line to unmount a Storage Checkpoint for an Oracle database.

### Usage Notes

- ◆ The `sfrac_ckptumount` command is used to unmount a mounted Storage Checkpoint from the file system namespace. Mounted Storage Checkpoints appear as any other file system on the machine and can be accessed using all normal file system based commands. When mounted Storage Checkpoints are not required, they can be unmounted.
- ◆ See the `sfrac_ckptumount(1M)` manual page for more information.

### ▼ To unmount Storage Checkpoints

Use the `sfrac_ckptumount` command as in the following example:

```
$ /opt/VRTS/bin/sfrac_ckptumount -S DB12 \  
-c Checkpoint_971672042_wr001
```



## Performing Storage Rollback Using `sfrac_ckptrollback`

You can use the VERITAS Storage Foundation for RAC `sfrac_ckptrollback` command to roll back an Oracle instance to a Storage Checkpoint.

### Usage Notes

- ◆ The `sfrac_ckptrollback` rolls an Oracle database back to a specified Storage Checkpoint. You can perform a storage rollback for the entire database, a specific tablespace, or list of datafiles.
- ◆ Database rollback for the entire database requires that all database instances be stopped before storage rollback commences. The `sfrac_ckptrollback` command does not commence if any Oracle database instance is active. However, to perform a storage rollback of a tablespace or datafile, only the tablespace or datafile to be rolled back must be offline (not the entire database).
- ◆ Use the `sfrac_ckptumount` command to unmount a checkpoint mounted as read-only before using that checkpoint with the `sfrac_ckptrollback` command.
- ◆ See the `sfrac_ckptrollback(1M)` manual page for more information.

#### ▼ To roll back an offline Oracle instance to a Storage Checkpoint

Use the `sfrac_ckptrollback` command as in the following example:

```
$ /opt/VRTS/bin/sfrac_ckptrollback -S DB12 -H $ORACLE_HOME \  
-c Checkpoint_903937870
```

#### ▼ To roll back a tablespace to a Storage Checkpoint

Use the `sfrac_ckptrollback` command with the `-T` option as in the following example:

```
$ /opt/VRTS/bin/sfrac_ckptrollback -S DB12 -H $ORACLE_HOME \  
-T DATA01 -c Checkpoint_903937870
```

---

**Note** If the Oracle instance is running, you must take the tablespace offline before running this command. If the tablespace is online, the command will fail.

---



### ▼ To roll back datafiles to a Storage Checkpoint

Use the `sfrac_ckptrollback` command with the `-F` option as in the following example:

```
$ /opt/VRTS/bin/sfrac_ckptrollback -S DB12 -H $ORACLE_HOME \  
-F /share/oradata1/data01.dbf,/share/oradata2/index01.dbf \  
-c Checkpoint_903937870
```

---

**Note** If the Oracle instance is running, you must take the datafile offline before running this command. If the datafile is online, the command will fail.

---

## Removing Storage Checkpoints Using `sfrac_ckptremove`

You can use the VERITAS Storage Foundation for Oracle RAC `sfrac_ckptremove` command at the command line to remove a Storage Checkpoint for an Oracle database.

### Usage Notes

- ◆ The `sfrac_ckptremove` command is used to remove a Storage Checkpoint from the file system, or file systems, it is associated with. The Storage Checkpoint must have been created using the `sfrac_ckptcreate(1M)` command.
- ◆ See the `sfrac_ckptremove(1M)` manual page for more information.
- ◆ You must unmount the Storage Checkpoint before you can remove it.

### ▼ To remove Storage Checkpoints

Use the `sfrac_ckptremove` command as follows:

```
$ /opt/VRTS/bin/sfrac_ckptremove -S DB12 \  
-c Checkpoint_971672042_wr001
```





# Using Storage Mapping

---

8

The storage mapping feature is available with SFRAC and enables you to map datafiles to physical devices. To obtain and view detailed storage topology information, use the `vxstorage_stats` command or the VERITAS Storage Foundation *for Oracle* GUI. You can also use the Oracle Enterprise Manager to access storage mapping information.

## Understanding Storage Mapping

Access to mapping information allows for a detailed understanding of the storage hierarchy in which files reside. This information is critical to effectively evaluate I/O performance.

Mapping files to their underlying device is straightforward when datafiles exist directly on a raw device. With the introduction of host-based volume managers and sophisticated storage subsystems that provide RAID features, mapping files to physical devices is more difficult.

With the VERITAS Storage Mapping option, you can map datafiles to physical devices. VERITAS Storage Mapping relies on VERITAS Federated Mapping Service (VxMS), a library that assists in the development of distributed SAN applications that must share information about the physical location of files and volumes on a disk.

The VERITAS Storage Mapping option supports the Oracle set of storage APIs called Oracle Mapping (ORAMAP) that lets Oracle determine the mapping information for files and devices.

Oracle provides a set of dynamic performance views (`v$` views) that show the complete mapping of a file to intermediate layers of logical volumes and physical devices. These views enable you to locate the exact disk on which any specific block of a file resides. Use these mappings, along with device statistics, to evaluate I/O performance.

The VERITAS Storage Mapping option supports a wide range of storage devices and allows for “deep mapping” into EMC arrays. Deep mapping information identifies the physical disks that comprise each LUN and the hardware RAID information for the LUNs.

View information on storage mapping topology and I/O statistics using:



- ◆ The `vxstorage_stats` command. This command displays the complete I/O topology mapping of a specific datafile from logical volumes down to actual physical devices.
- ◆ The VERITAS Storage Foundation *for Oracle* GUI. This GUI performs file mapping and displays information on storage mapping topology and I/O statistics.

The Oracle Enterprise Manager GUI also displays storage mapping information after file mapping takes place. Oracle Enterprise Manager does not display details on I/O statistics. Unlike the information displayed in the VERITAS Storage Foundation *for Oracle* GUI, the information displayed in Oracle Enterprise Manager may be “stale” or out of date.

## Verifying VERITAS Storage Mapping Setup

Before using the VERITAS Storage Mapping option, verify the feature setup:

1. Verify that you have a license key for the storage mapping option.

```
# /opt/VRTS/bin/vxlictest -n "VERITAS Mapping Services" -f \  
"Found_Edi_map"  
Found_Edi_map feature is licensed
```

2. Verify that the VRTSvxmsa package is installed.

```
# swlist VRTSvxmsa  
# VRTSvxmsa                2.1-REV=2002.08.15 VxMS Application  
                           Deployment Package  
VRTSvxmsa.admin           VERITAS Federated Mapping Service  
VRTSvxmsa.libraries       VxMS libraries  
VRTSvxmsa.logging         VxMS logging  
VRTSvxmsa.plugins         VxMS plugins
```

3. Verify the VRTSormap package is installed:

```
# swlist VRTSormap  
# Initializing...  
# Contacting target "hpslia20"...  
#  
# Target: hpslia20:/  
#  
# VRTSormap 4.1 VERITAS Database Edition Utilities for RAC  
.
```

## Using vxstorage\_stats

The `vxstorage_stats` command displays detailed storage mapping information and I/O statistics about one or more VxFS files. The mapping information and I/O statistics exist only for VxFS files and VxVM volumes.

In `vxstorage_stats` command output, I/O topology information appears first followed by summary statistics for each object.

Some highlights on this command:

- ◆ You must log in as the database administrator (typically, the user ID `oracle`).
- ◆ The `-m` option displays the I/O topology for the specified file.
- ◆ The `-s` option displays the file statistics for the specified file.
- ◆ The `-c count` option specifies the number of times to display statistics.
- ◆ The `-i interval` option specifies the interval frequency for displaying updated I/O statistics.
- ◆ The `-f filename` option specifies the file to display I/O mapping and statistics for.
- ◆ Refer the `vxstorage_stats(1m)` online manual page for more information.

## Displaying Storage Mapping Information

Use the `vxstorage_stats` command with the `-m` option to display storage mapping information:

```
$ /opt/VRTSormap/bin/vxstorage_stats -m -f file_name
```

For example:

```
$ /opt/VRTSormap/bin/vxstorage_stats -m -f /oradata/system01.dbf
```

---

**Note** For file type (`fi`), the `SIZE` column is number of bytes; for volume (`v`), plex (`p1`), sub-disk (`sd`), and physical disk (`da`), the `SIZE` column is in 512-byte blocks. Stripe sizes are given in sectors.

---

## Displaying I/O Statistics Information

To display I/O statistics information, use the `vxstorage_stats` command with the `-s` option:

```
$ /opt/VRTSormap/bin/vxstorage_stats -s -f file_name
```

For example:



```
$ /opt/VRTSormap/bin/vxstorage_stats -s -f \  
/data/system01.dbf/PRODqio/PRODqiotbs
```

To display storage mapping and I/O statistics information at repeated intervals, use the `vxstorage_stats` command with the `-i interval` and `-c count` options. The `-i interval` option specifies the interval frequency for displaying updated I/O statistics and the `-c count` option specifies the number of times to display statistics:

```
$ /opt/VRTSormap/bin/vxstorage_stats [-m] [-s] \  
[-i interval -c count ] -f file_name
```

For example, type the following command to display statistics two times with a time interval of two seconds:

```
$ /opt/VRTSormap/bin/vxstorage_stats -s -i2 -c2 \  
-f /data/system01.dbf
```

## Oracle File Mapping (ORAMAP)

VERITAS has defined and implemented two libraries: `libvxoramap_64.so` (HPUX PA) and `libvxoramap_64.sl` (HPUX IA). These two libraries provide a mapping interface to Oracle9i release 2 or a later release. `libvxoramap_64.so` serves as a bridge between the Oracle set of storage APIs (ORAMAP) and VERITAS Federated Mapping Service (VxMS), a library that assists in the development of distributed SAN applications that must share information about the physical location of files and volumes on a disk.

The VERITAS Storage Mapping option enables you to view the complete I/O topology mapping of datafiles from logical volumes to actual physical devices. Use this information to determine the exact location of an Oracle data block on a physical device and to help identify hot spots.

---

**Note** Mapping requires Oracle 9.2.0.3 or a later version.

---

### Mapping Components

This section covers the mapping components in the System Global Area (SGA) and the representation of these components by Oracle. You must understand these components to interpret the mapping information in Oracle dynamic performance views.

The mapping information in the dynamic performance views consists of:

- ◆ File components

A mapping file component is a mapping structure describing a file. This component provides a set of attributes for a file, including the size, number of extents, and type. `v$map_file` exports file components to the user.

- ◆ File extent components

A mapping file extent component describes a contiguous group of blocks residing on one element. The description specifies the device offset, extent size, file offset, extent type (`Data` or `Parity`), and name of the element where the extent resides.

- ◆ Element components

A mapping element component is a mapping structure that describes a storage component within the I/O stack. Elements can be mirrors, stripes, partitions, RAID5, concatenated elements, and disks.

This component contains information about the mapping structure, such as the size, type, number of subelements, and a brief description of the element.

`v$map_element` exports element components are exported to the user.



- ◆ Subelement components

A mapping subelement component describes the link between an element and the next element in the I/O stack. The subelement component contains the subelement number and size, as well as the element name for the subelement and element offset. Subelement components are exported to the user through `v$map_subelement`.

These four types of mapping components describe the mapping information for an Oracle instance.

## Storage Mapping Views

The mapping information that is captured appears in Oracle dynamic performance views. Review the brief descriptions of these views below; refer to Oracle documentation for details.

---

<b>View</b>	<b>Description</b>
<code>V\$MAP_LIBRARY</code>	Contains a list of all mapping libraries dynamically loaded by the external process.
<code>V\$MAP_FILE</code>	Contains a list of all file mapping structures in the shared memory of the instance.
<code>V\$MAP_FILE_EXTENT</code>	Contains a list of all the file extent mapping structures in the shared memory of the instance.
<code>V\$MAP_ELEMENT</code>	Contains a list of all the element mapping structures in the SGA of the instance.
<code>V\$MAP_EXT_ELEMENT</code>	Contains supplementary information for all element mapping structures.
<code>V\$MAP_SUBELEMENT</code>	Contains a list of all subelement mapping structures in the shared memory of the instance.
<code>V\$MAP_COMP_LIST</code>	Describes the component list associated with the element name.
<code>V\$MAP_FILE_IO_STACK</code>	Contains the hierarchical arrangement of storage containers for the file. This information appears as a series of rows. Each row represents a level in the hierarchy.

---



## Verifying Oracle File Mapping Setup

1. Verify \$ORACLE\_HOME is ready for Oracle file mapping (ORAMAP):

```
# cd $ORACLE_HOME/rdbms/filemap/bin
# ls -l
-r-xr-x--- 1 root system 900616 Apr 08 19:16 fmpu1
-r-sr-xr-x 1 root system 14614 Apr 08 19:16 fmpu1hp
```

2. Confirm the following items and make the appropriate corrections:

- ◆ root owns fmpu1hp and the setud bit is set.
- ◆ The permissions for fmpu1hp are -r-sr-xr-x.
- ◆ The permissions for fmpu1 are -r-xr-x---

## Enabling Oracle File Mapping

1. Ensure that the filemap.ora file exists and contains a valid entry for the VERITAS mapping library for Oracle storage mapping:

```
# cd $ORACLE_HOME/rdbms/filemap/etc
# cat filemap.ora
```

For 64-bit Oracle, the filemap.ora file should contain the following setting:

```
lib=VERITAS:/opt/VRTSormap/lib/libvxoramap_64.so for PA
lib=VERITAS:/opt/VRTSormap/lib/libvxoramap_64.s1 for IA
```

2. After verifying that the system uses the VERITAS library for Oracle storage mapping, set the file\_mapping initialization parameter to true:

```
SQL> alter system set file_mapping=true;
```

The default setting for the file\_mapping initialization parameter is false. You do not need to shut down the instance to set this parameter. Setting file\_mapping=true starts the fmon background process.

---

**Note** To enable storage mapping whenever starting an instance, set the file\_mapping initialization parameter to true in the init.ora file.

---



## Accessing Dynamic Performance Views

1. Confirm the VERITAS mapping library for Oracle file mapping is enabled:

```
SQL> select lib_idx idx, lib_name name, vendor_name vname, \
        path_name path from v$map_library;
```

IDX	NAME	VNAME	PATH
1	Veritas ORAMAP API	Veritas	/opt/VRTSormap/lib/libvxoramap.so

2. After enabling storage mapping, map Oracle datafiles using the DBMS\_STORAGE\_MAP package.

For more information about features and capabilities of the DBMS\_STORAGE\_MAP package, see the Oracle documentation.

3. Use SQL commands to display the mapping information captured in the dynamic performance views. To display the contents of v\$map\_file, type:

```
SQL> select file_name name, file_map_idx idx, \
        file_status status, file_type type, file_structure str, \
        file_size fsize, file_nexts nexts from v$map_file;
```

To display the contents of v\$map\_file\_extent, type:

```
SQL> select * from v$map_file_extent;
```

To display the contents of v\$map\_element, type:

```
SQL> select elem_idx idx, elem_name, elem_type type, elem_size, \
        elem_nsubelem nsub, elem_descr, stripe_size from \
        v$map_element;
```

To display the contents of v\$map\_subelement, type:

```
SQL> select * from v$map_subelement;
```

To display all the elements within the I/O stack for a specific file, type:

```
SQL> with fv as
2      (select file_map_idx, file_name from v$map_file
3       where file_name = '/ora92/dbs/qio10m.dbf')
4  select
5      fv.file_name, lpad(' ', 4 * (level - 1)) || \
        el.elem_name elem_name, el.elem_size, el.elem_type, \
        el.elem_descr
6  from
7      v$map_subelement sb, v$map_element el, fv,
8      (select unique elem_idx from v$map_file_io_stack io, fv
9       where io.file_map_idx = fv.file_map_idx) fs
```



```

10     where el.elem_idx = sb.child_idx
11     and fs.elem_idx = el.elem_idx
12     start with sb.parent_idx in
13     (select distinct elem_idx
14     from v$map_file_extent fe, fv
15     where fv.file_map_idx = fe.file_map_idx)
16     connect by prior sb.child_idx = sb.parent_idx;

```

## Using Oracle Enterprise Manager

Oracle Enterprise Manager is a Web console for managing Oracle databases. Use this GUI to perform administrative tasks such as creating tablespaces, tables, and indexes; managing user security; and backing up and recovering your database. Oracle Enterprise Manager enables you to view information on the performance and status of a database instance. Refer to Oracle documentation for details on Oracle Enterprise Manager.

From Oracle Enterprise Manager, view storage mapping information and a graphical display of the storage layout:

1. Start Oracle Enterprise Manager and select a database from the object tree of the Oracle Enterprise Manager Console.
2. Expand the **Databases** icon and select the desired database.  
The **Database Connect Information** window appears.
3. Enter a user name and password to log in to the database and click **OK**.
4. In the object tree, expand the **Storage** icon.
5. Under the **Storage** icon, expand the **Datafiles** icon.
6. Select the datafile for which you want to view storage layout information.
7. In the right pane, click the **Storage Layout** tab.
8. Expand the objects to display their storage layout.

Within the Oracle Enterprise Manager Console, you can point to an object on the screen to view a description in a pop-up field. If an object name or path appears truncated, point to it and the pop-up field will display the full object name and path. To view detailed information about the object, right-click the object and click **View Details**.



9. By default, storage layout information appears in tabular format; the **Tabular Display** icon is selected. To view a graphical display of the storage layout, click the **Graphical Display** icon.
10. Expand the objects to display their storage layout information graphically.
11. To exit, choose **Exit** from the **File** menu.



## Configuring Arrays for Storage Mapping and Statistics

SFRAC provides “deep” mapping information and performance statistics for supported storage arrays. Deep mapping involves identifying the physical disks comprising each LUN and hardware RAID information for the LUNs.

---

**Note** Deep mapping requires Oracle 9.2.0.3. or a later version.

---

VERITAS Array Integration Layer (VAIL) software interfaces third-party hardware storage arrays with VERITAS storage software. VAIL providers are software modules that enable VERITAS applications to discover, query, and manage third-party storage arrays. The `vx_emc_symmetrix` provider manages EMC Symmetrix arrays.

For the most up-to-date array support information, see the appropriate hardware compatibility list (HCL) on the VERITAS Technical Support Web page at:

<http://support.veritas.com>

If you want to use storage array information accessible through VAIL providers, install VAIL and perform any required configuration for the storage arrays and VAIL providers. To use deep mapping services and performance statistics for supported storage arrays, install both VAIL and VERITAS Mapping Services (VxMS).

You must install required third-party array CLIs and APIs on the host where you are going to install VAIL. If you install any required CLI or API after installing VAIL, rescan the arrays to enable SFRAC to discover them. For details on supported array models, see the *VERITAS Array Integration Layer Array Configuration Guide*.





# Adding and Removing SFRAC Nodes

---

9

An SFRAC cluster can have as many as four nodes; SFRAC enables you to add and remove nodes. The procedures show how to add a node to a two-node cluster.

## Adding a Node

You can add a node to your existing Storage Foundation *for Oracle RAC* cluster in phases:

- ◆ Phase 1: Checking the new node meets the requirements for installation of SFRAC.
- ◆ Phase 2: Physically adding a new node to the cluster.
- ◆ Phase 3: Installing SFRAC on the new node.
- ◆ Phase 4: Starting VxVM by running `vxinstall`.
- ◆ Phase 5: Configuring LLT, GAB, VCSMM, ODM, and VXFEN drivers.
- ◆ Phase 6: Configuring the CVM service group to reflect the new cluster node.
- ◆ Phase 7: Setting up Oracle (if Oracle is installed on shared storage) or installing Oracle (if Oracle is installed locally on the existing cluster nodes).
- ◆ Phase 8: Configuring a new Oracle instance on the new node.



## Phase 1 - Checking System Requirements for New Node

Verify the new nodes joining the cluster meet all of the requirements for installing and using SFRAC.

- ✓ The new node must have the identical operating system and patch level as the existing nodes.
- ✓ Use a text window of 80 columns (minimum) by 24 lines (minimum); 80 columns by 24 lines is the recommended size for the optimum display of the `installfrac` script.
- ✓ Verify the `/etc/fstab` file contains only valid entries, each of which specifies a file system to mount.

## Phase 2 - Physically Adding a New Node to the Cluster

The new node must have the identical operating system and patch level as the existing nodes. When you physically add the new node to the cluster, make sure the node is connected to the same shared storage devices as the existing nodes and has private network connections to two independent switches for the cluster. Refer to the *VERITAS Cluster Server Installation Guide*.

After installing SFRAC on the new node and starting VxVM, the new node can access the same shared storage devices (including coordinator disks). If the new node does not see the same disks as the existing nodes, the new node cannot join the cluster, as indicated by a CVM error on the console.

## Phase 3 - Installing SFRAC on the New Node

Phase 3 consists of checking requirements for installing SFRAC and performing the installation.

### Checking the New Node for Installation

1. Log in as `root` user to a cluster node or a remote node in the same subnet as the cluster.
2. Insert the VERITAS software disc into the CD drive and determine the appropriate CD-ROM device:

```
# ioscan -fn -C disk
```

For example, the listing may indicate the block device of the CD-ROM drive is `/dev/dsk/c0t0d0`; make a note of the device file as it applies to your system.





- If necessary, create a directory under which to mount the software disc and mount the disc:

```
# mkdir -p /cdrom
# mount -F cdfs /dev/dsk/c0t0d0 /cdrom
```

---

**Note** VERITAS advises using the `-F cdfs` option for the mount command.

---

Alternately, you can mount the disc using Portable File System (PFS):

- ◆ Check whether the PFS daemons are running:

```
# ps -ef | grep pfs_mountd
```

- ◆ If the daemons are running, look for output resembling the example below:

```
# ps -ef | grep pfs
root 17796 17795 0 Mar 23 ? 5:51 pfsd.rpc
root 17794 17793 0 Mar 23 ? 0:00 pfs_mountd.rpc
root 17795      1 0 Mar 23 ? 0:00 pfsd
root 17793      1 0 Mar 23 ? 0:00 pfs_mountd
```

- ◆ If the daemons are not running, start them:

```
# nohup pfs_mountd &
# nohup pfsd &
```

- ◆ If necessary, create a directory under which to mount the software disc:

```
# mkdir -p /cdrom
```

- ◆ Mount the CD-ROM. For example, to mount the CD-ROM to the `/cdrom` mount point, type:

```
# /usr/sbin/pfs_mount -o iso /dev/dsk/c0t0d0 /cdrom
```

`/dev/dsk/c0t0d0` is the block device file of the drive.

- Verify the mounting process:

```
# mount
```

In the output, look for a line that resembles the example below:

```
/cdrom on slias15:/cdrom ro,nointr,retrans=3,timeo=7,
  wsize=2048,proto=udp on Wed Mar 23 14:28:37 2005
```

- Navigate to the `/cdrom` folder containing the `installfrac` script:

```
# cd /cdrom/storage_foundation_for_oracle_rac
```



6. Run the `installsfrac` script with the `precheck` option to verify the current operating system level, patch level, licenses, and disk space are adequate for a successful installation:

```
# ./installsfrac -precheck saturn
```

The `precheck` function of the utility proceeds without user interaction. Upon completion, the utility displays the results of the verification and saves the results in a log file.

If the verification is successful, proceed to the next section to run `installsfrac` with the `-installonly` option. If the `precheck` function indicates a requirement for licensing, add the license when running the installation utility. The `precheck` function may prompt you for other requirements.

## Installing SFRAC on New Node Without Configuration

1. On the new node, use the `-installonly` option to install SFRAC without performing configuration:

```
# ./installsfrac -installonly
```

The new node will use the configuration from the existing cluster nodes.

2. Enter the name of the new node.
3. After the script performs initial checks, confirm to start the installation.
4. Review the output as the script checks for the presence of infrastructure component packages, `VRTScpi` and `VRTSvlic`, and installs them if they are not present.
5. Enter a license key if one is not present:

```
Checking SFRAC license key on saturn ..... not licensed
Enter a SFRAC license key for saturn: [?] XXXX-XXXX-XXXX-XXXX-XX
Registering VERITAS Storage Foundation for Oracle RAC PERMANENT
key on saturn
```

```
Do you want to enter another license key for saturn? [y,n,q,?]
(n)
```

```
SFRAC licensing completed successfully.
```

6. Review the output as the script lists the packages and patches to install and checks whether any of them are present on the node.
7. After the script installs the packages and patches, note the location of the summary, log, and response files in the output.



---

**Note** Ignore the message advising you to run `installsfrac -configure`. You must manually configure the new node joining the cluster.

---

## Phase 4 - Running vxinstall

1. To start VERITAS Volume Manager on the new node, use the `vxinstall` utility:

```
# vxinstall
```

2. VxVM uses license keys to control access. As you run the utility, answer "n" when prompted about licensing; you installed the appropriate license when you ran the `installsfrac` utility:

```
Licensing information:
  System host ID: 0083119567
  Host type: ia64 hp server rx2600
```

```
Are you prepared to enter a license key [y,n,q,?] (default: n)
```

```
Volume Manager Installation
Menu: VolumeManager/Install
```

```
VxVM will use the following format to name disks on the host:
```

```
<enclosure name>_<disk number>
```

In the above format, `<enclosure_name>` is the logical name of the enclosure to which the disk belongs. VxVM assigns default enclosure names which can be changed according to the user requirements.

Some examples would be:

```
hitachi0_2 - second disk detected in enclosure 'hitachi0'
enggdept_2 - second disk detected in enclosure 'enggdept'
dgc1_1 - first disk detected in enclosure 'dgc1'
jbod_1 - first disk detected in the jbod category
```

For non-public loop disks (Ex. FC disks directly connected to the host or through hubs), you could choose to use the `c#t#d#s#` naming format.

```
Do you want to use enclosure based names for all disks ?
[y,n,q,?] (default: n)
```



```
Populating VxVM DMP device directories ....
Generating list of attached disk arrays ....
```

Volume Manager Installation

Menu: VolumeManager/Install/VolumeManager/Install

The Volume Manager has detected the following categories of storage connected to your system:

```
Enclosures:      Disk      OTHER_DISKS
```

Hit RETURN to continue.

Volume Manager Installation

Menu: VolumeManager/Install/VolumeManager/Install

You will now be asked if you wish to use Quick Installation or Custom Installation. Custom Installation allows you to select how the Volume Manager will handle the installation of each disk attached to your system.

Quick Installation examines each disk attached to your system and attempts to create volumes for all non-LVM and non-root disks.

If you want to exclude any devices from being seen by VxVM or not be multipathed by vxdmp then use the Prevent multipathing/Suppress devices from VxVM's view option, before you choose Custom Installation or Quick Installation.

If you do not wish to use some disks with the Volume Manager, or if you wish to reinitialize some disks, use the Custom Installation option. Otherwise, we suggest that you use the Quick Installation option.

Hit RETURN to continue.

Volume Manager Installation Options

Menu: VolumeManager/Install/VolumeManager/Install

```
1      Quick Installation
2      Custom Installation
3      Prevent multipathing/Suppress devices from VxVM's view

?      Display help about menu
??     Display help about the menuing system
q      Exit from menus
```



```

Select an operation to perform: 1

Volume Manager Quick Installation
Menu: VolumeManager/Install/VolumeManager/Install/QuickInstall

Volume Manager Quick Installation
Menu: VolumeManager/Install/VolumeManager/Install/
      QuickInstall/Disk
Generating list of disks in disk array Disk....

<excluding root disk c2t1d0s2>
<excluding LVM disk c2t0d0>
No disks were found in disk array Disk !
Hit RETURN to continue.

Volume Manager Quick Installation
Menu: VolumeManager/Install/VolumeManager/Install/
      QuickInstall/OTHER_DISKS
Generating list of disks in disk array OTHER_DISKS....

-

Volume Manager Quick Installation For Disk Array OTHER_DISKS
Menu: VolumeManager/Install/VolumeManager/Install/
      QuickInstall/OTHER_DISKS

The Volume Manager has detected the following disks on disk
array OTHER_DISKS:

c4t0d0 c4t0d1 c4t0d2 c4t0d3 c4t0d4 c4t0d5 c4t0d6

Initialize all non-LVM/non-root disks on this disk array ?
(destroys data on these disks) [y,n,q,?] (default: n) y

Are you sure ? (destroys data on these disks) [y,n,q,?]
(default: n) y

Volume Manager will now initialize all the disks on this disk
array
Hit RETURN to continue.

Volume Manager Quick Installation

```



```
Menu: VolumeManager/Install/VolumeManager/Install/  
QuickInstall/OTHER_DISKS/Init
```

```
Use default disk names for these disks? [y,n,q,?] (default: y)
```

```
The c4t0d0 disk will be given disk name disk01
```

```
The c4t0d1 disk will be given disk name disk02
```

```
The c4t0d2 disk will be given disk name disk03
```

```
The c4t0d3 disk will be given disk name disk04
```

```
The c4t0d4 disk will be given disk name disk05
```

```
The c4t0d5 disk will be given disk name disk06
```

```
The c4t0d6 disk will be given disk name disk07
```

```
Hit RETURN to continue.
```

```
Volume Manager Quick Installation
```

```
Menu: VolumeManager/Install/VolumeManager/Install/QuickInstall
```

```
The following is a summary of your choices.
```

```
c4t0d0          disk01  
c4t0d1          disk02  
c4t0d2          disk03  
c4t0d3          disk04  
c4t0d4          disk05  
c4t0d5          disk06  
c4t0d6          disk07
```

```
Is this correct [y,n,q,?] (default: y)
```

```
The Volume Manager is now reconfiguring (partition phase)...
```

```
Volume Manager: Initializing c4t0d0 as a new disk.
```

```
Volume Manager: Initializing c4t0d1 as a new disk.
```

```
Volume Manager: Initializing c4t0d2 as a new disk.
```

```
Volume Manager: Initializing c4t0d3 as a new disk.
```



```
Volume Manager: Initializing c4t0d4 as a new disk.
Volume Manager: Initializing c4t0d5 as a new disk.
Volume Manager: Initializing c4t0d6 as a new disk.
The Volume Manager is now reconfiguring (initialization phase)...
Volume Manager: Adding disk01 (c4t0d0) as a new disk.
The Volume Daemon has been enabled for transactions.
Volume Manager: Adding disk02 (c4t0d1) as a new disk.
Volume Manager: Adding disk03 (c4t0d2) as a new disk.
Volume Manager: Adding disk04 (c4t0d3) as a new disk.
Volume Manager: Adding disk05 (c4t0d4) as a new disk.
Volume Manager: Adding disk06 (c4t0d5) as a new disk.
Volume Manager: Adding disk07 (c4t0d6) as a new disk.
Starting the relocation daemon, vxrelocd.
```

**3. Decline to set up a disk group for the node:**

```
Do you want to setup a system wide default disk group? [y,n,q,?]
(default: y) n
The installation is successfully completed.
```

**4. Verify the daemons are up and running:**

```
# vxdisk list
```

Make sure the output displays the shared disks without errors.



## Phase 5 - Configuring LLT, GAB, VCSMM, ODM, and VXFEN Drivers

1. On the new node, set the shared memory parameter using SAM; refer to the *Oracle9i Installation Guide* for details. The value of the shared memory parameter becomes effective when the system restarts.
2. Edit the `/etc/llthosts` file on the existing nodes. Using `vi` or another text editor, add the line for the new node to the file. The file resembles:

```
0 galaxy
1 nebula
2 saturn
```

3. Copy the `/etc/llthosts` file from one of the existing nodes to the new node. The `/etc/llthosts` file must be identical on all cluster nodes.
4. Create the `/etc/llttab` file on the new node. For example:

```
set-node saturn
set-cluster 7
link eth0 eth0 - ether --
link eth1 eth1 - ether --
```

Except for the first line that refers to the node, the file resembles the `/etc/llttab` files on the existing nodes. The second line *must* be the same line on *all* nodes.

5. Use `vi` or another text editor to create the `/etc/gabtab` file on the new node. The file resembles:

```
/sbin/gabconfig -c -nN
```

Where *N* represents the number of nodes in the cluster. For a three-node cluster, the value of *N* is 3.

6. Edit the `/etc/gabtab` file on each of the existing nodes, changing the content to match the file on the new node.
7. Set up the `/etc/vcsmmtab` and `/etc/vxfendg` files on the new node by copying them from one of the existing nodes:

```
# scp galaxy:/etc/vcsmmtab /etc
# scp galaxy:/etc/vxfendg /etc
```



8. Start LLT and GAB on the new node:

```
# /sbin/init.d/llt start
# /sbin/init.d/gab start
```

9. On the new node, start the VXFEN, VCSMM, and LMX drivers. Use the following commands in this order:

```
# /sbin/init.d/vxfen start
# /sbin/init.d/vcsmm start
# /sbin/init.d/lmx start
```

10. On the new node, start the ODM driver. Use the following commands in this order:

```
# kcmodule vxgms=loaded
# kcmodule odm=loaded
# /sbin/init.d/odm start
```

11. On the new node, verify that the GAB port memberships are a, b, d, and o:

```
# gabconfig -a
GAB Port Memberships
=====
Port a gen      df204 membership 012
Port b gen      df20d membership 012
Port d gen      df20a membership 012
Port o gen      df207 membership 012
```

## Phase 6 - Configuring CVM

You can modify the VCS configuration by editing `/etc/VRTSvcs/conf/config/main.cf` directly, using the VCS GUI (Cluster Manager), or using the command line, as illustrated in the following example. Refer to the *VERITAS Cluster Server User's Guide* for details on configuring VCS.

1. On one of the existing nodes, check the groups dependent on the CVM service group:

```
# hagrps -dep cvm
Parent      Child      Relationship
oradb1_grp  cvm        online local firm
oradb2_grp  cvm        online local firm
```

2. Enable write access to the VCS configuration:

```
# haconf -makerw
```



3. Add the new node to the cluster:

```
# hasys -add saturn
```

4. If the ClusterService service group is configured, add the new node to its system list and specify a failover priority of "2":

```
# hagrps -modify ClusterService SystemList -add saturn 2
```

5. If the ClusterService service group is configured, add the new node to its AutoStartList:

```
# hagrps -modify ClusterService AutoStartList galaxy nebula saturn
```

6. Add the new node to the cvm service group system list and specify a failover priority of "2":

```
# hagrps -modify cvm SystemList -add saturn 2
```

7. Add the new node to the cvm service group AutoStartList:

```
# hagrps -modify cvm AutoStartList galaxy nebula saturn
```

8. Add the new node and node ID (refer to the /etc/llthosts changes in [step 2](#) on page 174) to the cvm\_cluster resource:

```
# hares -modify cvm_clus CVMNodeId -add saturn 2
```

9. If the IP resource is *not* part of the cvm service group, skip to the next step. If the IP is part of the cvm service group, add the new node to the IP resource:

```
# hares -modify listener_ip Address 10.182.2.130 -sys saturn
```

10. If the listener name is the default name, skip this step. Otherwise, add the local listener name to the Netlsnr resource:

```
# hares -modify LISTENER Listener listener_saturn -sys saturn
```

---

**Note** You must similarly add the new node information for other VCS resources in case any additional resources exist.

---

11. Save the new configuration to disk:

```
# haconf -dump -makero
```

12. On each of the existing nodes, enable the nodes to recognize the new node:

```
# /opt/VRTS/bin/vxclustadm -m vcs -t gab reinit
```



## Phase 7 - Setting Up or Installing Oracle

1. On the new node, create a local group and local user for Oracle. Make sure to assign the same group ID, user ID, and home directory as the ones for the current cluster nodes. For example, enter:

```
# groupadd -g 1000 dba
# useradd -g dba -u 1000 -d /oracle oracle
```

Create a password for the oracle user:

```
# passwd oracle
```

2. If Oracle system binaries are installed on shared storage on the existing nodes, skip to [step 3](#). If Oracle is installed locally on the existing nodes, install Oracle on the local disk of the new node. Use the same location where Oracle is installed on the existing nodes.
  - a. Refer to [“Installing Oracle9i Software”](#) on page 65. In that chapter, review instructions to install Oracle Release 2 on the local file system of a node. Install the release and patch level used by the existing nodes.
  - b. Relink the Oracle binary to the VERITAS libraries.
  - c. While installing Oracle on the new node, make sure the `/var/opt/oracle/srvConfig.loc` file is identical to the one for existing nodes, including contents, permissions, and ownership. If necessary, copy the file from one of the other nodes.
  - d. Edit the `listener.ora` file on the new node to specify the IP address (or the virtual IP address) for the new node. For example, edit the section for `LISTENER` to resemble:

```
.
.
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP) (HOST = 192.2.40.23) (PORT =
          1521))
      )
    )
  )
.
.
```



3. If Oracle system binaries are installed on a cluster file system, set up the new node for Oracle.
  - a. Create mount points for the shared file system on the new node; the mount points should have the same name as the one for the shared file system on the existing cluster nodes:

```
# mkdir /orasrv
# mkdir /oracle
```

- b. Mount the shared file systems on the new node:

```
# mount -F vxfs -o cluster,largefiles
    /dev/vx/dsk/orasrv_dg/srvm_vol /orasrv

# mount -F vxfs -o cluster,largefiles
    /dev/vx/dsk/orabinvol_dg/orabinvol /oracle
```

After mounting the file system, change the ownerships again:

```
# chown -R oracle:dba /orasrv
# chown -R oracle:dba /oracle
# chmod 775 /oracle
```

- c. On the new node, open a new window and log in as `oracle` user. Edit the `listener.ora` file and add the IP address (or the virtual IP address) for the new node. For example, create a section for `LISTENER_saturn` that resembles:

```
.
LISTENER_saturn =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP) (HOST = 192.2.40.23) (PORT =
          1521))
      )
    )
  )
.
```

- d. While installing Oracle on the new node, make sure the `/var/opt/oracle/srvConfig.loc` file is identical to the one for existing nodes, including contents, permissions, and ownership. If necessary, copy the file from one of the other nodes.

4. Restart the new node:

```
# /usr/sbin/shutdown -r now
```

As the new node boots, the VCS propagates the configuration from the existing cluster nodes to the new node. All the configuration files located in the `/etc/VRTSvcs/conf/config` directory, including `main.cf`, `CVMTypes.cf`, `CFSTypes.cf`, and `OracleTypes.cf`, are identical on each node.

At this point, GAB membership shows membership for all the nodes. All the following ports should be up on all the nodes:

```
# gabconfig -a
GAB Port Memberships
```

```
=====
Port a gen df205 membership 012
Port b gen df20e membership 012
Port d gen df20f membership 012
Port f gen df219 membership 012
Port h gen df211 membership 012
Port o gen df208 membership 012
Port v gen df215 membership 012
Port w gen df217 membership 012
```

5. Verify the CVM group is configured and online on each node (including the new node):

```
# hastatus -sum
-- SYSTEM STATE
-- System      State      Frozen
A galaxy      RUNNING   0
A nebula      RUNNING   0
A saturn      RUNNING   0

-- GROUP STATE
-- Group       System    Probed    AutoDisabled  State
B cvm         galaxy   Y         N              ONLINE
B cvm         nebula   Y         N              ONLINE
B cvm         saturn   Y         N              ONLINE
```

6. On one of the existing nodes, ensure CVM recognizes the new node:

```
# /opt/VRTS/bin/vxclustadm nidmap
Name      CVM Nid    CM Nid    State
galaxy    0          0         Joined: Slave
nebula    1          1         Joined: Master
saturn    2          2         Joined: Slave
```



7. Whether you installed Oracle9i locally or on shared storage, run the Global Services Daemon (`gsd`) in the background on the new node as `oracle` user:

```
$ $ORACLE_HOME/bin/psdctl start
```

## Phase 8 - Configuring New Oracle Instance

1. On an existing node, add a new instance. Refer to the *Oracle9i Installation Guide*. Highlights of the steps to add a new instance include:

- ◆ Logging in as the `oracle` user and connecting to the instance.
- ◆ Creating a new "undotbs" tablespace for the new instance. For example, if the tablespace is for the third instance, name it "undotbs3". If the database uses raw volumes, create the volume first. Use the same size as the one for existing "undotbs" volumes.
- ◆ Create two new "redo" log groups for the new instance. For example, if the tablespace is for the third instance, create the tablespaces "redo3\_1" and "redo3\_2". If the database uses raw volumes, create the volume for the redo logs first. Use the size used by the existing redo volumes.
- ◆ Enable "thread 3" where 3 is the number of the new instance.
- ◆ Prepare the `init{SID}.ora` file for the new instance on the new node.
- ◆ If Oracle is installed locally on the new node, prepare the `bdump`, `cdump`, `udump`, and `pfile` directories.

2. If you use in-depth monitoring for the database, create the table for the database instance. Create the table on the new node. Refer to the *VERITAS Cluster Server Enterprise Agent for Oracle, Installation and Configuration Guide* for instructions on creating the table.

3. Configure the ODM port on the new node.

- a. Unmount the ODM directory to unconfigure port `d`:

```
# /sbin/init.d/odm stop
```

The installation automatically mounts ODM on the new node but does not link it.

- b. Mount the ODM directory. Re-mounting the ODM directory configures port `d` and re-links the ODM libraries with SFRAC:

```
# /sbin/init.d/odm start
```

4. Create a mount point for the shared file system:

```
# mkdir /rac_ts
```

5. From the same node, mount the file system:

```
# mount -F vxfs -o cluster /dev/vx/dsk/rac_dg/rac_vol1 /rac_ts
```

6. Set "oracle" as the owner of the file system, and set "755" as the permissions:

```
# chown oracle:dba /rac_ts
# chmod 755 /rac_ts
```

7. Log in as oracle user and attempt to manually start the new instance; the following example is for a third system:

```
$ export ORACLE_SID=rac3
$ sqlplus '/as sysdba'
sqlplus> startup pfile=/oracle/orahome/dbs/initrac3.ora
```

8. After the new Oracle instance is brought up manually on the new node, place the instance under VCS control.

- a. Add the new node to the SystemList. For example, if the existing nodes (galaxy and nebula) are nodes 0 and 1, the new node (saturn) is node 2:

```
# haconf -makerw
# hagrps -modify oradb1_grp SystemList -add saturn 2
```

- b. Add the new node to the AutoStartList for oradb1\_grp:

```
# hagrps -modify oradb1_grp AutoStartList galaxy nebula saturn
```

- c. Modify the Sid (system ID) and Pfile (parameter file location) attributes of the Oracle resource. For example:

```
# hares -modify VRTdb Sid rac3 -sys Saturn
# hares -modify VRTdb Pfile /oracle/orahome/dbs/initrac3.ora
      -sys Saturn
```

- d. If you created a table for in-depth monitoring, modify the Table attribute of the Oracle resource. For example:

```
# hares -modify VRTdb Table vcstable_saturn -sys saturn
```

- e. Close and save the configuration:

```
# haconf -dump -makero
```



9. From the new node, verify the configuration:

```
# hastop -local
```

VCS takes all resources offline on the new node.

10. Verify all resources come online after starting VCS on the new node:

```
# hastart
```





---

## Removing a Node

You can remove one or more nodes from an SFRAC cluster.

### Stopping gsd

Stop the `gsd` processes on each node you plan to remove from the SFRAC cluster. As `oracle` user, type:

```
$ $ORACLE_HOME/bin/gsdctl stop
```

### Stopping Applications Using CFS Outside of VCS Control

Stop all applications using CFS mounts *not* under VCS control. Ensure that no processes are using the CFS mount point.

```
# fuser -m mount_point
```

Stop any processes using the CFS mount point.

### Stopping VCS

1. Log in as `root` user on the node to remove.
2. Stop VCS on the node.

```
# hastop -local
```

Stopping VCS takes all service groups on the system offline.



## Uninstalling Oracle if Installed Locally (Optional)

If Oracle is installed on a cluster file system, proceed to [“Running the uninstallsfrac Utility”](#) on page 185.

If Oracle is installed locally and you choose *not* to uninstall Oracle, proceed to [“Unlinking the Oracle Binary from VERITAS Libraries \(Required if Oracle is Installed Locally\)”](#) on page 184.

If Oracle is installed locally and you choose to uninstall it, use the Oracle `runInstaller` utility. Run the utility on each node you are removing from the cluster.

1. On one node, log in as `oracle`.

2. Set the `DISPLAY` variable.

If you use the Bourne Shell (`bash`, `sh` or `ksh`):

```
$ export DISPLAY=host:0.0
```

If you use the C Shell (`csh` or `tcsh`):

```
$ setenv DISPLAY host:0.0
```

3. On the first node, run the Oracle9i `runInstaller` utility:

```
$ /cdrom/Disk1/runInstaller
```

As the utility starts up, select the option to uninstall the Oracle software from the local node you are removing. Refer to the *Oracle Installation Guide* for details on the utility.

## Unlinking the Oracle Binary from VERITAS Libraries (Required if Oracle is Installed Locally)

Refer to [“Unlinking the Oracle Binary From VERITAS Libraries”](#) on page 207 for details.

## Running the `uninstallsfrac` Utility

Run the script from any node in the cluster. Prior to invoking the `uninstallsfrac` script, you must take all service groups offline and shut down VCS.

The example in this procedure involves removing SFRAC from the `saturn` node.

1. As `root` user, start the uninstallation:

```
# cd /opt/VRTS/install
# ./uninstallsfrac
```

2. Decline to uninstall SFRAC from *all* systems:

VCS configuration files exist on this system with the following information:

```
Cluster Name: rac_cluster1
Cluster ID Number: 7
Systems: galaxy nebula saturn
Service Groups: cvm oradb1_grp
```

Do you want to uninstall SFRAC from these systems? [y,n,q] (y) **n**

---

**Caution** Be sure to answer "n". Otherwise, the utility begins the process of uninstalling SFRAC from all systems.

---

3. Specify the name of the system on which you are uninstalling SFRAC:

Enter the system names separated by spaces on which to uninstall SFRAC: **saturn**

4. Review the output as the script checks for packages currently installed on your system. The utility also checks for dependencies between packages to determine the packages it can safely uninstall and in which order:

```
Checking SFRAC packages installed on galaxy:
Checking VRTSvxmsa package ..... version 4.1 installed
Checking VRTSvxmsa dependencies ..... no dependencies
Checking VRTSvcsvr package ..... version 4.1 installed
Checking VRTSvcsvr dependencies ..... no dependencies
Checking VRTSvcsvp package ..... version 4.1 installed
Checking VRTSvcsvp dependencies ..... no dependencies
.
.
Checking VRTSgab package ..... version 4.1 installed
Checking VRTSgab dependencies .... VRTSdbac VRTSglm VRTSgms
VRTSvcs VRTSvcsag VRTSvcsmg VRTSvxfen
Checking VRTSllt package ..... version 4.1 installed
```



```
Checking VRTS11t dependencies ... VRTSdbac VRTSgab VRTSglm
VRTSgms VRTSvxfen
```

```
.
```

**5. Confirm to uninstall SFRAC:**

```
uninstallsfrac is now ready to uninstall SFRAC packages.
All SFRAC processes that are currently running will be stopped.
```

```
Are you sure you want to uninstall SFRAC packages? [y,n,q] (y)
```

**6. Review the output as the script stops processes and drivers running on each node:**

```
Stopping SFRAC processes on galaxy:
```

```
Checking VRTSweb process ..... not running
Checking had process ..... running
Stopping had ..... Done
Checking hashadow process ..... not running
Checking CmdServer process ..... running
Killing CmdServer ..... Done
Checking notifier process ..... not running
Checking vcsmm driver ..... vcsmm module loaded
Stopping vcsmm driver ..... Done
```

```
.
```

```
.
```

**7. After the script uninstalls the packages, it describes the location of the summary and log files.**

```
Uninstallation of Storage Foundation for Oracle RAC has completed
successfully.
```

```
The uninstallation summary is saved at:
```

```
/opt/VRTS/install/logs/uninstallsfracdate_time.summary
```

```
The uninstallsfrac log is saved at:
```

```
/opt/VRTS/install/logs/uninstallsfracdate_time.log
```



---

## Editing VCS Configuration Files on Existing Nodes

After running `uninstallsfrac`, modify the configuration files on the existing nodes to remove references to the removed nodes.

### Editing `/etc/llthosts`

On each of the existing nodes, use `vi` or another editor to edit `/etc/llthosts`. Remove lines corresponding to the removed nodes; for example, if `saturn` is the node removed from the cluster, remove the line "2 saturn" from the file:

```
0 galaxy
1 nebula
2 saturn
```

Change to:

```
0 galaxy
1 nebula
```

### Editing `/etc/gabtab`

In `/etc/gabtab`, change the command contained in the file to reflect the number of cluster nodes after the removal of a node:

```
/sbin/gabconfig -c -nN
```

$N$  is the number of remaining nodes. For example, a file with two remaining nodes appears as:

```
/sbin/gabconfig -c -n2
```



## Modifying the VCS Configuration to Remove the Node

Modify the VCS configuration by editing `/etc/VRTSvcs/conf/config/main.cf` directly, using the VCS GUI (Cluster Manager), or using the command line, as illustrated in the following example. Refer to the *VERITAS Cluster Server User's Guide* for details on configuring VCS.

Complete the procedure on one of the existing nodes.

1. As root user, enable write access to the configuration:

```
# haconf -makerw
```

2. Remove the node from the `AutoStartList` of the `oradb1_grp` Oracle service group by specifying the remaining nodes in the desired order:

```
# hagrps -modify oradb1_grp AutoStartList galaxy nebula
```

3. Remove the node from the `SystemList` of the `oradb1_grp` service group:

```
# hagrps -modify oradb1_grp SystemList -delete saturn
```

4. If you have other Oracle service groups with Oracle resources that have the removed node in their configuration, perform [step 2](#) and [step 3](#) for each of them.

5. Delete the node from the `CVMCluster` resource by removing it from the `CVMNodeId` attribute key list:

```
# hares -modify cvm_clus CVMNodeId -delete saturn
```

6. Delete the node from the `cvm AutoStartList` by specifying the remaining nodes in the desired order:

```
# hagrps -modify cvm AutoStartList galaxy nebula
```

7. Delete the node from the system lists for the `cvm` and `ClusterService` service groups:

```
# hagrps -modify cvm SystemList -delete saturn
# hagrps -modify ClusterService SystemList -delete saturn
```

8. After deleting the removed node from all service groups in the configuration, delete the node from the cluster system list:

```
# hasys -delete saturn
```

9. Save the new configuration to disk:

```
# haconf -dump -makero
```



---

## Removing Temporary Files

After running the `uninstallsfrac` utility, remove the following directories if they are still present on the node:

```
/var/tmp/uninstallsfrac_date  
/tmp/uninstallsfrac_date
```

## Removing the Infrastructure Packages (Optional)

The `installer` utility does not remove infrastructure packages used by VERITAS products other than VERITAS Storage Foundation *for Oracle RAC*. These packages include:

- ◆ `VRTSat`, VERITAS Authentication Service
- ◆ `VRTScpi`, VERITAS Cross Platform Installation Framework
- ◆ `VRTSob`, VERITAS Enterprise Administrator Service
- ◆ `VRTSobgui`, VERITAS Enterprise Administrator
- ◆ `VRTSperl`, VERITAS Perl 5.8.0 for VERITAS
- ◆ `VRTSvlic`, VERITAS License Utilities

If necessary, remove these packages using the `swremove` command.

## Removing Licenses

Navigate to the `/etc/vx/licenses/lic` directory and remove unwanted licenses. You can also rename or move them to another directory.







# Upgrading to SFRAC 4.1

---

10

If you are using VERITAS DBE/AC 3.5 for Oracle9i RAC in a cluster, you can upgrade the cluster to SFRAC 4.1. The upgrade process involves the following tasks:

- ✓ [“Uninstalling DBE/AC for Oracle9i RAC”](#) on page 191.
- ✓ [“Backing up and Removing Configuration Files”](#) on page 201.
- ✓ [“Upgrading the HP-UX Operating System”](#) on page 201.
- ✓ [“Installing and Configuring SFRAC 4.1”](#) on page 202.
- ✓ [“Installing License Key for Storage Mapping and Storage Checkpoints”](#) on page 202.
- ✓ [“Copying the VCS Configuration File”](#) on page 202.
- ✓ [“Editing the VCS Configuration File”](#) on page 202.
- ✓ [“Verifying and Copying the main.cf to Cluster Nodes”](#) on page 203.
- ✓ [“Starting VCS, CVM, and CFS on All Nodes”](#) on page 203.
- ✓ [“Upgrading the Disk Layout”](#) on page 203.
- ✓ [“Linking SFRAC Libraries with Oracle”](#) on page 204.

## Uninstalling DBE/AC for Oracle9i RAC

Uninstalling VERITAS DBE/AC 3.5 for Oracle9i RAC involves stopping `gsd` processes, unmounting CFS file systems outside of VCS control, taking Oracle and Sqlnet resources offline, unlinking the Oracle binary from VERITAS libraries, taking Oracle service groups offline, and running the `uninstallDBAC` script. You do not have to remove the Oracle database or uninstall Oracle9i to upgrade SFRAC. Refer to the VERITAS Database Edition / Advanced Cluster 3.5 for Oracle9i RAC documentation for more details on uninstalling the product.



## Stopping gsd and Applications Using CFS

1. As `oracle` user, stop `gsd` processes on each node. To determine if `gsd` processes are running, type:

```
$ $ORACLE_HOME/bin/psdctl stat
```

If `gsd` processes are running, stop them:

```
$ $ORACLE_HOME/bin/psdctl stop
```

2. Stop all applications using CFS mounts outside of VCS control. Ensure processes do not use the CFS mount point:

```
# fuser -c mount_point
```

To kill the processes using the CFS mount point, type:

```
# fuser -ck mount_point
```

## Unmounting CFS File Systems Outside of VCS Control

On each node, unmount any CFS file systems outside of VCS control.

1. Determine the file systems to unmount by checking the `/etc/mnttab` file. For example:

```
# cat /etc/mnttab | grep vxfs | grep cluster
```

The output shows each line of the `/etc/mnttab` file that contains an entry for a VxFS file system mounted in cluster mode.

2. To unmount each of the file systems listed in the output, specify the appropriate mount point:

```
# umount mount_point
```

## Taking Oracle and Sqlnet Resources Offline

1. Log in as `root` user to a system in the cluster.
2. On each node, take the Oracle and Sqlnet resources in the VCS configuration offline:

```
# hares -offline VRT -sys galaxy
# hares -offline rac -sys galaxy
# hares -offline VRT -sys nebula
# hares -offline rac -sys nebula
# hares -offline LISTENER -sys galaxy
# hares -offline LISTENER -sys nebula
```

These commands stop the Oracle instances running on the specified systems.

## Unlinking the Oracle Binary From VERITAS Libraries

If you uninstalled Oracle9i in your environment, proceed to [“Taking the Oracle Service Groups Offline”](#) on page 195.

1. Log in as the `oracle` user.
2. Using `vi` or another text editor, edit the `init$ORACLE_SID.ora` file:

```
$ cd $ORACLE_HOME/dbs/
$ vi init$ORACLE_SID.ora
```

Set the `cluster_database` parameter to `FALSE`:

```
cluster_database=FALSE
```

3. Navigate to the following directory:
 

```
$ cd $ORACLE_HOME/lib
```
4. Remove the file linked to the VERITAS ODM library.

For PA systems:

```
$ rm $ORACLE_HOME/lib/libodmd9.sl
```

For IA systems:

```
$ rm $ORACLE_HOME/lib/libodmd9.so
```



5. Copy the original ODM library file to remove the Veritas ODM function.

For PA systems:

```
$ cp $ORACLE_HOME/lib/libodmd9.sl.oracle
   $ORACLE_HOME/lib/libodmd9.sl
```

For IA systems:

```
$ cp $ORACLE_HOME/lib/libodmd9.so.oracle
   $ORACLE_HOME/lib/libodmd9.so
```

6. To convert from Oracle RAC binary to Oracle single-instance binary, enter:

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk rac_off
$ make -f ins_rdbms.mk ioracle
```

7. If \$ORACLE\_HOME is located on the file systems of the local nodes, perform [step 1](#) through [step 6](#) on each of the other nodes.
8. To verify the state of the Oracle RAC configuration, type:

```
$ ldd $ORACLE_HOME/bin/oracle | egrep "lltdb|vcsmm"
```

- ◆ Oracle single-instance binary is configured if the command does not generate output.
- ◆ Oracle RAC is enabled if the following output appears.

For PA systems:

```
/usr/lib/pa20_64/libvcsmm.1 => /usr/lib/pa20_64/libvcsmm.1
/usr/lib/pa20_64/liblltdb.1 => /usr/lib/pa20_64/liblltdb.1
```

For IA systems:

```
/usr/lib/hpux64/libvcsmm.1 => /usr/lib/hpux64/libvcsmm.1
/usr/lib/hpux64/liblltdb.so.1 => /usr/lib/hpux64/
                               liblltdb.so.1
```

## Taking the Oracle Service Groups Offline

1. Log in as `root` user on one cluster system.
2. Take the Oracle database service groups in the VCS configuration offline. For example:

```
# hagrps -offline oradb1_grp -sys galaxy
# hagrps -offline oradb2_grp -sys galaxy
# hagrps -offline oradb1_grp -sys nebula
# hagrps -offline oradb2_grp -sys nebula
```

These commands take the Oracle service groups, including all their configured resources, offline.

## Using the `uninstallDBAC` Script

The `uninstallDBAC` script removes packages installed by `installDBAC` on all systems in the cluster. This script stops VCS and takes all service groups in the VCS configuration offline on each node.

1. As `root` user, change to the directory containing the `uninstallDBAC` program.

```
# cd /opt/VRTSvcs/install
```

2. Start `uninstallDBAC`:

```
# ./uninstallDBAC
```

3. The script begins by referring you to the appropriate chapters in this document and describing the packages it is to remove:

```
Thank You for using the VERITAS Database Edition/Advanced
Cluster. Please ensure that you have read the uninstallation
chapter (to uninstall from the entire cluster) or the chapter
on removing DBE/AC cluster nodes (to uninstall from selected
nodes) in the DBE/AC Installation and Configuration Guide.
```

The following packages will be removed:

```
VRTScavf    VRTSdbac    VRTSgab    VRTSg1m    VRTSgms
VRTS11t    VRTSodm    VRTSperl   VRTSvcs    VRTSvcsag
VRTSvcsdc  VRTSvcsmg  VRTSvcsmn  VRTSvcsor
```

The uninstallation procedure is interactive and requires responses to queries.

```
Continue? [Y/N] :y
```



**Note** The following packages (and associated patches) used by DBE/AC are *not* removed by the `uninstallDBAC` script: `VRTSvxvm`, `VRTSvmpro`, `VRTSfspro`, `VRTSob`, `VRTSobgui`, and `VRTSvxfs`.

---

4. The script lists the systems in order from which it is to remove the packages, and creates files for the logging of the uninstallation activities:

The packages will be removed on following nodes in the order given below:

```
galaxy
nebula
```

5. Note that the installation actions are recorded in a log:

The output of the operations are temporarily being logged to `/var/tmp/uninstallDBAC_062403_091711/logfile`

6. The script verifies communications between the local system and the remote systems:

```
Checking communication with nebula
Pinging nebula
Checking remsh connectivity..... OK
Communication check completed successfully
```

7. The script stops VCS and unconfigures DBE/AC drivers on each system:

```
Stopping VCS on all systems..
VCS successfully stoppped on system galaxy
VCS successfully stoppped on system nebula

Unconfiguring DBE/AC drivers on galaxy
Unconfiguring CM driver ..... Done
Unconfiguring VXFEN driver .....Done
Unconfiguring ODM .....Done
Closing CFS Port ..... Done
Closing QLOG Port ..... Done
Unconfiguring DBE/AC drivers on nebula
Unconfiguring CM driver .....Done
Unconfiguring VXFEN driver .....Done
Unconfiguring ODM .....Done
Closing CFS Port ..... Done
Closing QLOG Port ..... Done
```

8. The script verifies whether the packages to be uninstalled are required or depended upon by other packages. It copies scripts and checks for packages on each node:

```
Checking galaxy
Checking nebula
```

9. The script removes VRTScavf, VRTSglm, VRTSgms, and VRTSodm packages:

```
Removing VRTScavf VRTSglm VRTSgms VRTSodm from galaxy
```

```
Removing VRTScavf ..... Done
Removing VRTSglm ..... Done
Removing VRTSgms ..... Done
Removing VRTSodm ..... Done
```

```
Removing VRTScavf VRTSglm VRTSgms VRTSodm from nebula
```

```
Removing VRTScavf ..... Done
Removing VRTSglm ..... Done
Removing VRTSgms ..... Done
Removing VRTSodm ..... Done
```

10. The script displays the cluster information and prompts you to indicate that you want VCS packages uninstalled:

```
VCS configuration files exist on this system with the following
information:
```

```
Cluster Name: vcs_racluster
Cluster ID Number: 7
Systems: galaxy nebula
Service Groups: cvm oradb1_grp oradb2_grp
```

```
Do you want to uninstall VCS from all of the systems in this
cluster? (Y) y
```

11. The script checks for the operating system level and for system-to-system communications:

```
Checking OS version on galaxy ..... HP-UX B.11.11
Verifying communication with nebula ..... ping successful
Attempting remsh with nebula ..... remsh successful
Checking OS version on nebula ..... HP-UX B.11.11
Creating /tmp subdirectory on nebula../tmp subdirectory created
```

```
Using /usr/bin/remsh and /usr/bin/rcp to communicate with
nebula
```

```
Communication check completed successfully
```



**12.** The script checks currently installed DBE/AC packages on each system:

Checking current installation on galaxy:

```
Checking product mode ..... DBE/AC licensed
Checking VRTSdbac ..... version 3.5 installed
Checking VRTSdbac dependencies ..... none
Checking VRTSvcsor ..... version 3.5 installed
Checking VRTSvcsor dependencies ..... none
Checking VRTSvcsmn ..... version 3.5 installed
```

.

.

Checking current installation on nebula:

```
Checking VRTSdbac ..... version 3.5 installed
Checking VRTSdbac dependencies ..... none
Checking VRTSvcsor ..... version 3.5 installed
Checking VRTSvcsor dependencies ..... none
Checking VRTSvcsmn ..... version 3.5 installed
Checking VRTSvcsmn dependencies ..... none
```

.

.

.

Uninstallation check completed successfully

**13.** The DBE/AC processes are stopped on each node:

Stopping DBE/AC processes on galaxy:

```
Checking DBE/AC ..... not running
Checking CmdServer ..... running
Killing CmdServer ..... Done
Checking Web GUI processes ..... not running
Checking notifier processes ..... not running
Checking VCSMM ..... not running
Checking VXFEN ..... not running
Checking LMX ..... not running
Checking GAB ..... GAB module loaded
Stopping GAB ..... Done
Unloading GAB module on galaxy .....Done
Checking LLT ..... LLT module loaded
Stopping LLT ..... Done
Unloading LLT module on galaxy ..... Done
```

Stopping DBE/AC processes on nebula:





```

Checking DBE/AC ..... not running
Checking CmdServer ..... running
Killing CmdServer ..... Done
Checking Web GUI processes ..... not running
Checking notifier processes ..... not running
Checking VCSMM ..... not running
Checking VXFEN ..... not running
Checking LMX ..... not running
Checking GAB ..... GAB module loaded
Stopping GAB ..... Done
Unloading GAB module on nebula ..... Done
Checking LLT ..... LLT module loaded
Stopping LLT ..... Done
Unloading LLT module on nebula ..... Done

```

#### 14. VCS and DBE/AC related packages are removed from each system:

Uninstalling DBE/AC on galaxy:

```

Removing VRTSdbac package ..... Done
Removing VRTSvcsor package ..... Done
Removing VRTSvcsmn package ..... Done
Removing VRTSvcsdc package ..... Done
Removing VRTSvcsag package ..... Done
Removing VRTSvcsmg package ..... Done
Removing VRTSvcs package ..... Done
Removing VRTSgab package ..... Done
Removing VRTSllt package ..... Done
Removing VRTSperl package ..... Done

```

Uninstalling DBE/AC on nebula:

```

Removing VRTSdbac package ..... Done
Removing VRTSvcsor package ..... Done
Removing VRTSvcsmn package ..... Done
Removing VRTSvcsdc package ..... Done
Removing VRTSvcsag package ..... Done
Removing VRTSvcsmg package ..... Done
Removing VRTSvcs package ..... Done
Removing VRTSgab package ..... Done
Removing VRTSllt package ..... Done
Removing VRTSperl package ..... Done

```

DBE/AC configuration files are backed up with a  
.prev.vcs\_racluster extension

A DBE/AC uninstall report has been saved at:



/var/VRTSvcs/uninstallvcsReport

DBE/AC uninstallation completed successfully

End of DBE/AC Uninstallation: Tue Jun 24 09:32:04 PDT 2003

IMPORTANT: Check the following log files for any errors/warnings:

/var/tmp/uninstallDBAC\_062403\_091711/logfile

/var/VRTSvcs/uninstallvcsReport\*

/var/adm/sw/swremove.log (on all cluster nodes)



## Backing up and Removing Configuration Files

Complete the following steps for the SFRAC configuration files; you do not need to remove temporary files or 3.5 license files.

1. Note the cluster ID from the `llttab` file. For example:

```
# cat /etc/llttab
set-node slpas19
set-cluster 192
link lan1 /dev/lan:1 - ether - -
link lan2 /dev/lan:2 - ether - -
```

This `llttab` file has a cluster ID of 192. VERITAS recommends reusing the existing cluster id.

2. Back up the 3.5 `/etc/VRTSvcs/conf/config/main.cf`. For example:

```
# cd /etc/VRTSvcs/conf/config/
# cp main.cf main.cf_3.5
```

3. Remove additional configuration files. For example:

```
# rm CFSTypes.cf OracleTypes.cf types.cf CVMTypes.cf
# rm VRTSWebAppType.cf master.main.cf
```

4. Remove the remaining SFRAC configuration files:

```
# rm /etc/llthosts
# rm /etc/llttab
# rm /etc/gabtab
# rm /etc/vcsmmtab
# rm /etc/vxfendg
# rm /etc/vxfentab
```

5. Remove any VxVM or VxFS patches using the `swremove` command.

## Upgrading the HP-UX Operating System

Refer to HP-UX upgrade documentation for instructions on upgrading the operating system from the 11.11 to 11.23 PI LR version.



## Installing and Configuring SFRAC 4.1

Complete the procedures in [Chapter 3. “Installing SFRAC 4.1 and Configuring Components”](#) on page 27 to set up SFRAC 4.1.

## Installing License Key for Storage Mapping and Storage Checkpoints

1. Start the license utility:

```
# vxlicinst
```

2. Enter the license key. The utility confirms the key installation for VERITAS Mapping Services.

## Copying the VCS Configuration File

1. Save the existing configuration:

```
# haconf -dump -makero
```

2. Stop VCS on all nodes:

```
# hastop -all
```

3. Copy the backup of the 3.5 `main.cf` to the current `main.cf`. For example:

```
# cd /etc/VRTSvcs/conf/config/  
# cp main.cf_3.5 main.cf
```

## Editing the VCS Configuration File

Edit the `main.cf` for the SFRAC 4.1 and VCS resource hierarchy. The hierarchy is slightly different for SFRAC 4.1. Remove the `qllogckd` resource from the `main.cf` file and refer the sample `main.cf` in [Appendix A. “Sample Configuration Files”](#) on page 229 to modify the resource hierarchy. Make sure to change the `Sqlnet` resource to `Netlsnr`.



## Verifying and Copying the main.cf to Cluster Nodes

1. Save and close the edited main.cf file after completing the required changes.
2. Verify the syntax of the /etc/VRTSvcs/conf/config/main.cf file:
3. Using rcp or another utility, copy the VCS configuration file from a node (for example, slpas06) to the remaining cluster nodes. On each remaining node, type:

```
# hacf -verify /etc/VRTSvcs/conf/config
```

```
# rcp slpas06:/etc/VRTSvcs/conf/config/main.cf
      /etc/VRTSvcs/conf/config
```

## Starting VCS, CVM, and CFS on All Nodes

With the configuration file in place on each node, start VCS, CVM, and CFS:

```
# hstart
```

Make sure to run this command from each node.

## Upgrading the Disk Layout

VERITAS DBE/AC 3.5 for Oracle9i RAC used an older disk layout version. The following message appears while mounting the shared volume with CFS 3.5:

```
UX:vxfs mount: WARNING: V-3-24304: Disk layout versions older than
Version 6 will not be supported in the next release for cluster
mount. It is advisable to upgrade to disk layout Version 6 now. See
vxupgrade(1M) for information on upgrading a VxFS file system and
see the VxFS Release Notes for information on disk layout support.
```

The following commands shows the upgrade process from a version 4 disk layout (for VERITAS DBE/AC 3.5) to a version 6 disk layout (for SFRAC 4.1):

```
# mount -F vxfs -o cluster /dev/vx/dsk/oracledg/srvm_vol /orasrv
UX:vxfs mount: WARNING: V-3-24304: Disk layout versions older than
Version 6 will not be supported in the next release for cluster
mount. It is advisable to upgrade to disk layout Version 6 now. See
vxupgrade(1M) for information on upgrading a VxFS file system and
see the VxFS Release Notes for information on disk layout support.
```

```
# vxupgrade -n 5 /orasrv
# vxupgrade -n 6 /orasrv
```



## Linking SFRAC Libraries with Oracle

Refer to “[Linking to VERITAS Libraries](#)” on page 81.



# Uninstalling SFRAC 4.1

11

This chapter describes how to remove the Storage Foundation *for Oracle RAC* 4.1 software from the systems in a cluster.

## Stopping gsd

As `oracle` user, stop the `gsd` processes running on all nodes in the SFRAC cluster:

- ◆ To determine if `gsd` processes are running, type:

```
$ $ORACLE_HOME/bin/gsdctl stat
```

- ◆ To stop the processes, type:

```
$ $ORACLE_HOME/bin/gsdctl stop
```

## Stopping Applications Using CFS (Outside of VCS Control)

Stop all applications using the CFS mounts *not* under VCS control.

- ◆ To ensure that no processes are using the CFS mount point, type:

```
# fuser -c mount_point
```

- ◆ To stop any processes using a CFS mount point, type:

```
# fuser -ck mount_point
```

## Unmounting CFS File Systems (Outside of VCS Control)

Unmount any CFS file systems *not* under VCS control on all nodes:

1. Determine the file systems to unmount by checking the `/etc/mnttab` file. For example:

```
# cat /etc/mnttab | grep vxfs | grep cluster
```



The output shows each line of the `/etc/mnttab` file that contains an entry for a VxFS file system mounted in the cluster mode.

2. By specifying its mount point, unmount each file system listed in the output:

```
# umount mount_point
```

## Taking the Oracle and Netlsnr Resources Offline

1. Log in as `root` user.
2. On each node, take the Oracle and Netlsnr resources in the VCS configuration file (`main.cf`) offline. For example:

```
# hares -offline VRT -sys slpas06  
# hares -offline rac -sys slpas06  
# hares -offline VRT -sys slpas07  
# hares -offline rac -sys slpas07  
# hares -offline LISTENER -sys slpas06  
# hares -offline LISTENER -sys slpas07
```

These commands stop the Oracle instances running on the specified nodes.

## Removing the Oracle Database (Optional)

You can remove the Oracle database after safely relocating the data as necessary.

## Uninstalling Oracle9i (Optional)

If you choose not to uninstall Oracle, proceed to [“Unlinking the Oracle Binary From VERITAS Libraries”](#) on page 207.

If you choose to uninstall Oracle9i, use the Oracle `runInstaller` utility. Run the utility on each node if Oracle9i is installed locally. If Oracle9i is installed on a cluster file system, run the utility once.

1. Log in as `oracle` user.
2. Set the `DISPLAY` variable.

If you use the Bourne Shell (`sh` or `ksh`):

```
$ DISPLAY=host:0.0;export DISPLAY
```





If you use the C Shell (`csh` or `tcsh`):

```
$ setenv DISPLAY host:0.0
```

3. Run the Oracle9i `runInstaller` utility:

```
$ /cdrom/Disk1/runInstaller
```

Select the option to uninstall the Oracle9i software. Refer to the Oracle9i documentation for details on this utility.

4. If necessary, remove Oracle9i from the other nodes.

## Unlinking the Oracle Binary From VERITAS Libraries

If you uninstalled Oracle9i, proceed to “[Stopping VCS](#)” on page 208. Otherwise, unlink the VERITAS libraries:

1. Log in as the `oracle` user.
2. Using `vi` or another text editor, edit the `init$ORACLE_SID.ora` file:

```
$ cd $ORACLE_HOME/dbs/
$ vi init$ORACLE_SID.ora
```

Set the `cluster_database` parameter to `FALSE`:

```
cluster_database=FALSE
```

3. Navigate to the following directory:
4. Remove the file linked to the VERITAS ODM library.

```
$ cd $ORACLE_HOME/lib
```

For PA systems:

```
$ rm $ORACLE_HOME/lib/libodmd9.s1
```

For IA systems:

```
$ rm $ORACLE_HOME/lib/libodmd9.so
```

5. Copy the original ODM library file to remove the Veritas ODM function.

For PA systems:

```
$ cp $ORACLE_HOME/lib/libodmd9.s1.oracle
$ORACLE_HOME/lib/libodmd9.s1
```

For IA systems:



```
$ cp $ORACLE_HOME/lib/libodmd9.so.oracle
   $ORACLE_HOME/lib/libodmd9.so
```

6. To convert from Oracle RAC binary to Oracle single-instance binary, enter:

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -f ins_rdbms.mk rac_off
$ make -f ins_rdbms.mk ioracle
```

7. If \$ORACLE\_HOME is located on the file systems of the local nodes, perform [step 1](#) through [step 6](#) on each of the other nodes.

8. To verify the state of the Oracle RAC configuration, type:

```
$ ldd $ORACLE_HOME/bin/oracle | egrep "lltdb|vcsmm"
```

- ◆ Oracle single-instance binary is configured if the command does not generate output.
- ◆ Oracle RAC is enabled if the following output appears.

For PA systems:

```
/usr/lib/pa20_64/libvcsmm.1 => /usr/lib/pa20_64/libvcsmm.1
/usr/lib/pa20_64/liblltdb.1 => /usr/lib/pa20_64/liblltdb.1
```

For IA systems:

```
/usr/lib/hpux64/libvcsmm.1 => /usr/lib/hpux64/libvcsmm.1
/usr/lib/hpux64/liblltdb.so.1 => /usr/lib/hpux64/
                               liblltdb.so.1
```

## Stopping VCS

Stop VCS to take the service groups on all nodes offline:

1. Log in as root user on one cluster node.
2. Stop VCS on all nodes.

```
# hastop -all
```



## Running the installer or uninstallsfrac Script

The VERITAS product installer or `uninstallsfrac` script removes SFRAC 4.1 packages on all nodes in the cluster. The installer offers a high-level approach to uninstalling multiple VERITAS products. The `uninstallsfrac` script offers a direct approach to uninstalling SFRAC.

### Using the VERITAS Product Installer to Uninstall SFRAC

Review this section to uninstall SFRAC using the VERITAS product installer; you can also uninstall SFRAC using the `uninstallsfrac` utility as described on [page 209](#).

1. Navigate to the directory containing the installer program:

```
# # cd /cdrom
```

2. Start the product installer:

```
# ./installer
```

3. Enter "u" to select the "Uninstall a Product" option.
4. Select "VERITAS Storage Foundation for Oracle RAC" in the product list.
5. Proceed to [step 3](#) of "Using the `uninstallsfrac` Script to Uninstall SFRAC" on [page 209](#).

### Using the `uninstallsfrac` Script to Uninstall SFRAC

Review this section to install SFRAC using the `uninstallsfrac` utility; you can also uninstall SFRAC using the VERITAS product installer as described on [page 209](#).

1. Navigate to the folder containing the script:

```
# cd /cdrom/storage_foundation_for_oracle_rac
```

2. Start the script:

```
# ./uninstallsfrac
```

3. Indicate whether or not you want to remove VERITAS VM and VVR packages from all specified systems; enter "y" only if the root disk is outside of VM control.
4. If you invoke the `uninstallsfrac` script from a cluster node, review the cluster information and confirm to uninstall SFRAC:



VCS configuration files exist on this system with the following information:

```
Cluster Name: racclus124
Cluster ID Number: 120
Systems: hpslia20 hpslia21 hpslia22 hpslia23
Service Groups: cvm
```

Do you want to uninstall SFRAC from these systems? [y,n,q] (y)

If you invoke the uninstallsfrac script from a remote machine in the same subnet, enter the systems where the uninstallation will take place:

Enter the system names separated by spaces on which to uninstall SFRAC: **slpas06 slpas07 slpas08 slpas09**

5. Review the output as the script checks the operating system on each system, verifies system-to-system communication, and sets up the log file noted below:

Checking system communication:

```
Verifying communication with slpas06 ..... ping successful
Attempting remsh with slpas06 ..... remsh successful
Attempting rcp with slpas06 ..... rcp successful
Checking OS version on slpas06 ..... HP-UX B.11.23
Creating log directory on slpas06 ..... Done
.
.
```

Logs for uninstallsfrac are being created in /var/tmp/uninstallsfrac128201025.

Using /usr/bin/remsh and /usr/bin/rcp to communicate with remote systems.

Initial system check completed successfully.

The script checks for Storage Foundation *for Oracle RAC* packages currently installed on the nodes. This process involves identifying system uninstall requirements and dependencies between packages to determine the safety and order of uninstalling packages.

Checking SFRAC packages installed on slpas06:

```
Checking VRTSvxmsa package ..... version 4.1 installed
Checking VRTSvxmsa dependencies ..... no dependencies
Checking VRTScsocw package ..... version 4.1 installed
Checking VRTScsocw dependencies ..... no dependencies
Checking VRTSvcsor package ..... version 4.1 installed
Checking VRTSvcsor dependencies ..... no dependencies
```



```

.
.
Checking VRTSalloc package ..... version 4.1 installed
Checking VRTSalloc dependencies ..... no dependencies
Checking VRTSvxvm package ..... version 4.1 installed
Checking VRTSvxvm dependencies ..... no dependencies
.

```

**6. Confirm to uninstall SFRAC:**

```

installer is now ready to uninstall SFRAC packages.
All SFRAC processes that are currently running will be stopped.

Are you sure you want to uninstall SFRAC packages? [y,n,q] (y)

```

**7. Review the output as the script stops processes and drivers running on each node, and reports its activities:**

```

Stopping Cluster Manager and Agents on all the nodes .... Done
Checking the boot disk on slpas06 ..... not VxVM-controlled
Checking slpas06 for open volumes ..... None
Stopping SFRAC processes on slpas06:

Checking odm driver ..... odm module loaded
Stopping odm driver ..... Done
Unloading odm module on slpas06 ..... Done
Checking vxgms driver ..... vxgms module loaded
Unloading vxgms module on slpas06 ..... Done
.

```

**8. Review the output as the script indicates the progress of removing packages from the nodes by listing the steps that are completed. The total number of steps depends on the nature of the installation. For example:**

```

Uninstalling Storage Foundation for Oracle RAC 4.1 on all systems
simultaneously:

Uninstalling VRTSvxmsa 4.1 on slpas09 ..... Done 1 of 144 steps
Uninstalling VRTSvxmsa 4.1 on slpas07 ..... Done 2 of 144 steps
Uninstalling VRTSvxmsa 4.1 on slpas06 ..... Done 3 of 144 steps
Uninstalling VRTSvxmsa 4.1 on slpas08 ..... Done 4 of 144 steps
Uninstalling VRTSscocw 4.1 on slpas07 ..... Done 5 of 144 steps
.
.
Uninstalling VRTSvxvm 4.1 on slpas06 .... Done 141 of 144 steps
Uninstalling VRTSvxvm 4.1 on slpas09 .... Done 142 of 144 steps
Uninstalling VRTSvxvm 4.1 on slpas08 .... Done 143 of 144 steps
Uninstalling VRTSvxvm 4.1 on slpas07 .... Done 144 of 144 steps

```



Storage Foundation for Oracle RAC package uninstall completed successfully.

**9.** If necessary, review the summary and log files of uninstallation activities.

Uninstallation of Storage Foundation for Oracle RAC has completed successfully.

The uninstallation summary is saved at:

```
/opt/VRTS/install/logs/uninstallsfracdate_time.summary
```

The uninstallsfrac log is saved at:

```
/opt/VRTS/install/logs/uninstallsfracdate_time.log
```

**10.** Reboot the nodes to completely uninstall SFRAC:

```
# ./usr/sbin/shutdown -r now
```

## Removing VERITAS Infrastructure Packages (Optional)

The `installer` utility does not remove infrastructure packages used by VERITAS products other than VERITAS Storage Foundation *for Oracle RAC*. These packages include:

- ◆ VRTSat, VERITAS Authentication Service
- ◆ VRTScpi, VERITAS Cross Platform Installation Framework
- ◆ VRTSob, VERITAS Enterprise Administrator Service
- ◆ VRTSobgui, VERITAS Enterprise Administrator
- ◆ VRTSperl, VERITAS Perl 5.8.0 for VERITAS
- ◆ VRTSvlic, VERITAS License Utilities

If necessary, remove these packages using the `swremove` command.



---

## Removing License Files

1. Navigate to the directory containing the license key files and list them. Enter:

```
# cd /etc/vx/licenses/lic  
# ls -a
```

2. Identify and delete unwanted key files.

## Removing Other Configuration Files (Optional)

You can remove the following configuration files:

```
/etc/vcsmmtab  
/etc/vxfentab  
/etc/vxfendg  
/etc/llttab  
/etc/gabtab  
/etc/llthosts
```







This chapter presents various problem scenarios and solutions.

## Running Scripts for Engineering Support Analysis

You can use a set of three scripts that gather information about the configuration and status of your cluster and its various modules. The scripts also identify package information, debugging messages, console messages, and information about disk groups and volumes. You can forward the output of each of these scripts to VERITAS customer support who can analyze the information and assist you in solving any problems.

### getdbac

This script gathers information about the VERITAS Storage Foundation *for Oracle RAC* modules. The file `/tmp/vcsopslog.time_stamp.sys_name.tar.z` contains the script's output. Enter the following command on *each* node:

```
# /opt/VRTSvcs/bin/getdbac -local
```

### getcomms

This script gathers information about the GAB and LLT modules. The file `/tmp/commslog.time_stamp.tar` contains the script's output. On *each* node, enter:

```
# /opt/VRTSgab/getcomms -local
```

### hagetcf

This script gathers information about the VCS cluster and the status of resources. The script's output is placed in a tar file, `/tmp/vcsconf.sys_name.tar`, on each node. To run this script, enter the following command on *each* node:

```
# /opt/VRTSvcs/bin/hagetcf
```



## Troubleshooting Topics

The following troubleshooting topics have headings that indicate likely symptoms or that indicate procedures required for a solution.

### Oracle User Must be Able to Read /etc/littab File

Check the permissions of the `/etc/littab` file; the `oracle` user must be allowed to read it.

### Missing Dialog Box During Installation of Oracle9i Release 2

During installation of Oracle9i Release 2 using the `runInstaller` utility, if you choose the Enterprise Edition or Custom Install (with RAC option), a dialog box prompting you about the installation nodes should appear. If the dialog box fails to appear, exit the `runInstaller` utility and complete the following steps:

1. On the cluster node, check for the VERITAS CM library.

For PA clusters:

- ◆ The 32-bit `libnmap2.sl` is located in `/opt/nmap2/lib/`.
- ◆ The 64-bit `libnmap2.sl` is located in `/opt/nmap2/lib/pa20_64`.

For IA clusters:

- ◆ The 32-bit `libnmap2.so` is located in `/opt/nmap2/lib/hpux32/lib/`.
- ◆ The 64-bit `libnmap2.so` is located in `/opt/nmap2/lib/hpux64/lib/`.

2. Start the VCSMM driver on both the nodes by entering:

```
# /sbin/vcsmmconfig -c
```



## Error When Starting an Oracle Instance

If the VCSMM driver (the membership module) is not configured, an error is displayed on starting the Oracle instance that resembles:

```
ORA-29702: error occurred in Cluster Group Operation
```

To start the driver, enter the following command:

```
# /sbin/vcsmmconfig -c
```

The command included in the `/etc/vcsmmtab` file enables the VCSMM driver to be started at system boot.

## Instance Numbers Must be Unique (Error Code 205)

If you encounter error code 205 when the `skgxnreg` function fails (look in the Oracle trace files to find the error returned), make sure there is a unique instance number specified in the `$ORACLE_HOME/dbs/init${ORACLE_SID}.ora` file on each node.

## ORACLE\_SID Must be Unique (Error Code 304)

If you encounter error code 304 when the `skgxnreg` function fails (look in the Oracle trace file to find the error returned), make sure that the `ORACLE_SID` environment variable specified during Oracle startup is unique on each node in your cluster. Also, make sure that the `SID` attribute for the Oracle resource in the `main.cf` is specified locally and is unique.

## Oracle Log Files Show Shutdown Called Even When Not Shutdown Manually

The Oracle enterprise agent calls shutdown if monitoring of the Oracle/Netlsnr resources fails for some reason. On all cluster nodes, look at the following VCS and Oracle agent log files for any errors or status:

```
/var/VRTSvcs/log/engine_A.log  
/var/VRTSvcs/log/Oracle_A.log
```

## Set MLOCK Privilege for DBA User

If `ASYNCH_IO` errors occur during select and update queries on the Oracle database, the workaround involves setting the `MLOCK` privilege for the `dba` user:



1. Give the MLOCK privilege to the dba group:

```
# setprivgrp dba MLOCK
```

2. Create the /etc/privgroup file and add the line:

```
dba MLOCK
```

3. Verify the availability of MLOCK privilege for the dba group:

```
# /usr/bin/getprivgrp dba
```

## File System Configured Incorrectly for ODM Shuts Down Oracle

Linking Oracle9i with the VERITAS ODM libraries provides the best file system performance. See [“Linking to VERITAS Libraries”](#) on page 81 for instructions on creating the link and confirming that Oracle uses the libraries. Shared file systems in RAC clusters without ODM Libraries linked to Oracle9i may exhibit slow performance and are *not* supported.

If ODM cannot find the resources it needs to provide support for cluster file systems, it does not allow Oracle to identify cluster files and causes Oracle to fail at startup. Run the following command:

```
# cat /dev/odm/cluster
cluster status: enabled
```

If the status is “enabled,” ODM is supporting cluster files. Any other cluster status indicates that ODM is not supporting cluster files. Other possible values include:

pending	ODM cannot yet communicate with its peers, but anticipates being able to eventually.
failed	ODM cluster support has failed to initialize properly. Check console logs.
disabled	ODM is not supporting cluster files. If you think it should, check: <ul style="list-style-type: none"> <li>• /dev/odm mount options in /etc/vfstab. If the “nocluster” option is being used, it can force the “disabled” cluster support state.</li> <li>• Make sure the VRTSgms (group messaging service) package is installed. Run the following command: <pre># /opt/VRTSvcS/bin/chk_sfrac_pkgs</pre> The utility reports any missing required packages.</li> </ul>

If /dev/odm is not mounted, no status can be reported.



## VCSIPC Wait Warning Messages in Oracle Trace/Log Files

When Gigabit Ethernet interconnections are used, a high load can cause LMX/LLT to flow-control VCSIPC, resulting in warning messages to be reported in the Oracle trace file. The default location for the trace file is `$ORACLE_HOME/rdbms/log`; it may have changed if the parameters `background_dump_dest` or `user_dump_dest` have been changed. The messages resemble:

```
.
Unix process pid; 9560, image: oracle@MCB4800 (LMS0)
*** 2003-03-22 10:18:46.370
*** SESSION ID: (5.1) 2003-03-22 10:18:44.387
VCSIPC wait: WARNING: excessive poll done, 1001 times
VCSIPC wait: WARNING: excessive poll done, 1001 times
VCSIPC wait: WARNING: excessive poll done, 1002 times
VCSIPC wait: WARNING: excessive poll done, 1003 times
VCSIPC wait: WARNING: excessive poll done, 1004 times
VCSIPC wait: WARNING: excessive poll done, 1005 times
.
```

As a workaround, you can change the LLT lowwater mark, highwater mark, and window values for flow control. Please contact VERITAS support for more information about changing these values.

## VCSIPC Errors in Oracle Trace/Log Files

If you see any VCSIPC errors in the Oracle trace/log files, check `/var/adm/messages` for any LMX error messages. If you see messages that contain any of the following:

```
. . . out of buffers
. . . out of ports
. . . no minors available
```

Refer to [“Tunable Kernel Driver Parameters”](#) on page 253.

If you see any VCSIPC warning messages in Oracle trace/log files that resemble:

```
connection invalid
```

or,

```
Reporting communication error with node
```

check whether the Oracle Real Application Cluster instance on the other node is still running or has been restarted. The warning message indicates that the VCSIPC/LMX connection is no longer valid.



## Shared Disk Group Cannot be Imported

If you see a message resembling:

```
vxvm:vxconfigd:ERROR:vold_pgr_register(/dev/vx/rdmp/disk_name):
local_node_id<0
Please make sure that CVM and vxfen are configured and operating
correctly
```

This message is displayed when CVM cannot retrieve the node ID of the local system from the vxfen driver. This usually happens when port b is not configured. Verify that the vxfen driver is configured by checking the GAB ports with the command:

```
# /sbin/gabconfig -a
```

Port b must exist on the local system.

## CVMVolDg Does Not Go Online Even Though CVMCluster is Online

When the CVMCluster resource goes online, the shared disk groups are automatically imported. If the disk group import fails for some reason, the CVMVolDg resources fault. Clearing and offlining the CVMVolDg type resources does not fix the problem.

Workaround:

1. Fix the problem causing the import of the shared disk group to fail.
2. Offline the service group containing the resource of type CVMVolDg as well as the service group containing the CVMCluster resource type.
3. Bring the service group containing the CVMCluster resource online.
4. Bring the service group containing the CVMVolDg resource online.

## Restoring Communication Between Host and Disks After Cable Disconnection

If a fiber cable is inadvertently disconnected between the host and a disk, you can restore communication between the host and the disk without restarting by doing the following:

1. Reconnect the cable.
2. Use the `format` command to verify that the host sees the disks. It may take a few minutes before the host is capable of seeing the disk.

- Issue the following `vxctl` command to force the VxVM configuration daemon `vxconfigd` to rescan the disks:

```
# vxctl enable
```

## Node is Unable to Join Cluster While Another Node is Being Ejected

A cluster that is currently fencing out (ejecting) a node from the cluster prevents a new node from joining the cluster until the fencing operation is completed. The following are example messages that appear on the console for the new node:

```
...VCS FEN ERROR V-11-1-25 ... Unable to join running cluster
...VCS FEN ERROR V-11-1-25 ... since cluster is currently fencing
...VCS FEN ERROR V-11-1-25 ... a node out of the cluster.

...VCS GAB.. Port b closed
```

If you see these messages when the new node is booting, the startup script (`/etc/vxfen-startup`) on the node makes up to five attempts to join the cluster. If this is not sufficient to allow the node to join the cluster, restart the new node or attempt to restart `vxfs` driver with the command:

```
# /sbin/init.d/vxfen start
```

## vxfssthdw Fails When SCSI TEST UNIT READY Command Fails

If you see a message resembling:

```
Issuing SCSI TEST UNIT READY to disk reserved by other node FAILED.
Contact the storage provider to have the hardware configuration
fixed.
```

The disk array does not support returning success for a SCSI TEST UNIT READY command when another host has the disk reserved using SCSI-3 persistent reservations. This happens with Hitachi Data Systems 99XX arrays if bit 186 of the system mode option is not enabled.



## Removing Existing Keys From Disks

To remove the registration and reservation keys created by another node from a disk:

1. Create a file to contain the access names of the disks:

```
# vi /tmp/disklist
```

For example:

```
/dev/rdisk/c1t0d11s2
```

2. Read the existing keys:

```
# vxfenadm -g all -f /tmp/disklist
```

The output from this command displays the key:

```
Device Name: /dev/rdisk/c1t0d11s2
Total Number Of Keys: 1
key[0]:
  Key Value [Numeric Format]: 65,49,45,45,45,45,45,45
  Key Value [Character Format]: A1-----
```

3. If you know on which node the key was created, log in to that node and enter the following command:

```
# vxfenadm -x -kA1 -f /tmp/disklist
```

The key is removed.

4. If you do not know on which node the key was created, follow [step 5](#) through [step 7](#) to remove the key.

5. Register a second key “A2” temporarily with the disk:

```
# vxfenadm -m -kA2 -f /tmp/disklist
Registration completed for disk path /dev/rdisk/c1t0d11s2
```

6. Remove the first key from the disk by preempting it with the second key:

```
# vxfenadm -p -kA2 -f /tmp/disklist -vA1
key: A2----- preempted the key: A1----- on disk
/dev/rdisk/c1t0d11s2
```

7. Remove the temporary key assigned in [step 5](#).

```
# vxfenadm -x -kA2 -f /tmp/disklist
Deleted the key : [A2-----] from device /dev/rdisk/c1t0d11s2
```

No registration keys exist for the disk.





## System Panics to Prevent Potential Data Corruption

When a node experiences a split brain condition and is ejected from the cluster, it panics and displays the following console message:

```
VXFEN:vxfen_plat_panic: Local cluster node ejected from cluster to
prevent potential data corruption.
```

### How vxfen Driver Checks for Pre-existing Split Brain Condition

The vxfen driver functions to prevent an ejected node from rejoining the cluster after the failure of the private network links and before the private network links are repaired.

For example, suppose the cluster of system 1 and system 2 is functioning normally when the private network links are broken. Also suppose system 1 is the ejected system. When system 1 restarts before the private network links are restored, its membership configuration does not show system 2; however, when it attempts to register with the coordinator disks, it discovers system 2 is registered with them. Given this conflicting information about system 2, system 1 does not join the cluster and returns an error from vxfenconfig that resembles:

```
vxfenconfig: ERROR: There exists the potential for a preexisting
split-brain. The coordinator disks list no nodes which are in the
current membership. However, they also list nodes which are not
in the current membership.
```

```
I/O Fencing Disabled!
```

Also, the following information is displayed on the console:

```
<date> <system name> vxfen: WARNING: Potentially a preexisting
<date> <system name> split-brain.
<date> <system name> Dropping out of cluster.
<date> <system name> Refer to user documentation for steps
<date> <system name> required to clear preexisting split-brain.
<date> <system name>
<date> <system name> I/O Fencing DISABLED!
<date> <system name>
<date> <system name> gab: GAB:20032: Port b closed
```

However, the same error can occur when the private network links are working and both systems go down, system 1 restarts, and system 2 fails to come back up. From the view of the cluster from system 1, system 2 may still have the registrations on the coordinator disks.



### **Case 1: System 2 Up, System 1 Ejected (Actual Potential Split Brain)**

Determine if system1 is up or not. If it is up and running, shut it down and repair the private network links to remove the split brain condition. restart system 1.

### **Case 2: System 2 Down, System 1 Ejected (Apparent Potential Split Brain)**

1. Physically verify that system 2 is down.
2. Verify the systems currently registered with the coordinator disks. Use the following command:

```
# vxfenadm -g all -f /etc/vxfentab
```

The output of this command identifies the keys registered with the coordinator disks.

3. Clear the keys on the coordinator disks as well as the data disks using the command `/opt/VRTSvcs/rac/bin/vxfenclearpre`. See [“Using vxfenclearpre Command to Clear Keys After Split Brain”](#) on page 225.
4. Make any necessary repairs to system 2 and restart.

## Using vxfenclearpre Command to Clear Keys After Split Brain

When you have encountered a split brain condition, use the `vxfenclearpre` command to remove SCSI-3 registrations and reservations on the coordinator disks as well as on the data disks in all shared disk groups.

1. Shut down all other nodes in the cluster that have access to the shared storage. This prevents data corruption.
2. Start the script:
3. Read the script's introduction and warning. Then, you can choose to let the script run.

```
Do you still want to continue: [y/n] (default : n)
y
```

**Note** Informational messages resembling the following may appear on the console of one of the nodes in the cluster when a node is ejected from a disk/LUN:

```
<date> <system name> scsi: WARNING: /sbus@3,0/lpfs@0,0/sd@0,1(sd91):
<date> <system name> Error for Command: <undecoded cmd 0x5f> Error Level:
Informational
<date> <system name> scsi: Requested Block: 0 Error Block 0
<date> <system name> scsi: Vendor: <vendor> Serial Number: 0400759B006E
<date> <system name> scsi: Sense Key: Unit Attention
<date> <system name> scsi: ASC: 0x2a (<vendor unique code 0x2a>), ASCQ: 0x4,
FRU: 0x0
```

These informational messages may be ignored.

```
Cleaning up the coordinator disks...
```

```
Cleaning up the data disks for all shared disk groups...
```

```
Successfully removed SCSI-3 persistent registration and
reservations from the coordinator disks as well as the shared
data disks.
```

```
Reboot the server to proceed with normal cluster startup...
#
```

4. Restart all nodes in the cluster.



## Removing or Adding Coordinator Disks

This section describes how to:

- ◆ Replace coordinator disk in the coordinator disk group
- ◆ Destroy a coordinator disk group

---

**Note** Adding or removing coordinator disks requires all services be shut down.

---

Note the following about the procedure:

- ✓ A coordinator disk group requires an odd number (three minimum) of disks/LUNs.
- ✓ When adding a disk, add the disk to the disk group `vxfcntlorddg` and retest the group for support of SCSI-3 persistent reservations.
- ✓ You can destroy the coordinator disk group such that no registration keys remain on the disks. The disks can then be used elsewhere.

### ▼ To remove and replace a disk in the coordinator disk group

1. Log in as root user on one of the cluster nodes.
2. If VCS is running, shut it down:

```
# hactop -all
```

3. Stop the VCSMM driver on each node:

```
# /sbin/vcsmmconfig -U
```

4. Stop I/O fencing on all nodes:

```
# /sbin/init.d/vxfen stop
```

This removes any registration keys on the disks.

5. Import the coordinator disk group. The file `/etc/vxfendg` includes the name of the disk group (typically, `vxfcntlorddg`) that contains the coordinator disks, so use the command:

```
# vxdg -tfc import `cat /etc/vxfendg`
```

where:

- t specifies that the disk group is imported only until the node restarts.
- f specifies that the import is to be done forcibly, which is necessary if one or more disks is not accessible.
- C specifies that any import blocks are removed.



- 
- To remove disks from the disk group, use the VxVM disk administrator utility, `vxdiskadm`.

---

**Note** You may also destroy the existing coordinator disk group. For example:

```
# vxdg destroy vxfencoordg
```

---

- Add the new disk to the node, initialize it as a VxVM disk, and add it to the `vxfencoordg` disk group. Refer to [“Creating the vxfencoordg Disk Group”](#) on page 55
- Test the recreated disk group for SCSI-3 persistent reservations compliance. Refer to [“Requirements for Testing the Coordinator Disk Group”](#) on page 55.
- After replacing disks in a coordinator disk group, deport the disk group:

```
# vxdg deport `cat /etc/vxfendg`
```
- On each node, start the I/O fencing driver:

```
# /sbin/init.d/vxfen start
```
- On each node, start the VCSMM driver:

```
# /sbin/init.d/vcsmm start
```
- If necessary, restart VCS on each node:

```
# hastart
```





# Sample Configuration Files

---

**A**

This appendix provides sample files for the `main.cf`, `types.cf`, `CFSTypes.cf`, `CVMTypes.cf`, `OracleTypes.cf`, and `VRTSWebAppType.cf`.

## Sample 1: main.cf

```
include "types.cf"
include "CFSTypes.cf"
include "CVMTypes.cf"
include "OracleTypes.cf"

cluster vcs (
    UserNames = { admin = fopHojOlpKppNxpJom }
    Administrators = { admin }
    CounterInterval = 5
    UseFence = SCSI3
)

system sysa (
)

system sysb (
)

system sysc (
)

system sysd (
)

group cvm (
    SystemList = { sysa = 0, sysb = 1, sysc = 2, sysd = 3 }
    AutoFailOver = 0
    Parallel = 1
    AutoStartList = { sysa, sysb, sysc, sysd }
)
```



```
CFSMount orabin_mnt (  
    Critical = 0  
    MountPoint = "/oracle"  
    BlockDevice = "/dev/vx/dsk/orabindg/orabinvol"  
)  
  
CFSfsckd vxfsckd (  
)  
  
CVMCluster cvm_clus (  
    Critical = 0  
    CVMClustName = vcs  
    CVMNodeId = { sysa = 0, sysb = 1, sysc = 2,  
                 sysd = 3 }  
    CVMTransport = gab  
    CVMTimeout = 200  
)  
  
CVMVolDg orabin_voldg (  
    Critical = 0  
    CVMDiskGroup = orabindg  
    CVMVolume = { orabinvol, srvmvol }  
    CVMActivation = sw  
)  
  
CVMVxconfigd cvm_vxconfigd (  
    Critical = 0  
    CVMVxconfigdArgs = { syslog }  
)  
  
IP listener_ip (  
    Critical = 0  
    Device = lan0  
    Address @sysa = "192.2.40.21"  
    Address @sysb = "192.2.40.22"  
    Address @sysc = "192.2.40.23"  
    Address @sysd = "192.2.40.24"  
    NetMask = "255.255.240.0"  
)  
  
NIC listener_lan0 (  
    Critical = 0  
    Device = lan0  
    NetworkType = ether  
    NetworkHosts = { "10.180.11.1", "10.180.11.2" }  
)
```



```

Netlsnr LISTENER (
    Critical = 0
    Owner = oracle
    Home = "/oracle/orahome"
    TnsAdmin = "/oracle/orahome/network/admin"
    Listener @sysa = LISTENER_a
    Listener @sysb = LISTENER_b
    Listener @sysc = LISTENER_c
    Listener @sysd = LISTENER_d
    EnvFile = "/opt/VRTSvcs/bin/Netlsnr/envfile"
    MonScript = "./bin/Netlsnr/LsnrTest.pl"
)

```

```

LISTENER requires listener_ip
LISTENER requires orabin_mnt
cvm_clus requires cvm_vxconfigd
listener_ip requires listener_lan0
orabin_mnt requires orabin_voldg
orabin_mnt requires vxfsckd
vxfsckd requires cvm_clus

```

```

// resource dependency tree
//
//     group cvm
//     {
//         Netlsnr LISTENER
//         {
//             IP listener_ip
//             {
//                 NIC listener_lan0
//             }
//             CFSMount orabin_mnt
//             {
//                 CFSfsckd vxfsckd
//                 {
//                     CVMCluster cvm_clus
//                     {
//                         CVMVxconfigd cvm_vxconfigd
//                     }
//                 }
//             }
//             CVMVoldg orabin_voldg
//         }
//     }
// }

```



```
group oradb1_grp (
    SystemList = { sysa = 0, sysb = 1, sysc = 2, sysd = 3 }
    Parallel = 1
    AutoStartList = { sysa, sysb, sysc, sysd }
)

CFMount oradb1_mnt (
    MountPoint = "/oradb1"
    BlockDevice = "/dev/vx/dsk/oradb1dg/oradb1vol"
)

CVMVolDg oradb1_voldg (
    CVMDiskGroup = oradb1dg
    CVMVolume = { oradb1vol }
    CVMActivation = sw
)

Oracle VRT (
    Sid @sysa = VRT1
    Sid @sysb = VRT2
    Sid @sysc = VRT3
    Sid @sysd = VRT4
    Owner = oracle
    Home = "/oracle/orahome"
    Pfile @sysa = "/oracle/orahome/dbs/initVRT1.ora"
    Pfile @sysb = "/oracle/orahome/dbs/initVRT2.ora"
    Pfile @sysc = "/oracle/orahome/dbs/initVRT3.ora"
    Pfile @sysd = "/oracle/orahome/dbs/initVRT4.ora"
    EnvFile = "/opt/VRTSvc/bin/Oracle/envfile"
    MonScript = "./bin/Oracle/SqlTest.pl"
    User = scott
    Pword = tiger
    Table @sysa = vcstable_sysa
    Table @sysb = vcstable_sysb
    Table @sysc = vcstable_sysc
    Table @sysd = vcstable_sysd
)

requires group cvm online local firm
VRT requires oradb1_mnt
oradb1_mnt requires oradb1_voldg

// resource dependency tree
//
//     group oradb1_grp
//     {
```



```
//      Oracle VRT
//      {
//          CFSSMount oradb1_mnt
//          {
//              CVMVolDg oradb1_voldg
//          }
//      }

group oradb2_grp (
    SystemList = { sysa = 1, sysb = 0, sysc = 2, sysd = 3 }
    Parallel = 1
    AutoStartList = { sysb, sysa, sysc, sysd }
)

CFSSMount oradb2_mnt (
    MountPoint = "/oradb2"
    BlockDevice = "/dev/vx/dsk/oradb2g2/oradb2vol"
)

CVMVolDg oradb2_voldg (
    CVMDiskGroup = oradb2g2
    CVMVolume = { oradb2vol }
    CVMActivation = sw
)

Oracle rac (
    Sid @sysa = rac1
    Sid @sysb = rac2
    Sid @sysc = rac3
    Sid @sysd = rac4
    Owner = oracle
    Home = "/oracle/orahome"
    Pfile @sysa = "/oracle/orahome/dbs/initrac1.ora"
    Pfile @sysb = "/oracle/orahome/dbs/initrac2.ora"
    Pfile @sysc = "/oracle/orahome/dbs/initrac3.ora"
    Pfile @sysd = "/oracle/orahome/dbs/initrac4.ora"
    EnvFile = "/opt/VRTSvcs/bin/Oracle/envfile"
    MonScript = "./bin/Oracle/SqlTest.pl"
    User = scott
    Pword = tiger
    Table @sysa = vcstable_sysa
    Table @sysb = vcstable_sysb
    Table @sysc = vcstable_sysc
    Table @sysd = vcstable_sysd
)
```



```
requires group cvm online local firm
oradb2_mnt requires oradb2_voldg
rac requires oradb2_mnt

// resource dependency tree
//
//     group oradb2_grp
//     {
//     Oracle rac
//     {
//     CFSMount oradb2_mnt
//     {
//     CVMVolDg oradb2_voldg
//     }
//     }
//     }
// }
```

## Sample 2: types.cf

```
type Application (
    static str ArgList[] = { User, StartProgram, StopProgram,
                            CleanProgram, MonitorProgram,
                            PidFiles, MonitorProcesses }

    str User = root
    str StartProgram
    str StopProgram
    str CleanProgram
    str MonitorProgram
    str PidFiles[]
    str MonitorProcesses[]
)

type DiskGroup (
    static int NumThreads = 1
    static int OnlineRetryLimit = 1
    static str ArgList[] = { DiskGroup, StartVolumes,
                            StopVolumes, MonitorOnly }

    str DiskGroup
    str StartVolumes = 1
    str StopVolumes = 1
)

type ElifNone (
```



```
static int OfflineMonitorInterval = 60
static str ArgList[] = { PathName }
static str Operations = None
str PathName
)

type FileNone (
    static int OfflineMonitorInterval = 60
    static str ArgList[] = { PathName }
    static str Operations = None
    str PathName
)

type FileOnOff (
    static str ArgList[] = { PathName }
    str PathName
)

type FileOnOnly (
    static str ArgList[] = { PathName }
    static str Operations = OnOnly
    str PathName
)

type IP (
    static str ArgList[] = { Device, Address, NetMask, Options,
                           ArpDelay, IfconfigTwice }

    str Device
    str Address
    str NetMask
    str Options
    int ArpDelay = 1
    int IfconfigTwice
)

type IPMultiNIC (
    static int MonitorTimeout = 120
    static str ArgList[] = { "MultiNICResName:Device", Address,
                           NetMask,
                           "MultiNICResName:ArpDelay",
                           Options,
                           "MultiNICResName:Probed",
                           MultiNICResName,
                           IfconfigTwice }

    str Address
    str NetMask
    str Options
```



```
        str MultiNICResName
        int IfconfigTwice
    )

type LVMCombo (
    static str ArgList[] = { VolumeGroup, LogicalVolumes }
    str VolumeGroup
    str LogicalVolumes[]
)

type LVMLogicalVolume (
    static str ArgList[] = { LogicalVolume, VolumeGroup }
    str LogicalVolume
    str VolumeGroup
)

type LVMVolumeGroup (
    static str ArgList[] = { VolumeGroup }
    str VolumeGroup
)

type Mount (
    static str ArgList[] = { MountPoint, BlockDevice, FSType,
                           MountOpt, FsckOpt, SnapUmount }

    str MountPoint
    str BlockDevice
    str FSType
    str MountOpt
    str FsckOpt
    int SnapUmount
)

type MultiNICA (
    static int MonitorTimeout = 300
    static int OfflineMonitorInterval = 60
    static str ArgList[] = { Device, NetMask, ArpDelay,
                           Options, RouteOptions,
                           MonitorOnly, IfconfigTwice,
                           HandshakeInterval,
                           NetworkHosts }

    static str Operations = None
    str Device{}
    str NetMask
    int ArpDelay = 1
    str Options
    str RouteOptions
    int IfconfigTwice
```

```
    int HandshakeInterval = 90
    str NetworkHosts[]
)

type NFS (
    static int RestartLimit = 1
    static str ArgList[] = { Nservers, Protocol }
    static str Operations = OnOnly
    int Nservers = 4
    str Protocol = all
)

type NIC (
    static int OfflineMonitorInterval = 60
    static str ArgList[] = { Device, NetworkType,
                           NetworkHosts }

    static str Operations = None
    str Device
    str NetworkType
    str NetworkHosts[]
)

type NotifierMngr (
    static int RestartLimit = 3
    static str ArgList[] = { EngineListeningPort,
                           MessagesQueue,
                           NotifierListeningPort,
                           SnmpdTrapPort, SnmpCommunity,
                           SnmpConsoles, SntpServer,
                           SntpRecipients }

    int EngineListeningPort = 14141
    int MessagesQueue = 30
    int NotifierListeningPort = 14144
    int SnmpdTrapPort = 162
    str SnmpCommunity = public
    str SnmpConsoles{}
    str SntpServer
    str SntpRecipients{}
)

type Phantom (
    static str ArgList[] = { Dummy }
    str Dummy
)

type Process (
    static str ArgList[] = { PathName, Arguments }
```



```
        str PathName
        str Arguments
    )

type Proxy (
    static int OfflineMonitorInterval = 60
    static str ArgList[] = { TargetResName, TargetSysName,
                            "TargetResName:Probed",
                            "TargetResName:State" }

    static str Operations = None
    str TargetResName
    str TargetSysName
)

type ServiceGroupHB (
    static int OnlineRetryLimit = 5
    static str ArgList[] = { Disks, AllOrNone }
    str Disks[]
    boolean AllOrNone = 1
)

type Share (
    static int NumThreads = 1
    static str ArgList[] = { PathName, Options }
    str PathName
    str Options
)

type VRTSWebApp (
    static int NumThreads = 1
    static str ArgList[] = { AppName, InstallDir,
                            TimeForOnline }

    str AppName
    str InstallDir
    int TimeForOnline
)

type Volume (
    static int NumThreads = 1
    static str ArgList[] = { Volume, DiskGroup }
    str Volume
    str DiskGroup
)
```



## Sample 3: CFSTypes.cf

```
type CFSSMount (
    static keylist RegList = { MountOpt, Policy, NodeList,
                             ForceOff, SetPrimary }
    static int FaultOnMonitorTimeouts = 1
    static int OnlineWaitLimit = 0
    static str ArgList[] = { MountPoint, BlockDevice,
                             MountOpt }

    str MountPoint
    str MountType
    str BlockDevice
    str MountOpt
    keylist NodeList
    keylist Policy
    str Primary
    str SetPrimary
    str RemountRes
    str ForceOff
)

type CFSQlogckd (
    static str ArgList[] = { ResType }
    str ResType
)

type CFSfsckd (
    static int RestartLimit = 1
    static str ArgList[] = { ResType }
    str ResType
    str ActivationMode{}
)
```



## Sample 4: CVMTypes.cf

```

type CVMCluster (
    static int NumThreads = 1
    static int OnlineRetryLimit = 2
    static int OnlineTimeout = 400
    static str ArgList[] = { CVMTransport, CVMClustName,
                            CVMNodeAddr, CVMNodeId,
                            PortConfigd, PortKmsgd,
                            CVMTimeout }

    str CVMClustName
    str CVMNodeAddr{}
    str CVMNodeId{}
    str CVMTransport
    int PortConfigd
    int PortKmsgd
    int CVMTimeout
)

type CVMVolDg (
    static keylist RegList = { CVMActivation }
    static str ArgList[] = { CVMDiskGroup, CVMVolume,
                            CVMActivation }

    str CVMDiskGroup
    keylist CVMVolume
    str CVMActivation
)

```

## Sample 5: OracleTypes.cf

```

type Netlsnr (
    static keylist SupportedActions = { VRTS_GetInstanceName,
                                        VRTS_GetRunningServices
    }

    static str ArgList[] = { Owner, Home, TnsAdmin, Listener,
                            EnvFile, MonScript, LsnrPwd,
                            AgentDebug, Encoding }

    str Owner
    str Home
    str TnsAdmin
    str Listener
    str EnvFile
    str MonScript = "./bin/Netlsnr/LsnrTest.pl"
    str LsnrPwd
    boolean AgentDebug = 0
)

```



```

        str Encoding
    )

type Oracle (
    static keylist SupportedActions = { VRTS_GetInstanceName,
                                        VRTS_GetRunningServices,
                                        DBRestrict,
                                        DBUndoRestrict, DBResume,
                                        DBSuspend, DBTbspBackup }

    static str ArgList[] = { Sid, Owner, Home, Pfile,
                            StartUpOpt, ShutDownOpt,
                            EnvFile, AutoEndBkup,
                            DetailMonitor, User, Pword,
                            Table, MonScript, AgentDebug,
                            Encoding }

    str Sid
    str Owner
    str Home
    str Pfile
    str StartUpOpt = STARTUP
    str ShutDownOpt = IMMEDIATE
    str EnvFile
    boolean AutoEndBkup = 1
    int DetailMonitor
    str MonScript = "./bin/Oracle/SqlTest.pl"
    str User
    str Pword
    str Table
    boolean AgentDebug = 0
    str Encoding
)

```

## Sample 6: VRTSWebAppType.cf

```

type VRTSWebApp (
    static str ArgList[] = { AppName, InstallDir,
                            TimeForOnline }

    NameRule = VRTSWebApp_ + resource.AppName
    str AppName
    str InstallDir
    int TimeForOnline
    static int NumThreads = 1
)

```





# CVMCluster, CVMVxconfigd, CVMVoIdg, and CFSMount Agents

---

This appendix describes the entry points and attributes of the CVMCluster, CVMVxconfigd, CVMVoIdg, and CFSMount agents. Use this information to make necessary changes to the configuration. Refer to the *VERITAS Cluster Server User's Guide* for information on how to modify the VCS configuration.

## CVMCluster Agent

The SFRAC installation automatically configures the CVMCluster resource. The CVMCluster agent controls system membership on the cluster port associated with VxVM.

### CVMCluster Agent: Entry Points

The following table describes the entry points for the CVMCluster agent.

---

<b>Entry Point</b>	<b>Description</b>
Online	Joins a system to the CVM cluster port. Enables Volume Manager cluster functionality by autoimporting shared disk groups.
Offline	Removes a system from the CVM cluster port.
Monitor	Monitors the state of CVM cluster membership for a system.

---



## CVMCluster Agent: Type Attributes

The following table describes modifiable attributes of the CVMCluster resource type.

Attribute	Dimension	Description
CVMClustName	string-scalar	Name of the cluster.
CVMNodeId{}	string-association	Associative list. The first part names the system; the second part contains the LLT ID number for the system.
CVMTransport	string-scalar	Cluster messaging mechanism. Default = gab
CVMTimeout	integer-scalar	Timeout in seconds for CVM cluster reconfiguration. Default = 200

## CVMCluster Agent: Type Definition

The `CVMTypes.cf` file includes the following type definition. The `CVMNodeAddr`, `PortConfigd`, and `PortKmsgd` attributes are not applicable in a RAC environment because GAB, the required mechanism for cluster communication, does not use these attributes.

```

type CVMCluster (
    static int NumThreads = 1
    static int OnlineRetryLimit = 2
    static int OnlineTimeout = 400
    static str ArgList[] = { CVMTransport, CVMClustName,
        CVMNodeAddr, CVMNodeId, PortConfigd, PortKmsgd,
        CVMTimeout }
    NameRule = ""
    str CVMClustName
    str CVMNodeAddr{}
    str CVMNodeId{}
    str CVMTransport
    int PortConfigd
    int PortKmsgd
    int CVMTimeout
)

```



## CVMCluster Agent: Sample Configuration

Below is a sample definition for the CVMCluster resource in the CVM service group. See [Appendix A, “Sample Configuration Files”](#) on page 229 for an extensive `main.cf` that includes the CVMCluster resource.

```
CVMCluster cvm_clus (  
    Critical = 0  
    CVMClustName = RACCluster1  
    CVMNodeId = { galaxy = 0, nebula = 1 }  
    CVMTransport = gab  
    CVMTimeout = 200  
)
```



## CVMVxconfigd Agent

The CVMVxconfigd agent starts and monitors the `vxconfigd` daemon. The `vxconfigd` daemon maintains disk and disk group configurations, communicates configuration changes to the kernel, and modifies configuration information stored on disks. CVMVxconfigd must be present in the CVM service group.

The CVMVxconfigd agent is an OnOnly agent; the agent starts the resource when the cluster starts up and VCS restarts the resource when necessary. The `Operations` attribute specifies these default aspects of startup.

VERITAS recommends starting the `vxconfigd` daemon with the `syslog` option, which enables logging of debug messages. The SFRAC installation configures the `syslog` option for the CVMVxconfigd agent.

### CVMVxconfigd Agent: Entry Points

---

Entry Point	Description
Online	Starts the <code>vxconfigd</code> daemon.
Offline	N/A.
Monitor	Monitors whether <code>vxconfigd</code> daemon is running.

---

### CVMVxconfigd Agent: Type Attribute

---

Attribute	Dimension	Description
CVMVxconfigdArgs	keylist	List of arguments sent to the <code>online</code> entry point. VERITAS recommends always specifying the <code>syslog</code> option.

---



## CVMVxconfigd: Type Definition

The `CVMTypes.cf` file includes the following type definition.

```
type CVMVxconfigd (
    static int FaultOnMonitorTimeouts = 2
    static int RestartLimit = 5
    static str ArgList[] { CVMVxconfigdArgs }
    static str Operations = OnOnly
    keylist CVMVxconfigdArgs
)
```

## CVMVxconfigd Agent: Sample Configuration

Below is a sample definition for the CVMVxconfigd resource in the CVM service group. See [Appendix A, “Sample Configuration Files”](#) on page 229 for an extensive `main.cf` that includes the CVMVxconfigd resource.

```
CVMVxconfigd cvm_vxconfigd (
    Critical = 0
    CVMVxconfigdArgs = { syslog }
)
.
.
cvm_clus requires cvm_vxconfigd
// resource dependency tree
//
// group cvm
// {
// CVMCluster cvm_clus
//     {
//         CVMVxconfigd cvm_vxconfigd
//     }
// }
```



## Configuring the CVMVolDg and CFSSMount Resources

The CVMVolDg agent represents and controls CVM disk groups and CVM volumes within the disk groups. The global nature of CVM disk groups and volumes requires importing them only once on the CVM master node.

Configure the CVMVolDg agent for each disk group used by an Oracle service group. Configure a disk group for only a single Oracle service group. If the database uses cluster file systems, configure the CFSSMount agent for each volume in the disk group.

### CVMVolDg Agent: Entry Points

The following table describes the entry points for the CVMVolDg agent.

Entry Point	Description
Online	Starts all volumes in the shared disk group specified by the <code>CVMVolume</code> attribute. Sets the disk group activation mode to shared-write if the value of the <code>CVMActivation</code> attribute is <code>sw</code> . You can set the activation mode on both slave and master systems.
Offline	Cleans up the temporary files created by the <code>online</code> entry point.
Monitor	Monitors specified critical volumes in the disk group. The <code>CVMVolume</code> attribute specifies these volumes. SFRAC requires specifying at least one volume in a disk group.
Clean	Cleans up the temporary files created by the <code>online</code> entry point.

## CVMVolDg Agent: Type Attributes

The following table describes modifiable attributes of the CVMVolDg resource type.

Attribute	Dimension	Description
CVMDiskGroup	string-scalar	Names the disk group.
CVMVolume	string-keylist	Lists critical volumes in the disk group. SFRAC requires specifying at least one volume in the disk group.
CVMActivation	string-scalar	Sets the activation mode for the disk group. Default = sw

## CVMVolDg Agent: Type Definition

The `CVMTypes.cf` file includes the CVMVolDg type definition.

```

type CVMVolDg (
    static keylist RegList = { CVMActivation }
    static str ArgList[] = { CVMDiskGroup, CVMVolume,
        CVMActivation }
    str CVMDiskGroup
    keylist CVMVolume[]
    str CVMActivation
    temp int voldg_stat
)

```

## CVMVolDg Agent: Sample Configuration

Each Oracle service group requires a defined CVMVolDg resource type.

```

CVMVolDg ora_voldg (
    CVMDiskGroup = oradatadg
    CVMVolume = { oradata1, oradata2 }
    CVMActivation = sw
)

```



## CFSSMount Agent: Entry Points

The CFSSMount agent brings online, takes offline, and monitors a cluster file system mount point. The agent executable is `/opt/VRTSvcs/bin/CFSSMount/CFSSMountAgent`. The CFSSMount type definition is in the `/etc/VRTSvcs/conf/config/CFSSTypes.cf` file.

Entry Point	Description
Online	Mounts a block device in cluster mode.
Offline	Unmounts the file system, forcing the process if necessary, and sets primary to secondary if necessary.
Monitor	Determines whether the file system is mounted. Checks mount status using the <code>fsclustadm</code> command.
Clean	Generates a null operation for a cluster file system mount.

## CFSSMount Agent Type: Attribute Descriptions

The table lists modifiable attributes of the CFSSMount Agent resource type.

Attribute	Dimension	Description
MountPoint	string-scalar	Directory for the mount point.
BlockDevice	string-scalar	Block device for the mount point.
NodeList (optional)	string-keylist	List of nodes on which to mount. If NodeList is NULL, the agent uses the service group system list.
MountOpt (optional)	string-scalar	Options for the mount command. To create a valid MountOpt attribute string: <ul style="list-style-type: none"> <li>◆ Use the VxFS type-specific options only.</li> <li>◆ Do not use the <code>-o</code> flag to specify the VxFS-specific options.</li> <li>◆ Do not use the <code>-F VxFS</code> file system type option.</li> <li>◆ Be aware the cluster option is not required.</li> <li>◆ Specify options in comma-separated list as in these examples:               <pre>ro ro,cluster blkclear,mincache=closesync</pre> </li> </ul>



Attribute	Dimension	Description
Policy (optional)	string-scalar	List of nodes to assume primaryship of the cluster file system if the primary node fails. If set to NULL or if none of the hosts specified in the list is active when the primary node fails, a node is randomly selected from the set of active nodes to assume primaryship.
Primary (Not set by user)	string-scalar	Information only. Stores the primary node name for a VxFS file system. The value is automatically modified in the configuration file when an unmounted file system is mounted or another node becomes the primary node. The user does not set this attribute and user programs do not rely on this attribute.

## CFSMount Agent: Type Definition

The `CFSTypes.cf` file includes the CFSMount agent type definition.

```

type CFSMount (
    static keylist RegList = { MountOpt, Policy, NodeList }
    static int FaultOnMonitorTimeouts = 1
    static int OnlineWaitLimit = 0
    static str ArgList[] = { MountPoint, BlockDevice,
        MountOpt }
    NameRule = resource.MountPoint
    str MountPoint
    str MountType
    str BlockDevice
    str MountOpt
    str Primary
    keylist NodeList
    keylist Policy
)

```



## CFSMount Agent: Sample Configuration

Each Oracle service group requires a defined CFSMount resource type.

```
CFSMount ora_mount (  
    MountPoint = "/oradata"  
    BlockDevice = "/dev/vx/dsk/oradatadg/oradatavol1"  
    Primary = nebula  
)
```

## Tunable Kernel Driver Parameters

Do not use the tunable parameters described in this appendix to enhance performance. Several of the parameters preallocate memory for critical data structures; a change in their values could increase memory use or degrade performance.

---

**Note** VERITAS does not recommend changing the tunable kernel parameters without assistance from VERITAS support personnel.

---

### LMX Tunable Parameters

The following table describes tunable parameters for the LMX driver:

LMX Parameter	Description	Default Value	Maximum Value
lmx_minor_max	Specifies the maximum number of contexts system-wide. Each Oracle process typically has two LMX contexts. “Contexts” and “minors” are used interchangeably in the documentation; “context” is an Oracle-specific term to specify the value in the <code>lmx.conf</code> file.	8192	65535
lmx_port_max	Specifies the number of communication endpoints for transferring messages from the sender to receiver in a uni-directional manner.	4096	65535
lmx_buffer_max	Specifies the number of addressable regions in memory to copy LMX data.	4096	65535



## Configuring LMX Parameters

If you see the message “no minors available” on one node, add a configuration parameter that increases the value for the maximum number of contexts. (While the “minors” term appears in the error message, use the “contexts” term in the configuration file.) Increasing the number of contexts on a node has some impact on the resources of that node.

For the parameter changes to take effect, reconfigure the LMX module using the following steps:

1. Shut down all Oracle service groups on the node:

```
# hagrp -offline oragrp -sys galaxy
```

2. Stop all Oracle client processes, such as sqlplus and svrmgr1, on the node.

3. Unconfigure the LMX module:

```
# /sbin/lmxconfig -U
```

4. Unload the LMX module:

```
# /usr/sbin/kcmodule lmx=unused
```

5. Configure the tunable parameter:

```
# /usr/sbin/kctune tunable=value
```

For example:

```
# /usr/sbin/kctune lmx_minor_max=16384
```

6. Start the LMX module:

```
# /sbin/init.d/lmx start
```

7. Bring the service groups online:

```
# hagrp -online oragrp -sys galaxy
```



## VCSMM Tunable Parameters

The following table describes tunable parameters for the VCSMM driver:

VCSMM Parameter	Description	Default Value	Minimum Value	Maximum Value
mm_deblog_sz	Size of VCSMM debug log in bytes.	65536	257	65536
mm_msglog_sz	Size of VCSMM message log in bytes.	128	128	65536
mm_slave_max	Max slaves per primary node. Value must be identical on each node.	256	256	32768

### Configuring VCSMM Parameters

For parameter changes to take effect, reconfigure the VCSMM module using the following steps:

1. Shut down all Oracle service groups on the node:

```
# hagr -offline oragrp -sys galaxy
```

2. Stop all Oracle client processes, such as `sqlplus`, `svrmgr1`, and `gsd`, on the node.

3. Unconfigure the VCSMM module:

```
# /sbin/vcsmmconfig -U
```

4. Unload the module:

```
# /usr/sbin/kcmodule vcsmm=unused
```

5. Configure the tunable parameter:

```
# /usr/sbin/kctune tunable=value
```

For example:

```
# /usr/sbin/kctune mm_slave_max=256
```

6. Start the VCSMM module:

```
# /sbin/init.d/vcsmm start
```



7. Bring the service groups online:

```
# hagrp -online oragrp -sys galaxy
```

## VXFEN Tunable Parameters

The following table describes tunable parameters for the VXFEN driver:

vxfen Parameter	Description	Default Value	Minimum Value	Maximum Value
vxfen_debug_sz	Size of debug log in bytes.	65536	65536	256K
vxfen_max_read_coord_disk	Maximum number of times to read coordinator disks	25	1	1000

## Configuring the VXFEN Parameters

For the parameter changes to take effect, reconfigure the VXFEN module using the following steps:

1. Shut down all Oracle service groups on the node:

```
# hagrp -offline oragrp -sys galaxy
```

2. Stop all Oracle client processes, such as `sqlplus`, `svrmgr1`, and `gsd`, on the node.

3. Unconfigure the VXFEN module:

```
# /sbin/vxfenconfig -U
```

4. Unload the module:

```
# /usr/sbin/kcmodule vxfen=unused
```

5. Configure the tunable parameter:

```
# /usr/sbin/kctune tunable=value
```

For example:

```
# /usr/sbin/kctune vxfen_debug_sz=65536
```

6. Start the VXFEN module:

```
# /sbin/init.d/vxfen start
```

7. Bring the service groups online:

```
# hagrps -online oragrp -sys galaxy
```





# Error Messages



The error messages listed in this appendix are grouped by the software module in which the error occurs.

## LMX Error Messages, Critical

The following table lists LMX kernel module error messages. These messages report critical errors when the system runs out of memory, LMX is unable to communicate with LLT, or you are unable to load or unload LMX. Refer to [“Running Scripts for Engineering Support Analysis”](#) on page 215 on gathering information about systems and configurations for VERITAS support personnel.

Message ID	LMX Message
00001	lmxload packet header size incorrect ( <i>number</i> )
00002	lmxload invalid lmx_llt_port <i>number</i>
00003	lmxload context memory alloc failed
00004	lmxload port memory alloc failed
00005	lmxload buffer memory alloc failed
00006	lmxload node memory alloc failed
00007	lmxload msgbuf memory alloc failed
00008	lmxload tmp msgbuf memory alloc failed
00009	lmxunload node <i>number</i> conngrp not NULL
00010	lmxopen return, minor non-zero



Message ID	LMX Message
00011	lmxopen return, no minors available
00012	lmxconnect lmxlltopen(1) err= <i>number</i>
00013	lmxconnect new connection memory alloc failed
00014	lmxconnect kernel request memory alloc failed
00015	lmxconnect mblk memory alloc failed
00016	lmxconnect conn group memory alloc failed
00017	lmxlltfini: LLT unregister failed err = <i>number</i>
00018	lmxload contexts <i>number</i> > <i>number</i> , max contexts = system limit = <i>number</i>
00019	lmxload ports <i>number</i> > <i>number</i> , max ports = system limit = <i>number</i>
00020	lmxload buffers <i>number</i> > <i>number</i> , max buffers = system limit = <i>number</i>
00021	lmxload msgbuf <i>number</i> > <i>number</i> , max msgbuf size = system limit = <i>number</i>

## LMX Error Messages, Non-Critical

The following table contains LMX error messages that may appear during run-time. Refer to [“Running Scripts for Engineering Support Analysis”](#) on page 215 for information on gathering information about systems and configurations for VERITAS support personnel.

If the display of these messages creates errors while running an Oracle application, use the `lmxconfig` command to turn off the display. For example:

```
# /sbin/lmxconfig -e 0
```

To re-enable the display of the messages, type:

```
# /sbin/lmxconfig -e 1
```



<b>Message ID</b>	<b>LMX Message</b>
06001	lmxreqlink duplicate kreq= <i>0xaddress</i> , req= <i>0xaddress</i>
06002	lmxreqlink duplicate ureq= <i>0xaddress</i> kr1= <i>0xaddress</i> , kr2= <i>0xaddress</i> req type = number
06003	lmxrequnlink not found kreq= <i>0xaddress</i> from= number
06004	lmxrequnlink_l not found kreq= <i>0xaddress</i> from= number
06101	lmxpollreq not in doneq CONN kreq= <i>0xaddress</i>
06201	lmxnewcontext lltinit fail err= number
06202	lmxnewcontext lltregister fail err= number
06301	lmxrecvport port not found unode= number node= number ctx= <i>number</i>
06302	lmxrecvport port not found (no port) ctx= <i>number</i>
06303	lmxrecvport port not found ugen= <i>number</i> gen= <i>number</i> ctx= <i>number</i>
06304	lmxrecvport dup request detected
06401	lmxinitport out of ports
06501	lmxsendport lltsend node= <i>number</i> err= <i>number</i>
06601	lmxinitbuf out of buffers
06602	lmxinitbuf fail ctx= <i>number</i> ret= <i>number</i>
06701	lmxsendbuf lltsend node= <i>number</i> err= <i>number</i>
06801	lmxconfig insufficient privilege, uid= <i>number</i>
06901	lmxlltnodestat: LLT getnodeinfo failed err= <i>number</i>



## VxVM Error Messages Related to I/O Fencing

Message	Explanation
vold_pgr_register( <i>disk_path</i> ): failed to open the vxfen device. Please make sure that the vxfen driver is installed and configured.	The vxfen driver is not configured. Follow the instructions in “ <a href="#">Setting Up Coordinator Disks</a> ” on page 54 to set up these disks and start I/O fencing. You can then clear the faulted resources and bring the service groups online.
vold_pgr_register( <i>disk_path</i> ): Probably incompatible vxfen driver.	Incompatible versions of VxVM and the vxfen driver are installed on the system. Install the proper version of SFRAC.

## VXFEN Driver Error Messages

Message	Explanation
VXFEN: Unable to register with coordinator disk with serial number: <i>xxxx</i>	This message appears when the vxfen driver is unable to register with one of the coordinator disks. The serial number of the coordinator disk that failed is printed.
VXFEN: Unable to register with a majority of the coordinator disks. Dropping out of cluster.	This message appears when the vxfen driver is unable to register with a majority of the coordinator disks. The problems with the coordinator disks must be cleared before fencing can be enabled.  This message is preceded with the message “VXFEN: Unable to register with coordinator disk with serial number <i>xxxx</i> .”

## VXFEN Driver Informational Message

*date and time* VXFEN:00021:Starting to eject leaving nodes(s) from data disks.

*date and time* VXFEN:00022:Completed ejection of leaving node(s) from data disks.

These messages show the time required to fence data disks for nodes that left the cluster.





## Node Ejection Informational Messages

Informational messages resembling the example below may appear on the console of one of the cluster nodes when a node is ejected from a disk or LUN:

```
<date> <system name> scsi: WARNING:
/sbus@3,0/lpfs@0,0/sd@0,1(sd91):
<date> <system name> Error for Command: <undecoded cmd 0x5f> Error
Level: Informational
<date> <system name> scsi: Requested Block: 0 Error Block 0
<date> <system name> scsi: Vendor: <vendor> Serial Number:
0400759B006E
<date> <system name> scsi: Sense Key: Unit Attention
<date> <system name> scsi: ASC: 0x2a (<vendor unique code 0x2a>),
ASCQ: 0x4, FRU: 0x0
```

You can ignore these messages.





## Creating a Starter Database

---

The optional procedures in this appendix describe methods for creating a starter database on shared storage. Use your own tools or refer to the guidelines below on using the Oracle `dbca` (Database Creation Assistant) tool to create a database on shared raw VxVM volumes or shared VxFS file systems.

---

**Note** If you installed the Oracle binary on shared storage (CFS), VERITAS recommends using a different file system for the Oracle datafiles. The Storage Foundation *for Oracle RAC* configuration wizard does not support the use of a single file system for the binary and the datafiles.

---

## Creating a Starter Database on Raw Volumes Using `dbca`

To create a starter database on raw VxVM volumes, prepare the shared storage for the database tablespaces.

### Creating Shared Raw Volumes for Tablespaces

1. Log in as `root` user.
2. On the master node, create a shared disk group:

```
# vxvg -s init ora_dg c2t3d1s2
```
3. Create a volume in the shared group for *each* of the required tablespaces:

```
# vxassist -g ora_dg make VRT_system1 1000M
# vxassist -g ora_dg make VRT_spfile1 10M
.
.
```
4. Define the access mode and permissions for the volumes storing the Oracle data. For each volume listed in `$ORACLE_HOME/raw_config`, use the `vxedit(1M)` command:



```
vxedit -g disk_group set group=group user=user mode=660 volume
```

For example:

```
# vxedit -g ora_dg set group=dba user=oracle mode=660 VRT_system1
```

In this example, `VRT_system1` is the name of one of the volumes. Repeat the command to define access mode and permissions for each volume in the `ora_dg`.

## Running the dbca Utility for Raw VxVM Volumes

Use the `dbca` utility on the master node to create a general-purpose database on raw VxVM volumes. The utility is a graphical user interface and requires setting the `DISPLAY` environment variable.

1. Make sure an `oracle` account is created on all nodes.
2. Verify that `remsh` works among all nodes under the `oracle` account.
3. Make sure the file configured for `srvm` is accessible to the `oracle` user.
4. Start the Oracle Real Application Clusters Manageability daemon on each node:

```
$ gsdctl start
```

5. From one node, log in as `oracle` user.
6. Create and start the Oracle listener using the `NETCA` utility (refer to the *Oracle9i Installation Guide*):

```
$ netca
```

7. Run the `dbca` utility to create the database. Refer to Oracle documentation for details on the `dbca` utility.

## Creating a Starter Database on CFS Using dbca

When creating a starter database on a cluster file system, create a disk group and volume for the tablespaces before running the dbca utility.

### Preparing a Cluster File System (CFS) for Tablespaces

1. Create a disk group (for example, `rac_dg`):  

```
# vxhdg -s init rac_dg c2t3d1s2
```
2. Create a single shared volume (for example, `rac_vol1`) large enough to contain a file system for all the tablespaces. The tablespaces require about 6.8 GB:  

```
# vxassist -g rac_dg make rac_vol1 6600M
```
3. Start the volume in the disk group:  

```
# vxvol -g rac_dg startall
```
4. Set the activation for the disk group on all cluster nodes:  

```
# vxhdg -g rac_dg set activation=sw
```
5. Create a VxFS file system in this volume. From one node, type:  

```
# mkfs -F vxfs /dev/vx/rdisk/rac_dg/rac_vol1
```
6. Create a mount point for the shared file system:  

```
# mkdir /rac_ts
```
7. From the same node, mount the file system:  

```
# mount -F vxfs -o cluster /dev/vx/dsk/rac_dg/rac_vol1 /rac_ts
```
8. Set "oracle" as the owner of the file system and "755" as the permissions:  

```
# chown oracle:dba /rac_ts  
# chmod 755 /rac_ts
```
9. On the other node(s), complete [step 6](#) through [step 8](#).



## Running the dbca Utility for Cluster File System

As `oracle` user, use the `dbca` utility on the master node to create a general purpose database on a cluster file system. This utility is a graphical user interface and requires setting the `DISPLAY` environment variable.

1. Make sure an `oracle` account is created on all nodes.
2. Verify that `remsh` works among all the nodes under the `oracle` account.
3. Examine the `/var/opt/oracle/srvConfig_loc` configuration file and note the path name defined for the `srvconfig_loc` variable. For example, the path name may resemble:

```
srvconfig_loc=/orasrv/ora_srvm
```

where `/orasrv/ora_srvm` is the path name of the SRVM configuration file.

- a. Use the `ls -l pathname` command to display the long listing for the defined path. The file is initialized if the output shows a file size greater than zero bytes.

- b. If the variable is not initialized (size is 0), type:

```
$ srvconfig -init
```

- c. If the path is not yet defined (listing does not appear), initialize the variable after using the `touch` command to create it:

```
$ touch /orasrv/ora_srvm
```

```
$ srvconfig -init
```

4. Start the Oracle Real Application Clusters Manageability daemon on each node:

```
$ gsdctl start
```

5. From one node, log in as `oracle` user.

6. Create and start the Oracle listener using `NETCA` utility (refer to the *Oracle9i Installation Guide*):

```
$ netca
```

7. Run the `dbca` utility. When starting the utility with a cluster file system, use the `-datafileDestination` option to specify the mount point. For example:

```
$ dbca -datafileDestination /rac_ts
```

Refer to the Oracle documentation for details on the `dbca` utility.



## Linking VERITAS Libraries to Oracle

---

SFRAC provides the IPC, VCSMM, and ODM libraries for use with Oracle9i binaries. You must link these binaries and SFRAC libraries using the `linkrac9i` command described in the following section. Perform this linking procedure after:

- ◆ Installing a supported Oracle9i patch of a higher level to your existing SFRAC installation.
- ◆ Installing or reinstalling SFRAC.

### Running `linkrac9i`

1. On each node, shut down the Oracle service group:

```
# hagrps -offline oracle_database_grp -sys system_name
```

2. On one node, run the following command as `oracle` user to link Oracle with the VERITAS IPC, VCSMM, and ODM libraries:

```
$ /opt/VRTSvcs/rac/bin/linkrac9i
```

---

**Note** If the Oracle binaries are on a cluster file system, perform this step on only one node. If the Oracle binaries are on a local file system of each cluster node, run the command to link Oracle with VERITAS libraries on each cluster node.

---

If your system uses a bundled C compiler or a compiler other than the ANSI C compiler, you can safely ignore such warnings as:

```
(Bundled) cc: warning 922: "+Oshortdata=8" is unsupported in the  
bundled compiler, ignored.
```

Refer to *Oracle Metalink Document ID 66442.1* for more information.  
Search on: 66442.1 FAQ about C Compiler Issues on HP-UX.

3. On each node, start the Oracle service group:

```
# hagrps -online oracle_database_grp -sys system_name
```



4. After starting the Oracle instance, confirm Oracle uses the VERITAS libraries. Examine the Oracle alert file, `alert_$(ORACLE_SID).log`, for the following lines:

```
Oracle instance running with ODM: VERITAS 4.1 ODM Library,  
Version 1.0  
cluster interconnect IPC version:VERITAS IPC 4.1 08:44:27 Apr 20  
2005
```

---

**Note** If Oracle binaries are on a local file system of each cluster node, examine the Oracle alert file on each of the cluster nodes.

---





# Index

---

## A

- agents
  - CFSMount 250
  - CVMCluster 243
  - CVMVolDg 248
  - CVMVxconfigd 246
- attributes
  - CFSMount agent 250
  - CVMCluster agent 244
  - CVMVolDg agent 249
  - CVMVxconfigd 246
  - for CVM and Oracle 104
  - UseFence 59

## C

- Cache Fusion
  - description 3
- CFSMount agent
  - description 250
- CFSTypes.cf
  - sample 239
- Cluster File System (CFS)
  - creating databases on 267
  - overview 9
- Cluster Volume Manager (CVM)
  - overview 8
- commands
  - cron 146
  - format 221
  - installsfrac 31, 39
  - nohup 72, 76
  - sfrac\_ckptcreate 139, 144, 146
  - sfrac\_ckptdisplay 145
  - sfrac\_ckptmount 148
  - sfrac\_ckptremove 151
  - sfrac\_ckptrollback 150
  - sfrac\_ckptumount 149
  - sfrac\_update 143
  - uninstallsfrac 209

- vcsmmconfig 217
- vxassist 69, 267
- vxctl enable 221
- vxdiag list 66
- vxdedit 266
- vxfen start 58
- vxfenadm 127
- vxfenclearpre 225
- vxstorage\_stats 155
- vxvol 267
- configurations
  - backing up main.cf for SFRAC upgrade 201
  - copying main.cf for SFRAC upgrade 202
  - editing for removed nodes 187
  - modifying 105
  - of service groups 85
  - using installsfrac -configure 39
  - using product installer 38
- coordinator disks
  - description 15
  - setting up 54
- cron command 146
- CVMCluster agent
  - description 243
- CVMTypes.cf
  - sample 240
- CVMVolDg agent
  - description 248
- CVMVxconfigd agent
  - description 246

## D

- data corruption
  - preventing with I/O fencing 13
- data disks
  - description 14
- database instance
  - stopping 206



- 
- databases
    - creating 83, 265
    - upgrading 83
  - dbca
    - creating databases on CFS 267
    - creating databases on raw volumes 265
    - description 265
  - dependencies
    - between service groups 88
  - disk groups
    - overview 66
  - disk space
    - required for SFRAC 24
  - disks
    - adding and initializing 48
    - coordinator 15, 54
    - data 14
    - testing with vxfsentsthdw 51
    - verifying node access 49
  - drivers
    - tunable parameters 253
  - E**
    - enterprise agent for Oracle
      - See Oracle enterprise agent
    - environment variables
      - MANPATH 28
      - PATH 28
    - error messages
      - LMX 259
      - node ejection 263
      - VXFEN 262
      - VxVM errors related to I/O fencing 262
  - F**
    - format command 221
  - G**
    - GAB
      - overview 7
      - port memberships 60
    - getcomms
      - troubleshooting SFRAC 215
    - getdbac
      - troubleshooting SFRAC 215
    - gsd
      - stopping 205
  - H**
    - hagetcf
      - troubleshooting SFRAC 215
  - I**
    - I/O fencing
      - components 14
      - coordinator disks 15
      - data disks 14
      - event scenarios 124
      - overview 13
      - setting up 48
      - shared storage 119
      - starting 58
      - testing and scenarios 119
    - installation
      - of Oracle9i 65
      - of SFRAC 27
      - using installsfrac 31
      - using product installer 30
      - utilities 27
    - installsfrac
      - configuring SFRAC 39
      - installing SFRAC 31
  - L**
    - licenses
      - obtaining 21
      - removing keys 25
    - Listener
      - description 10, 96, 99, 104, 266
    - LLT
      - overview 6
    - LMX
      - error messages 259
      - tunable parameters 253
    - Lock management
      - description 3
    - logs
      - for SFRAC configuration 47
      - for SFRAC installation 35
      - for SFRAC uninstallation 212
      - for VCS and Oracle agents 105
  - M**
    - main.cf
      - after SFRAC and Oracle installation 229
      - after SFRAC installation 62
    - MANPATH environment variable 28
  - N**
    - nodes
      - adding 165
      - removing 183



- 
- nohup command 72, 76
  - O**
    - operating systems
      - supported 22
    - Oracle
      - supported versions 22
    - Oracle Disk Manager (ODM)
      - description 9
      - disabling library 82
    - Oracle instance
      - description 1
    - Oracle service group
      - configuring 101
      - creating 105
    - Oracle9i
      - adding patches 80
      - creating databases 83
      - creating storage location for SRVM 69
      - creating users and groups 68
      - disabling ODM library 82
      - disk groups and volumes 66
      - installing Release 2 71
      - linking Oracle to VERITAS libraries 81
      - linking to VERITAS libraries when upgrading Oracle patch 83
      - overview of installation 65
      - tasks to install and configure 69
      - upgrading databases 83
    - OracleTypes.cf
      - sample 240
  - P**
    - patches
      - for HP 23
      - for Oracle9i 80
    - PATH environment variable 28
    - product installer
      - configuring SFRAC 38
      - installing SFRAC 30
      - uninstalling SFRAC 209
  - R**
    - registrations
      - for I/O fencing 119
      - key formatting 128
    - reservations
      - description 14
      - SCSI-3 persistent 19
  - S**
    - SCSI-3 persistent reservations
      - description 13
      - verifying 48, 65
    - service groups
      - configuring for Oracle and CVM 90
      - configuring for SFRAC 85
      - dependencies between 88
      - Oracle 101, 105
      - overview of CVM 86
      - using configuration wizard 105
    - SFRAC
      - adding nodes 165
      - communications stack 5
      - configuration file after SFRAC installation 62
      - configuring 36
      - configuring VCS service groups 85
      - coordinator disks 54
      - data stack 4
      - disabling ODM library 82
      - error messages 259
      - high-level view 17
      - I/O fencing 48, 119
      - information required during installation 26
      - linking libraries with Oracle 81
      - overview 1
      - overview of installation methods 27
      - phases of installation and configuration 19
      - removing nodes 183
      - requirements for installation 21
      - sample configuration files 229
      - shared storage 48
      - Storage Checkpoints 129
      - Storage Mapping 153
      - Storage Rollback 129
      - troubleshooting 215
      - tunable parameters of kernel drivers 253
      - upgrading from 3.5 to 4.1 191
      - using installsfrac 31, 39, 209
      - using product installer 30, 38, 209
    - sfrac\_ckptcreate command 139, 144, 146
    - sfrac\_ckptdisplay command 145
    - sfrac\_ckptmount command 148
    - sfrac\_ckptremove command 151
    - sfrac\_ckptrollback command 150
    - sfrac\_ckptumount command 149



---

sfrac\_update command 143  
split brain  
    description 13  
SRVM  
    creating storage location 69  
storage  
    for I/O fencing 119  
Storage Checkpoints  
    backing up and recovering  
        databases 129, 135  
    creating 144  
    creating repositories for VxDBA 143  
    description 130  
    determining space requirements 131  
    displaying 145  
    mounting 148  
    performance 132  
    removing 151  
    scheduling 146  
    unmounting 149  
    using the CLI 141  
Storage Mapping  
    configuring arrays 163  
    description 153  
    ORAMAP 157  
    using Oracle Enterprise Manager 161  
    using the vxstorage\_stats command 155  
    verifying feature setup 154  
Storage Rollback  
    description 130  
    guidelines for recovery 139  
    using sfrac\_ckptrollback 150

## T

technical assistance xxv  
troubleshooting 215  
types.cf  
    sample 234

## U

Uninstalling DBE/AC 195

uninstallsfrac  
    procedures 209  
upgrade  
    disk layout 203  
    operating system 201  
SFRAC 191

## V

VCS  
    architecture 10  
VCSIPC  
    description 3, 12  
    errors in trace/log files 219  
    warnings in trace files 219  
VCSMM  
    description 3, 12  
    tunable parameters 255  
vcsmmconfig command 217  
volumes  
    creating databases on 265  
    overview 66  
VRTSWebAppType.cf  
    sample 241  
vxassist command 69, 267  
vxdctl command 221  
vxdg list command 66  
vxedit command 266  
VXFEN  
    informational messages 262  
    tunable parameters 256  
vxfen command 58  
vxfenadm command 127  
vxfenclearpre command 225  
vxfentsthaw utility 120  
vxstorage\_stats command 155  
VxVM  
    error messages related to I/O  
    fencing 262  
vxvol command 267

