

Oracle® Solaris Cluster システム管理

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する際、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用了ことに起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel、Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴ、AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

はじめに	11
1 Oracle Solaris Cluster の管理の概要	15
Oracle Solaris Cluster の管理の概要	16
ゾーンクラスタに関する作業	16
Oracle Solaris OS の機能制限	17
管理ツール	18
グラフィカルユーザーインターフェース	18
コマンド行インターフェース	18
クラスタ管理の準備	20
Oracle Solaris Cluster ハードウェア構成の記録	20
管理コンソールの使用	20
クラスタのバックアップ	21
クラスタ管理の開始	21
▼ クラスタにリモートでログインする方法	23
▼ クラスタコンソールに安全に接続する方法	24
▼ クラスタ構成ユーティリティーにアクセスする方法	25
▼ Oracle Solaris Cluster のパッチ情報を表示する方法	26
▼ Oracle Solaris Cluster のリリース情報とバージョン情報を表示する方法	27
▼ 構成されているリソースタイプ、リソースグループ、リソースを表示する方 法	30
▼ クラスタコンポーネントのステータスを確認する方法	31
▼ パブリックネットワークのステータスを確認する方法	34
▼ クラスタ構成を表示する方法	35
▼ 基本的なクラスタ構成を検証する方法	44
▼ グローバルマウントポイントを確認する方法	49
▼ Oracle Solaris Cluster のコマンドログの内容を表示する方法	51

2	Oracle Solaris Cluster と RBAC	53
	RBAC の設定と Oracle Solaris Cluster での使用	53
	Oracle Solaris Cluster RBAC の権利プロファイル	54
	Oracle Solaris Cluster 管理権利プロファイルによる RBAC 役割の作成と割り当て	55
	▼ 管理役割ツールを使用して役割を作成する方法	55
	▼ コマンド行から役割を作成する方法	57
	ユーザーの RBAC プロパティの変更	59
	▼ ユーザーアカウントツールを使用してユーザーの RBAC プロパティを変更する 方法	59
	▼ コマンド行からユーザーの RBAC プロパティを変更する方法	60
3	クラスタの停止とブート	61
	クラスタの停止とブートの概要	61
	▼ クラスタを停止する方法	63
	▼ クラスタをブートする方法	65
	▼ クラスタをリブートする方法	67
	クラスタ内の 1 つのノードの停止とブート	71
	▼ ノードを停止する方法	72
	▼ ノードをブートする方法	76
	▼ ノードをリブートする方法	78
	▼ 非クラスタモードでノードをブートする方法	82
	満杯の /var ファイルシステムを修復する	85
	▼ 満杯の /var ファイルシステムを修復する方法	86
4	データ複製のアプローチ	87
	データ複製についての理解	88
	サポートされるデータ複製方式	89
	クラスタ内でのストレージベースのデータ複製の使用	90
	クラスタ内でストレージベースのデータ複製を使用する際の要件と制限	92
	クラスタ内でストレージベースのデータ複製を使用する際の手動回復に関する懸 念事項	93
	ストレージベースのデータ複製を使用する際のベストプラクティス	94

5	グローバルデバイス、ディスクパス監視、およびクラスタファイルシステムの管理	95
	グローバルデバイスとグローバルな名前空間の管理の概要	95
	Solaris Volume Manager のグローバルデバイスのアクセス権	96
	グローバルデバイスでの動的再構成	96
	ストレージベースの複製されたデバイスの管理	97
	Hitachi TrueCopy で複製されたデバイスの管理	98
	EMC Symmetrix Remote Data Facility で複製されたデバイスの管理	109
	クラスタファイルシステムの管理の概要	121
	クラスタファイルシステムの制限事項	122
	デバイスグループの管理	122
	▼ グローバルデバイス名前空間を更新する方法	124
	▼ グローバルデバイス名前空間で使用する lofi デバイスのサイズを変更する方法	125
	グローバルデバイス名前空間を移行する	126
	▼ 専用パーティションから lofi デバイスにグローバルデバイス名前空間を移行する方法	127
	▼ lofi デバイスから専用パーティションにグローバルデバイス名前空間を移行する方法	128
	デバイスグループを追加および登録する	129
	▼ デバイスグループを追加および登録する方法 (Solaris Volume Manager)	129
	▼ デバイスグループ (raw ディスク) を追加および登録する方法	132
	▼ 複製デバイスグループ (ZFS) の追加と登録方法	132
	デバイスグループ名を保守する	134
	デバイスグループを削除して登録を解除する方法 (Solaris Volume Manager)	134
	▼ すべてのデバイスグループからノードを削除する方法	134
	▼ デバイスグループからノードを削除する方法 (Solaris Volume Manager)	135
	▼ raw ディスクデバイスグループからノードを削除する方法	137
	▼ デバイスグループのプロパティを変更する方法	139
	▼ デバイスグループのセカンダリノードの希望数を設定する方法	141
	▼ デバイスグループ構成の一覧を表示する方法	143
	▼ デバイスグループのプライマリノードを切り替える	145
	▼ デバイスグループを保守状態にする方法	146
	ストレージデバイス用の SCSI プロトコル設定の管理	148
	▼ すべてのストレージデバイスのデフォルトのグローバルな SCSI プロトコル設定を表示する方法	148
	▼ 単一ストレージデバイスの SCSI プロトコルを表示する方法	149

▼すべてのストレージデバイスのデフォルトのグローバルなフェンシングプロトコル設定を変更する方法	150
▼単一ストレージデバイスのフェンシングプロトコルを変更する方法	152
クラスタファイルシステムの管理	153
▼クラスタファイルシステムを追加する方法	154
▼クラスタファイルシステムを削除する方法	157
▼クラスタ内のグローバルマウントを確認する方法	159
ディスクパス監視の管理	160
▼ディスクパスを監視する方法	161
▼ディスクパスの監視を解除する方法	162
▼障害のあるディスクパスを表示する方法	163
▼ディスクパスのステータスエラーを解決する方法	164
▼ファイルからディスクパスを監視する方法	165
▼監視しているすべての共有ディスクパスが失敗したときのノードの自動リブートを有効にする方法	167
▼すべての監視共有ディスクパスが失敗した場合にノードの自動リブートを無効にする方法	168
6 定足数の管理	169
定足数デバイスの管理	169
定足数デバイスへの動的再構成	171
定足数デバイスの追加	172
定足数デバイスの削除または交換	181
定足数デバイスの保守	185
定足数のデフォルトのタイムアウトの変更	192
Oracle Solaris Cluster 定足数サーバーの管理	193
Quorum Server Software の起動および停止	193
▼定足数サーバーを起動する方法	193
▼定足数サーバーを停止する方法	194
定足数サーバーに関する情報の表示	195
期限切れの定足数サーバークラスタ情報のクリーンアップ	196
7 クラスタインターコネクトとパブリックネットワークの管理	199
クラスタインターコネクトの管理	200
クラスタインターコネクトでの動的再構成	201

▼ クラスタインターコネクトのステータスを確認する方法	202
▼ クラスタトランスポートケーブル、トランスポートアダプタ、トランスポートス イッチを追加する方法	203
▼ クラスタトランスポートケーブル、トランスポートアダプタ、トランスポートス イッチを削除する方法	205
▼ クラスタトランスポートケーブルを有効にする方法	208
▼ クラスタトランスポートケーブルを無効にする方法	210
▼ トランスポートアダプタのインスタンス番号を確認する方法	212
▼ 既存のクラスタのプライベートネットワークアドレスまたはアドレス範囲を変更 する方法	212
パブリックネットワークの管理	215
クラスタで IP ネットワークマルチパスグループを管理する方法	216
パブリックネットワークインタフェースでの動的再構成	217
8 ノードの追加と削除	219
クラスタへのノードの追加	219
▼ 既存のクラスタにノードを追加する方法	220
グローバルクラスタ内での非投票ノード (ゾーン) の作成	222
クラスタからのノードの削除	225
▼ ゾーンクラスタからノードを削除する方法	226
▼ クラスタソフトウェア構成からノードを削除する方法	227
▼ グローバルクラスタから非投票ノード (ゾーン) を削除する方法	230
▼ 2 ノード接続より大きなクラスタでアレイと単一ノード間の接続を削除する方 法	231
▼ エラーメッセージを修正する方法	233
9 クラスタの管理	235
クラスタの管理の概要	235
▼ クラスタ名を変更する方法	236
▼ ノード ID をノード名にマップする方法	238
▼ 新しいクラスタノード認証で作業する方法	238
▼ クラスタの時刻をリセットする方法	240
▼ SPARC: ノードで OpenBoot PROM (OBP) を表示する方法	243
▼ ノードのプライベートホスト名を変更する	243
▼ グローバルクラスタの非投票ノードにプライベートホスト名を追加する方法 ..	246
▼ グローバルクラスタの非投票ノードでプライベートホスト名を変更する方法 ..	246

▼ グローバルクラスタの非投票ノードのプライベートホスト名を削除する方法	248
▼ ノード名を変更する	249
▼ 既存の Oracle Solaris Cluster の論理ホスト名リソースで使用されている論理ホスト名を変更する	250
▼ ノードを保守状態にする	251
▼ ノードを保守状態から戻す	253
▼ クラスタノードから Oracle Solaris Cluster ソフトウェアをアンインストールする	255
ノードのアンインストールのトラブルシューティング	258
Oracle Solaris Cluster SNMP イベント MIB の作成、設定、および管理	259
負荷制限の構成	265
ゾーンクラスタ管理タスクの実行	267
▼ ゾーンクラスタにネットワークアドレスを追加する方法	269
▼ ゾーンクラスタを削除する	270
▼ ゾーンクラスタからファイルシステムを削除する	271
▼ ゾーンクラスタからストレージデバイスを削除する	273
トラブルシューティング	275
グローバルクラスタ外でのアプリケーションの実行	275
破損したディスクセットの復元	277
10 CPU 使用率の制御の構成	281
CPU 制御の概要	281
シナリオの選択	281
公平配分スケジューラ	282
CPU 制御の構成	283
▼ グローバルクラスタの投票ノードで CPU 使用率を制御する方法	283
▼ デフォルトプロセッサセットを使用してグローバルクラスタの非投票ノードで CPU 使用率を制御する方法	285
▼ 専用プロセッサセットを使用してグローバルクラスタの非投票ノードで CPU 使用率を制御する方法	288
11 Oracle Solaris Cluster ソフトウェアおよびファームウェアのパッチ適用	293
Oracle Solaris Cluster パッチ適用の概要	293
Oracle Solaris Cluster パッチのヒント	294
Oracle Solaris Cluster ソフトウェアのパッチ適用	295
▼ リブートするパッチを適用する方法 (ノード)	296

▼リブートするパッチを適用する方法 (クラスタ)	300
▼リブートしない Oracle Solaris Cluster パッチを適用する方法	304
▼フェイルオーバーゾーンを使用するノードにシングルユーザーモードでパッチを適用する方法	305
Oracle Solaris Cluster パッチの変更	309
12 クラスタのバックアップと復元	311
クラスタのバックアップ	311
▼バックアップするファイルシステム名を検索する方法	312
▼完全バックアップに必要なテープ数を調べる方法	312
▼ルート (/) ファイルシステムをバックアップする方法	313
▼ミラーのオンラインバックアップを実行する方法 (Solaris Volume Manager)	316
▼クラスタ構成をバックアップする方法	319
クラスタファイルの復元	319
▼個々のファイルを対話式に復元する方法 (Solaris Volume Manager)	320
▼ルート (/) ファイルシステムを復元する方法 (Solaris Volume Manager)	320
▼Solaris Volume Manager ボリューム上にあったルート (/) ファイルシステムを復元する方法	323
13 グラフィカルユーザーインターフェースによる Oracle Solaris Cluster の管理	329
Oracle Solaris Cluster Manager の概要	329
Oracle Solaris Cluster Manager の構成	330
RBAC の役割の設定	331
▼共通エージェントコンテナを使用して、サービスまたは管理エージェントのポート番号を変更する	331
▼Oracle Solaris Cluster Manager のサーバーアドレスを変更する方法	332
▼共通エージェントコンテナのセキュリティー鍵を再生成する方法	332
Oracle Solaris Cluster Manager ソフトウェアの起動	333
▼Oracle Solaris Cluster Manager を起動する方法	334
A 例	337
Availability Suite ソフトウェアを使用したホストベースのデータ複製の構成	337
クラスタにおける Availability Suite ソフトウェアの理解	338
クラスタ間でホストベースのデータ複製を構成するためのガイドライン	341
タスクマップ: データ複製の構成例	347

クラスタの接続とインストール	348
デバイスグループとリソースグループの構成例	350
データ複製の有効化例	364
データ複製の実行例	367
テイクオーバーの管理の例	373
索引	377

はじめに

『Oracle Solaris Cluster システム管理』では、SPARC および x86 ベースのシステムで Oracle Solaris Cluster の構成を管理する手順について説明します。

注 - この Oracle Solaris Cluster リリースは、SPARC および x86 ファミリーのプロセッサアーキテクチャー (UltraSPARC、SPARC64、AMD64、および Intel 64) を使用するシステムをサポートしています。このドキュメントでは、x86 とは、64 ビットの x86 互換製品の広範囲なファミリーを指します。このドキュメントの情報では、特に明示されている場合以外はすべてのプラットフォームに関係します。

このドキュメントは、Oracle のソフトウェアとハードウェアについて幅広い知識を持っている上級システム管理者を対象としています。このドキュメントは、計画や販売前のガイドとしては使用しないでください。

このドキュメントの説明は、Oracle Solaris オペレーティングシステムに関する知識と、Oracle Solaris Cluster とともに使用するボリューム管理ソフトウェアに関する専門知識を前提としています。

UNIX コマンドの使用

このドキュメントには、Oracle Solaris Cluster 構成の管理に特有のコマンドに関する情報が含まれています。このドキュメントには、UNIX の基本的なコマンドや手順について、詳細な説明が含まれていない場合もあります。

この情報については、次のうち 1 つ以上を参照してください。

- Oracle Solaris ソフトウェアのオンラインドキュメント
- システムに付属するその他のソフトウェアドキュメント
- Oracle Solaris オペレーティングシステム (Oracle Solaris OS) のマニュアルページ

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P-1 表記上の規則

字体または記号	説明	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% you have mail.
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	machine_name% su Password:
aabbcc123	プレースホルダー: 実際に使用する特定の 名前または値で置き換えます。	ファイルを削除するには、rm filename と入力します。
AaBbCc123	参照する書名を示します。	『ユーザーズガイド』の第 6 章 を参照ください。 キャッシュはローカルに保存さ れているコピーです。 ファイルを保存しないでくださ い。 注: 強調表示されたいくつかの項 目はオンラインで太字で表示さ れます。

コマンド例のシェルプロンプト

次の表は、Oracle Solaris OS に含まれるシェルの UNIX システムプロンプトとスーパーユーザーのプロンプトを示しています。コマンドの例では、シェルプロンプトは、コマンドを通常のユーザーと権限を持つユーザーのどちらで実行するかを示します。

表 P-2 シェルプロンプトについて

シェル	プロンプト
Bash シェル、Korn シェル、および Bourne シェル	\$

表 P-2 シェルプロンプトについて (続き)

シェル	プロンプト
スーパーユーザーの Bash シェル、Korn シェル、および Bourne シェル	#
C シェル	machine_name%
C シェルのスーパーユーザー	machine_name#

関連ドキュメント

Oracle Solaris Cluster の関連トピックについての情報は、次の表に示すドキュメントで参照できます。Oracle Solaris Cluster のすべてのドキュメントは、<http://www.oracle.com/technetwork/indexes/documentation/index.html> で入手可能です。

項目	ドキュメント
概念	『Oracle Solaris Cluster Concepts Guide』
ハードウェアの設置と管理	『Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual』 および個々のハードウェア管理ガイド
ソフトウェアのインストール	『Oracle Solaris Cluster ソフトウェアのインストール』
データサービスのインストールと管理	『Oracle Solaris Cluster Data Services Planning and Administration Guide』 および個々のデータサービスガイド
データサービスの開発	『Oracle Solaris Cluster Data Services Developer's Guide』
システム管理	『Oracle Solaris Cluster システム管理』 『Oracle Solaris Cluster Quick Reference』
ソフトウェアアップグレード	『Oracle Solaris Cluster Upgrade Guide』
エラーメッセージ	『Oracle Solaris Cluster Error Messages Guide』
コマンドと関数のリファレンス	『Oracle Solaris Cluster Reference Manual』 『Oracle Solaris Cluster Data Services Reference Manual』

Oracle Support へのアクセス

Oracle のお客様は、My Oracle Support を通じて電子的なサポートにアクセスできます。詳細は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> (聴覚に障害をお持ちの場合は <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>) を参照してください。

問い合わせについて

Oracle Solaris Cluster のインストールや使用について問題がある場合は、サービスプロバイダに連絡してください。サービスプロバイダには次の情報を提供してください。

- 名前と電子メールアドレス
- 会社名、住所、および電話番号
- システムの型番とシリアル番号
- オペレーティングシステムのリリース番号 (例: Oracle Solaris 10)
- Oracle Solaris Cluster のリリース番号 (例: Oracle Solaris Cluster 3.3)

次のコマンドを使用し、システムに関して、サービスプロバイダに必要な情報を収集してください。

コマンド	機能
<code>prtconf -v</code>	システムメモリのサイズと周辺デバイス情報を表示します
<code>psrinfo -v</code>	プロセッサの情報を表示する
<code>showrev -p</code>	インストールされているパッチを報告する
<code>SPARC: prtdiag -v</code>	システム診断情報を表示する
<code>/usr/cluster/bin/clnode show-rev -v</code>	Oracle Solaris Cluster のリリースおよびパッケージのバージョン情報を表示する

また、`/var/adm/messages` ファイルの内容もご用意ください。

Oracle Solaris Cluster の管理の概要

この章では、グローバルクラスタとゾーンクラスタの管理に関する次の情報と、Oracle Solaris Cluster 管理ツールの使用手順について説明します。

- 16 ページの「[Oracle Solaris Cluster の管理の概要](#)」
- 17 ページの「[Oracle Solaris OS の機能制限](#)」
- 18 ページの「[管理ツール](#)」
- 20 ページの「[クラスタ管理の準備](#)」
- 21 ページの「[クラスタ管理の開始](#)」

このガイドのすべての手順は、Oracle Solaris 10 オペレーティングシステムで使用するためのものです。

グローバルクラスタは、1つ以上のグローバルクラスタノードから構成されます。グローバルクラスタには、ノードではなく、HA for Zones データサービスで構成された native および cluster ブランドの非大域ゾーンを含めることもできます。ゾーンクラスタに関する一般的な情報については、『[Oracle Solaris Cluster Concepts Guide](#)』を参照してください。

ゾーンクラスタは、cluster ブランドの1つ以上の非大域ゾーンで構成されます。clzonecluster コマンドまたは clsetup ユーティリティを使用して、ゾーンクラスタを作成します。Oracle Solaris ゾーンで提供される分離を含めて、グローバルクラスタと同様にゾーンクラスタでサポートされるサービスを実行できます。ゾーンクラスタは、グローバルクラスタに依存しており、したがって、グローバルクラスタを必要とします。グローバルクラスタはゾーンクラスタを含みません。ゾーンクラスタは1つのマシン上に最大で1つのゾーンクラスタノードを持ちます。ゾーンクラスタノードは、同じマシン上のグローバルクラスタノードが動作している場合にのみ動作します。あるマシンのグローバルクラスタノードで障害が発生すると、そのマシン上のすべてのゾーンクラスタノードも動作しなくなります。ゾーンクラスタに関する一般的な情報については、『[Oracle Solaris Cluster Concepts Guide](#)』を参照してください。

Oracle Solaris Cluster の管理の概要

Oracle Solaris Cluster の高可用性環境によって、重要なアプリケーションの可用性がエンドユーザーに対して保証されます。システム管理者の業務は、Oracle Solaris Cluster 構成の安定した動作を保証することです。

管理タスクを始める前に、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』および『[Oracle Solaris Cluster Concepts Guide](#)』の計画情報をよく理解しておいてください。ゾーンクラスタの作成手順については、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』の「[ゾーンクラスタの構成](#)」を参照してください。Oracle Solaris Cluster の管理は、次のタスクごとに各マニュアルにまとめられています。

- グローバルクラスタまたはゾーンクラスタを定期的に (または毎日) 管理および維持するための標準的なタスク。これらのタスクは、このマニュアルで説明されています。
- インストール、構成、属性の変更などのデータサービスタスク。これらのタスクは、『[Oracle Solaris Cluster Data Services Planning and Administration Guide](#)』で説明されています。
- 記憶装置やネットワークハードウェアの追加や保守などのサービスタスク。これらのタスクは、『[Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual](#)』で説明されています。

一般に、クラスタが動作可能なときに Oracle Solaris Cluster 管理タスクを実行できます。クラスタからノードを取り外す必要がある場合、あるいはノードを停止する必要がある場合でも、残りのノードがクラスタを稼働している間に作業を行うことができます。明記されていないかぎり、Oracle Solaris Cluster の管理タスクはグローバルクラスタの投票ノードで行うべきです。クラスタ全体を停止する必要がある手順については、ダウタイムのスケジュールを通常の業務時間外に設定してシステムへの影響を最小限に抑えてください。クラスタまたはクラスタノードを停止する予定があるときは、あらかじめユーザーに通知しておいてください。

ゾーンクラスタに関する作業

Oracle Solaris Cluster の 2 つの管理コマンド (`cluster` および `clnode`) は、ゾーンクラスタでも実行できます。ただし、このコマンドの対象は、コマンドが発行されたゾーンクラスタに限定されます。たとえば、グローバルクラスタの投票ノードで `cluster` コマンドを使用すると、投票グローバルクラスタおよびすべてのゾーンクラスタに関するすべての情報が得られます。`cluster` コマンドをゾーンクラスタで使用すると、そのゾーンクラスタのみの情報が得られます。

`clzonecluster` コマンドを投票ノードで使用すると、グローバルクラスタ内のすべてのゾーンクラスタが対象になります。ゾーンクラスタコマンドはゾーンクラスタ上のすべてのノードを対象とします (コマンド発行時に停止していたノードも対象になります)。

ゾーンクラスタは、リソースグループマネージャ (Resource Group Manager、RGM) の制御下にあるリソースの委任管理をサポートしています。そのため、ゾーンクラスタの管理者は、クラスタ境界にまたがるゾーンクラスタ依存関係を表示できます (ただし、変更はできません)。クラスタ境界にまたがるゾーンクラスタ依存関係を作成し、変更し、削除できるのは、投票ノード内の管理者のみです。

次の表に、ゾーンクラスタで実行する主な管理タスクを示します。

- ゾーンクラスタの作成 – `clsetup` ユーティリティを使用して、ゾーンクラスタの構成ウィザードを起動するか、`clzonecluster install` コマンドを使用します。手順については、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』の「[ゾーンクラスタの構成](#)」を参照してください。
- ゾーンクラスタの起動と再起動 – [第3章「クラスタの停止とブート」](#)を参照してください。
- ゾーンクラスタへのノードの追加 – [第8章「ノードの追加と削除」](#)を参照してください。
- ゾーンクラスタからのノードの削除 – [226 ページの「ゾーンクラスタからノードを削除する方法」](#)を参照してください。
- ゾーンクラスタの構成の表示 – [35 ページの「クラスタ構成を表示する方法」](#)を参照してください。
- ゾーンクラスタの構成の検証 – [44 ページの「基本的なクラスタ構成を検証する方法」](#)を参照してください。
- ゾーンクラスタの停止 – [第3章「クラスタの停止とブート」](#)を参照してください。

Oracle Solaris OS の機能制限

Service Management Facility (SMF) 管理インタフェースを使用して、次の Oracle Solaris Cluster サービスを有効または無効にしないでください。

表 1-1 Oracle Solaris Cluster サービス

Oracle Solaris Cluster サービス	FMRI
<code>pnm</code>	<code>svc:/system/cluster/pnm:default</code>
<code>cl_event</code>	<code>svc:/system/cluster/cl_event:default</code>
<code>cl_eventlog</code>	<code>svc:/system/cluster/cl_eventlog:default</code>
<code>rpc_pmf</code>	<code>svc:/system/cluster/rpc_pmf:default</code>
<code>rpc_fed</code>	<code>svc:/system/cluster/rpc_fed:default</code>
<code>rgm</code>	<code>svc:/system/cluster/rgm:default</code>

表 1-1 Oracle Solaris Cluster サービス (続き)

Oracle Solaris Cluster サービス	FMRI
scdpm	svc:/system/cluster/scdpm:default
cl_ccra	svc:/system/cluster/cl_ccra:default
scsymon_srv	svc:/system/cluster/scsymon_srv:default
spm	svc:/system/cluster/spm:default
cl_svc_cluster_milestone	svc:/system/cluster/cl_svc_cluster_milestone:default
cl_svc_enable	svc:/system/cluster/cl_svc_enable:default
network-multipathing	svc:/system/cluster/network-multipathing

管理ツール

グラフィカルユーザーインターフェース (GUI) を使用するか、コマンド行を使用して、Oracle Solaris Cluster の構成に対する管理タスクを実行できます。次のセクションでは、GUI ツールとコマンド行ツールの概要を示します。

グラフィカルユーザーインターフェース

Oracle Solaris Cluster ソフトウェアは、クラスタに対してさまざまな管理タスクを実行するために使用できる GUI ツールをサポートしています。これらの GUI ツールが Oracle Solaris Cluster Manager です。詳細と、Oracle Solaris Cluster Manager を構成する手順については、[第 13 章「グラフィカルユーザーインターフェースによる Oracle Solaris Cluster の管理」](#)を参照してください。Oracle Solaris Cluster Manager の使用方法に関する具体的な情報については、オンラインヘルプで GUI について参照してください。

コマンド行インターフェース

Oracle Solaris Cluster のほとんどの管理タスクは、[clsetup\(1CL\)](#) ユーティリティを使用して対話形式で実行できます。可能なかぎり、本書の管理手順では clsetup ユーティリティを使用します。

clsetup ユーティリティを使用すると、「メイン」メニュー内の以下の項目を管理できます。

- 定足数 (quorum)
- リソースグループ
- データサービス

- クラスタインターコネクト
- デバイスグループとボリューム
- プライベートホスト名
- 新規ノード
- ゾーンクラスタ
- そのほかのクラスタタスク

次のリストに、Oracle Solaris Cluster 構成の管理に使用するその他のコマンドを示します。詳細は、マニュアルページを参照してください。

<code>ccp(1M)</code>	クラスタへのリモートコンソールアクセスを開始します。
<code>if_mpadm(1M)</code>	IP ネットワークマルチパスグループ内のあるアダプタから別のアダプタに IP アドレスを切り換えます。
<code>claccess(1CL)</code>	ノードを追加するために Oracle Solaris Cluster アクセスポリシーを管理します。
<code>cldevice(1CL)</code>	Oracle Solaris Cluster デバイスを管理します。
<code>cldevicegroup(1CL)</code>	Oracle Solaris Cluster デバイスグループを管理します。
<code>clinterconnect(1CL)</code>	Oracle Solaris Cluster インターコネクトを管理します。
<code>clnasdevice(1CL)</code>	Oracle Solaris Cluster 構成の NAS デバイスへのアクセスを管理します。
<code>clnode(1CL)</code>	Oracle Solaris Cluster ノードを管理します。
<code>clquorum(1CL)</code>	Oracle Solaris Cluster 定足数を管理します。
<code>clreslogicalhostname(1CL)</code>	論理ホスト名のために Oracle Solaris Cluster リソースを管理します。
<code>clresource(1CL)</code>	Oracle Solaris Cluster データサービスのリソースを管理します。
<code>clresourcegroup(1CL)</code>	Oracle Solaris Cluster データサービスのリソースを管理します。
<code>clresourcetype(1CL)</code>	Oracle Solaris Cluster データサービスのリソースを管理します。
<code>clressharedaddress(1CL)</code>	共有アドレスのために Oracle Solaris Cluster リソースを管理します。
<code>clsetup(1CL)</code>	Oracle Solaris Cluster クラスタを対話形式で構成し、ゾーンクラスタの作成と構成も行います。
<code>clsnmp(1CL)</code>	Oracle Solaris Cluster SNMP ホストを管理します。

<code>clsnmpmib(1CL)</code>	Oracle Solaris Cluster SNMP MIB を管理します。
<code>clsnmpuser(1CL)</code>	Oracle Solaris Cluster SNMP ユーザーを管理します。
<code>cltelemetryattribute(1CL)</code>	システムリソース監視を構成します。
<code>cluster(1CL)</code>	Oracle Solaris Cluster 構成のグローバル構成とグローバルステータスを管理します。
<code>clzonecluster(1CL)</code>	ゾーンクラスタの作成と変更を行います。

さらに、コマンドを使用して Oracle Solaris Cluster 構成のボリューム管理ソフトウェアを管理することもできます。これらのコマンドは、クラスタで使用されている特定のボリュームマネージャーに依存します。

クラスタ管理の準備

ここでは、クラスタの管理を準備する方法について説明します。

Oracle Solaris Cluster ハードウェア構成の記録

Oracle Solaris Cluster ハードウェア構成は時とともに変化していくので、サイトに固有なハードウェアの特徴は記録しておきます。クラスタを変更または更新するときには、このハードウェアの記録を参照することで管理作業を少なくすることができます。また、さまざまなクラスタコンポーネント間のケーブルや接続部にラベルを付けておくと、管理が簡単になります。

また、元のクラスタ構成とその後の変更を記録しておく、サン以外のサービスプロバイダがクラスタをサービスする時間を節約できます。

管理コンソールの使用

管理コンソールとして専用のワークステーション、または管理ネットワークを介して接続されたワークステーションを使用するとアクティブなクラスタを管理できます。クラスタコントロールパネル (CCP) とグラフィカルユーザーインターフェース (GUI) ツールは、通常、管理コンソールでインストールし、実行します。CCP の詳細は、[23 ページの「クラスタにリモートでログインする方法」](#)を参照してください。クラスタコントロールパネルモジュールと Oracle Solaris Cluster Manager GUI ツールをインストールする手順については、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』を参照してください。

管理コンソールはクラスタノードではありません。管理コンソールは、パブリックネットワークまたはネットワークベースの端末集配信装置(コンセントレータ)を通じてクラスタノードにリモートアクセスするために使用します。

Oracle Solaris Cluster には、専用の管理コンソールは必要ありませんが、専用コンソールを使用すると、次の利点が得られます。

- コンソールと管理ツールを同じマシンにまとめることで、クラスタ管理を一元化できます。
- システム管理者や保守担当者がすみやかに問題を解決できるようになる可能性があります。

クラスタのバックアップ

ご使用のクラスタを定期的にバックアップしてください。Oracle Solaris Cluster ソフトウェアは高可用性環境を備えており、データのミラー化されたコピーを記憶装置に保存していますが、これが定期的なバックアップの代わりになるとは考えないでください。Oracle Solaris Cluster 構成は、複数の障害に耐えることができますが、ユーザーやプログラムのエラー、または致命的な障害から保護する機能を備えていません。したがって、データ損失に対する保護のために、バックアップ手順を用意しておいてください。

次の情報もバックアップしてください。

- すべてのファイルシステムのパーティション
- DBMS データサービスを実行している場合は、すべてのデータベースのデータ
- すべてのクラスタディスクのディスクパーティション情報

クラスタ管理の開始

表 1-2 クラスタ管理の開始点を示します。

注 - グローバルクラスタの投票ノードからのみ実行する Oracle Solaris Cluster コマンドは、ゾーンクラスタには有効ではありません。各種ゾーンでのコマンドの有効な使用方法については、Oracle Solaris Cluster の該当するマニュアルページを参照してください。

表 1-2 Oracle Solaris Cluster 管理ツール

タスク	ツール	手順
クラスタへのリモートログイン	CCP コマンドを使用してクラスタコントロールパネル (CCP) を起動します。次に、 <code>cconsole</code> 、 <code>crlogin</code> 、 <code>cssh</code> 、または <code>ctelnet</code> のうち、いずれかのアイコンを選択します。	23 ページの「クラスタにリモートでログインする方法」 24 ページの「クラスタコンソールに安全に接続する方法」
対話形式でのクラスタの構成	<code>clzonecluster(1CL)</code> ユーティリティまたは <code>clsetup(1CL)</code> ユーティリティを起動します。	25 ページの「クラスタ構成ユーティリティにアクセスする方法」
Oracle Solaris Cluster のリリース番号とバージョン情報の表示	<code>clnode(1CL)</code> コマンドに <code>show-rev -v -node</code> サブコマンドとオプションを付けて使用します。	27 ページの「Oracle Solaris Cluster のリリース情報とバージョン情報を表示する方法」
インストールされているリソース、リソースグループ、リソースタイプの表示	リソース情報を表示するには、以下に示すコマンドを使用します。 <ul style="list-style-type: none"> ■ <code>clresource(1CL)</code> ■ <code>clresourcegroup(1CL)</code> ■ <code>clresourcetype(1CL)</code> 	30 ページの「構成されているリソースタイプ、リソースグループ、リソースを表示する方法」
クラスタコンポーネントのグラフィカルな監視	Oracle Solaris Cluster Manager を使用します。	オンラインヘルプを参照
一部のクラスタコンポーネントのグラフィカルな管理	SPARC ベースのシステムの Oracle Solaris Cluster でのみ使用できる Oracle Solaris Cluster Manager を使用します。	Oracle Solaris Cluster Manager については、オンラインヘルプを参照してください。
クラスタコンポーネントのステータスを確認します。	<code>cluster(1CL)</code> コマンドに <code>status</code> サブコマンドを付けて使用します。	31 ページの「クラスタコンポーネントのステータスを確認する方法」
パブリックネットワーク上の IP ネットワークマルチパスグループのステータス確認	グローバルクラスタの場合は、 <code>clnode(1CL) status</code> コマンドに <code>-m</code> オプションを付けて使用します。 ゾーンクラスタの場合は、 <code>clzonecluster(1CL) show</code> コマンドを使用します。	34 ページの「パブリックネットワークのステータスを確認する方法」

表 1-2 Oracle Solaris Cluster 管理ツール (続き)

タスク	ツール	手順
クラスタ構成を表示します。	グローバルクラスタの場合は、 <code>cluster(1CL)</code> コマンドに <code>show</code> サブコマンドを付けて使用します。 ゾーンクラスタの場合は、 <code>clzonecluster(1CL)</code> コマンドに <code>show</code> サブコマンドを付けて使用します。	35 ページの「クラスタ構成を表示する方法」
構成済み NAS デバイスの表示	グローバルクラスタまたはゾーンクラスタの場合は、 <code>clzonecluster(1CL)</code> コマンドに <code>show</code> サブコマンドを付けて使用します。	<code>clnasdevice(1CL)</code>
グローバルマウントポイントの確認またはクラスタ構成の検証	グローバルクラスタの場合は、 <code>cluster(1CL)cluster (1CL)</code> コマンドに <code>check</code> サブコマンドを付けて使用します。 ゾーンクラスタの場合は、 <code>clzonecluster(1CL) verify</code> コマンドを使用します。	44 ページの「基本的なクラスタ構成を検証する方法」
Oracle Solaris Cluster のコマンドログの内容の参照	<code>/var/cluster/logs/commandlog</code> ファイルを確認します。	51 ページの「Oracle Solaris Cluster のコマンドログの内容を表示する方法」
Oracle Solaris Cluster のシステムメッセージの参照	<code>/var/adm/messages</code> ファイルを確認します。	『Solaris のシステム管理 (上級編)』の「システムメッセージの表示」
Solaris Volume Manager のステータスの監視	<code>metastat</code> コマンドを使用します。	『Solaris Volume Manager 管理ガイド』

▼ クラスタにリモートでログインする方法

クラスタコントロールパネル (CCP) は、`cconsole`、`crlogin`、`cssh`、および `ctelnet` ツールの起動パッドです。すべてのツールが、一連の指定されたノードへの複数ウィンドウ接続を開始します。複数ウィンドウ接続は、指定されたノードごとのホストウィンドウと、共通ウィンドウから成ります。共通ウィンドウへの入力、それぞれのホストウィンドウに送信されるので、クラスタのすべてのノード上でコマンドを同時に実行できます。

コマンド行から `cconsole`、`crlogin`、`cssh`、または `ctelnet` セッションを開始することもできます。

デフォルトでは、`cconsole` ユーティリティはノードコンソールに対して `telnet` 接続を使用します。その代わりに、コンソールに対して Secure Shell 接続を確立するには、`cconsole` ウィンドウの「オプション」メニューの「SSH の使用」チェックボックスにチェックマークを付けます。または、`ccp` または `cconsole` コマンドを発行するときに `-s` オプションを指定します。

詳細については、[ccp\(1M\)](#) および [cconsole\(1M\)](#) のマニュアルページを参照してください。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

始める前に CCP を起動する前に次の前提条件が満たされていることを確認します。

- 管理コンソールに `SUNWcccon` パッケージをインストールします。
- 管理コンソールの `PATH` 変数に Oracle Solaris Cluster ツールのディレクトリである `/opt/SUNWcluster/bin` と `/usr/cluster/bin` が含まれていることを確認します。`$CLUSTER_HOME` 環境変数を設定すると、ツールディレクトリ用に代替の場所を指定できます。
- 端末集配信装置を使用している場合は、`clusters` ファイル、`serialports` ファイル、および `nsswitch.conf` ファイルを構成します。これらのファイルは、`/etc` ファイル、あるいは NIS または NIS+ データベースのいずれかでもかまいません。詳細については、[clusters\(4\)](#) および [serialports\(4\)](#) のマニュアルページを参照してください。

- 1 管理コンソールから CCP 起動パッドを起動します。

```
phys-schost# ccp clustername
```

CCP 起動パッドが表示されます。

- 2 クラスタとのリモートセッションを開始するには、CCP 起動パッドで「`cconsole`」アイコン、「`crlogin`」アイコン、「`cssh`」アイコン、または「`ctelnet`」アイコンをクリックします。

▼ クラスタコンソールに安全に接続する方法

この手順を実行し、クラスタノードのコンソールへの Secure Shell 接続を確立します。

始める前に 端末集配信装置を使用している場合は、clusters ファイル、serialports ファイル、および nsswitch.conf ファイルを構成します。これらのファイルは、/etc ファイル、あるいは NIS または NIS+ データベースのいずれかでもかまいません。

注-serialports ファイルで、各コンソールアクセスデバイスへのセキュア接続に使用するポート番号を割り当てます。Secure Shell 接続のデフォルトのポート番号は 22 です。

詳細については、clusters(4) および serialports(4) のマニュアルページを参照してください。

- 1 管理コンソールでスーパーユーザーになります。
- 2 セキュアモードで cconsole ユーティリティを起動します。

```
# cconsole -s [-l username] [-p ssh-port]
```

-s	Secure Shell 接続を有効にします。
-l username	リモート接続のユーザー名を指定します。-l オプションが指定されていない場合、ccconsole ユーティリティを起動したユーザー名が使用されます。
-p ssh-port	使用する Secure Shell ポート番号を指定します。-p オプションが指定されていない場合は、デフォルトのポート番号 22 がセキュア接続に使用されます。

▼ クラスタ構成ユーティリティにアクセスする方法

clsetup ユーティリティでは、ゾーンクラスタを対話形式で作成し、定足数、リソースグループ、クラスタトランスポート、プライベートホスト名、デバイスグループ、およびグローバルクラスタの新しいノードオプションを構成できます。clzonecluster ユーティリティは、同様な構成タスクをゾーンクラスタに対して実行します。詳細は、clsetup(1CL) と clzonecluster(1CL) のマニュアルページを参照してください。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 グローバルクラスタのアクティブなメンバーノードで、スーパーユーザーになります。グローバルクラスタのノードから、次の手順のステップをすべて実行します。

- 2 構成ユーティリティを起動します。

```
phys-schost# clsetup
```

- グローバルクラスタの場合は、**clsetup** コマンドでユーティリティを起動します。

```
phys-schost# clsetup
```

メインメニューが表示されます。

- ゾーンクラスタの場合は、**clzonecluster** コマンドでユーティリティを起動します。この例のゾーンクラスタは *sczone* です。

```
phys-schost# clzonecluster configure sczone
```

ユーティリティで実行可能な操作は、次のオプションで確認できます。

```
clzc:sczone> ?
```

対話型の **clsetup** ユーティリティを使用して、ゾーンクラスタを作成したり、ファイルシステムまたはストレージデバイスをクラスタスコープに追加することもできます。その他のすべてのゾーンクラスタ構成タスクは、**clzonecluster configure** コマンドで実行されます。**clsetup** ユーティリティの使用手順については、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』を参照してください。

- 3 使用する構成をメニューから選択します。画面に表示される指示に従って、タスクを完了します。詳細は、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』の「[ゾーンクラスタの構成](#)」に記述されている手順を参照してください。

参照 詳細は、**clsetup** または **clzonecluster** のオンラインヘルプを参照してください。

▼ Oracle Solaris Cluster のパッチ情報を表示する方法

この手順を実行するためにスーパーユーザーとしてログインする必要はありません。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- **Oracle Solaris Cluster** のパッチ情報を表示します。

```
phys-schost# showrev -p
```

Oracle Solaris Cluster アップデートリリースは、メイン製品のパッチ番号とアップデートバージョンによって識別されます。

例 1-1 Oracle Solaris Cluster のパッチ情報の表示

次の例は、パッチ 110648-05 に関する情報を示しています。

```
phys-schost# showrev -p | grep 110648
```

```
Patch: 110648-05 Obsoletes: Requires: Incompatibles: Packages:
```

▼ Oracle Solaris Cluster のリリース情報とバージョン情報を表示する方法

この手順を実行するためにスーパーユーザーとしてログインする必要はありません。グローバルクラスタのノードから、次の手順のステップをすべて実行します。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- **Oracle Solaris Cluster** のリリース情報およびバージョン情報の表示:

```
phys-schost# clnode show-rev -v -node
```

このコマンドは、すべての Oracle Solaris Cluster パッケージについて Oracle Solaris Cluster のリリース番号とバージョン文字列を表示します。

例 1-2 Oracle Solaris Cluster のリリース情報およびバージョン情報の表示

次の例は、クラスタのリリース情報とパッケージのバージョン情報を示しています。

```
phys-schost# clnode show-rev
```

```
3.3
```

```
phys-schost# clnode show-rev -v
```

```
Oracle Solaris Cluster 3.3 for Solaris 10 sparc
```

```
SUNWcccon: 3.3.0,REV=2010.06.14.03.44
```

```
SUNWcccon: 3.3.0,REV=2010.06.14.03.44
```

```
SUNWcsc: 3.3.0,REV=2010.06.14.03.44
```

```
SUNWcscspmu 3.3.0,REV=2010.06.14.03.44
```

```

SUNWscssv: 3.3.0,REV=2010.06.14.03.44
SUNWeccn: 3.3.0,REV=2010.06.14.03.44
SUNWesc: 3.3.0,REV=2010.06.14.03.44
SUNWescspmu: 3.3.0,REV=2010.06.14.03.44
SUNWescssv: 3.3.0,REV=2010.06.14.03.44
SUNWfccon: 3.3.0,REV=2010.06.14.03.44
SUNWfsc: 3.3.0,REV=2010.06.14.03.44
SUNWfscspmu: 3.3.0,REV=2010.06.14.03.44
SUNWfscssv: 3.3.0,REV=2010.06.14.03.44
SUNWjccon: 3.3.0,REV=2010.06.14.03.44
SUNWjcommonS: 3.3.0,REV=2010.06.14.03.44
SUNWjsc: 3.3.0,REV=2010.06.14.03.44
SUNWjscman: 3.3.0,REV=2010.06.14.03.44
SUNWjscspmu: 3.3.0,REV=2010.06.14.03.44
SUNWjscssv: 3.3.0,REV=2010.06.14.03.44
SUNWkccon: 3.3.0,REV=2010.06.14.03.44
SUNWksc: 3.3.0,REV=2010.06.14.03.44
SUNWkscspmu: 3.3.0,REV=2010.06.14.03.44
SUNWkscssv: 3.3.0,REV=2010.06.14.03.44
SUNWscu: 3.3.0,REV=2010.06.14.03.44
SUNWsccomu: 3.3.0,REV=2010.06.14.03.44
SUNWsczr: 3.3.0,REV=2010.06.14.03.44
SUNWscmzu: 3.3.0,REV=2010.06.14.03.44
SUNWsczu: 3.3.0,REV=2010.06.14.03.44
SUNWscsckr: 3.3.0,REV=2010.06.14.03.44
SUNWscscku: 3.3.0,REV=2010.06.14.03.44
SUNWscr: 3.3.0,REV=2010.06.14.03.44
SUNWscrdt: 3.3.0,REV=2010.06.14.03.44
SUNWscrif: 3.3.0,REV=2010.06.14.03.44
SUNWscrtlh: 3.3.0,REV=2010.06.14.03.44
SUNWscnmr: 3.3.0,REV=2010.06.14.03.44
SUNWscnmu: 3.3.0,REV=2010.06.14.03.44
SUNWscdev: 3.3.0,REV=2010.06.14.03.44
SUNWscgds: 3.3.0,REV=2010.06.14.03.44
SUNWscsmf: 3.3.0,REV=2010.06.14.03.44
SUNWscman: 3.3.0,REV=2010.05.21.18.40
SUNWscsal: 3.3.0,REV=2010.06.14.03.44
SUNWscsam: 3.3.0,REV=2010.06.14.03.44
SUNWscvm: 3.3.0,REV=2010.06.14.03.44
SUNWmdmr: 3.3.0,REV=2010.06.14.03.44
SUNWmdmu: 3.3.0,REV=2010.06.14.03.44
SUNWscmasa: 3.3.0,REV=2010.06.14.03.44
SUNWscmasar: 3.3.0,REV=2010.06.14.03.44
SUNWscmasasen: 3.3.0,REV=2010.06.14.03.44
SUNWscmasazu: 3.3.0,REV=2010.06.14.03.44
SUNWscmasau: 3.3.0,REV=2010.06.14.03.44
SUNWscmautil: 3.3.0,REV=2010.06.14.03.44
SUNWscmautilr: 3.3.0,REV=2010.06.14.03.44
SUNWjfreechart: 3.3.0,REV=2010.06.14.03.44
SUNWjfreechartS: 3.3.0,REV=2010.06.14.03.44
ORCLscPeopleSoft: 3.3.0,REV=2010.06.14.03.44
ORCLscobiee: 3.3.0,REV=2010.06.14.03.44
ORCLscoep: 3.3.0,REV=2010.06.14.03.44
ORCLscohs: 3.3.0,REV=2010.06.14.03.44
ORCLscopmn: 3.3.0,REV=2010.06.14.03.44
ORCLscsapnetw: 3.3.0,REV=2010.06.14.03.44
SUNWcvm: 3.3.0,REV=2010.06.14.03.44
SUNWcvmr: 3.3.0,REV=2010.06.14.03.44

```

SUNWiimsc: 3.3.0,REV=2010.06.14.03.44
 SUNWscspmr: 3.3.0,REV=2010.06.14.03.44
 SUNWscspmu: 3.3.0,REV=2010.06.14.03.44
 SUNWscderby: 3.3.0,REV=2010.06.14.03.44
 SUNWsc telemetry: 3.3.0,REV=2010.06.14.03.44
 SUNWscgrepavs: 3.2.3,REV=2009.10.23.12.12
 SUNWscgrepavsu: 3.3.0,REV=2010.06.14.03.44
 SUNWscgrepodg: 3.3.0,REV=2010.06.14.03.44
 SUNWscgrepodgu: 3.3.0,REV=2010.06.14.03.44
 SUNWscgrepsbpu: 3.3.0,REV=2010.06.14.03.44
 SUNWscgrepsrdf: 3.2.3,REV=2009.10.23.12.12
 SUNWscgrepsrdfu: 3.3.0,REV=2010.06.14.03.44
 SUNWscgreptc: 3.2.3,REV=2009.10.23.12.12
 SUNWscgreptcu: 3.3.0,REV=2010.06.14.03.44
 SUNWscgspm: 3.3.0,REV=2010.06.14.03.44
 SUNWscghb: 3.2.3,REV=2009.10.23.12.12
 SUNWscghbr: 3.3.0,REV=2010.06.14.03.44
 SUNWscgman: 3.3.0,REV=2010.06.14.03.44
 ORCLscgrepszfssa: 3.3.0,REV=2010.06.14.03.44
 SUNWscgctl: 3.2.3,REV=2009.10.23.12.12
 SUNWscgctlr: 3.3.0,REV=2010.06.14.03.44
 SUNWscims: 6.0,REV=2003.10.29
 SUNWscics: 6.0,REV=2003.11.14
 SUNWscids: 3.3.0,REV=2010.06.14.03.44
 SUNWscapc: 3.2.0,REV=2006.12.06.18.32
 SUNWscdns: 3.2.0,REV=2006.12.06.18.32
 SUNWschadb: 3.2.0,REV=2006.12.06.18.32
 SUNWschtt: 3.2.0,REV=2006.12.06.18.32
 SUNWscslas: 3.2.0,REV=2006.12.06.18.32
 SUNWscsrb5: 3.2.0,REV=2006.12.06.18.32
 SUNWscnfs: 3.2.0,REV=2006.12.06.18.32
 SUNWscor: 3.2.0,REV=2006.12.06.18.32
 SUNWscpax: 3.3.0,REV=2010.06.14.03.44
 SUNWscslmq: 3.2.0,REV=2006.12.06.18.32
 SUNWscsap: 3.2.0,REV=2006.12.06.18.32
 SUNWscslc: 3.2.0,REV=2006.12.06.18.32
 SUNWscmd: 3.3.0,REV=2010.06.14.03.44
 SUNWscsapdb: 3.2.0,REV=2006.12.06.18.32
 SUNWscsapenq: 3.2.0,REV=2006.12.06.18.32
 SUNWscsaprepl: 3.2.0,REV=2006.12.06.18.32
 SUNWscsapscs: 3.2.0,REV=2006.12.06.18.32
 SUNWscsapwebas: 3.2.0,REV=2006.12.06.18.32
 SUNWscsbl: 3.2.0,REV=2006.12.06.18.32
 SUNWscsyb: 3.2.0,REV=2006.12.06.18.32
 SUNWscucm: 3.3.0,REV=2010.06.14.03.44
 SUNWscwls: 3.3.0,REV=2010.06.14.03.44
 SUNWudlm: 3.3.0,REV=2010.06.14.03.44
 SUNWudlmr: 3.3.0,REV=2010.06.14.03.44
 SUNWscwls: 3.2.0,REV=2006.12.06.18.32
 SUNWsc9ias: 3.2.0,REV=2006.12.06.18.32
 SUNWscPostgreSQL: 3.2.0,REV=2006.12.06.18.32
 SUNWscTimesTen: 3.3.0,REV=2010.06.14.03.44
 SUNWsczone: 3.2.0,REV=2006.12.06.18.32
 SUNWscdhc: 3.2.0,REV=2006.12.06.18.32
 SUNWscsbs: 3.2.0,REV=2006.12.06.18.32
 SUNWscmqi: 3.2.0,REV=2006.12.06.18.32
 SUNWscmqs: 3.2.0,REV=2006.12.06.18.32
 SUNWscmys: 3.2.0,REV=2006.12.06.18.32

```
SUNWscsge:      3.2.0,REV=2006.12.06.18.32
SUNWscsaa:      3.2.0,REV=2006.12.06.18.32
SUNWscsag:      3.2.0,REV=2006.12.06.18.32
SUNWscsmb:      3.2.0,REV=2006.12.06.18.32
SUNWscsps:      3.2.0,REV=2006.12.06.18.32
SUNWsctomcat:   3.2.0,REV=2006.12.06.18.32
```

▼ 構成されているリソースタイプ、リソースグループ、リソースを表示する方法

この手順は、Oracle Solaris Cluster Manager GUI を使用しても実行できます。詳細については、[第 13 章「グラフィカルユーザーインターフェースによる Oracle Solaris Cluster の管理」](#)を参照するか、Oracle Solaris Cluster Manager オンラインヘルプを参照してください。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

始める前に スーパーユーザー以外のユーザーがこのサブコマンドを使用するには、RBAC の承認 `solaris.cluster.read` が必要です。

- クラスタで構成されているリソースタイプ、リソースグループ、リソースを表示します。グローバルクラスタのノードから、次の手順のステップをすべて実行します。

```
phys-schost# cluster show -t resource,resourcetype,resourcegroup
```

個別のリソース、リソースグループ、およびリソースタイプの詳細については、次のいずれかのコマンドとともに `show` サブコマンドを使用します。

- `resource`
- `resource group`
- `resourcetype`

例 1-3 構成されているリソースタイプ、リソースグループ、リソースの表示

次に、クラスタ `schost` に対して構成されているリソースタイプ (RT Name)、リソースグループ (RG Name)、リソース (RS Name) の例を示します。

```
phys-schost# cluster show -t resource,resourcetype,resourcegroup
```

```
=== Registered Resource Types ===
```

```
Resource Type:                SUNW.qfs
RT_description:                SAM-QFS Agent on Oracle Solaris Cluster
RT_version:                    3.1
API_version:                   3
RT_basedir:                    /opt/SUNWsamfs/sc/bin
Single_instance:               False
Proxy:                         False
Init_nodes:                    All potential masters
Installed_nodes:               <All>
Failover:                      True
Pkglist:                       <NULL>
RT_system:                     False
Global_zone:                   True
```

```
=== Resource Groups and Resources ===
```

```
Resource Group:                qfs-rg
RG_description:                <NULL>
RG_mode:                       Failover
RG_state:                      Managed
Failback:                      False
Nodelist:                      phys-schost-2 phys-schost-1
```

```
--- Resources for Group qfs-rg ---
```

```
Resource:                      qfs-res
Type:                          SUNW.qfs
Type_version:                  3.1
Group:                         qfs-rg
R_description:
Resource_project_name:         default
Enabled{phys-schost-2}:        True
Enabled{phys-schost-1}:        True
Monitored{phys-schost-2}:      True
Monitored{phys-schost-1}:      True
```

▼ クラスタコンポーネントのステータスを確認する方法

この手順は、Oracle Solaris Cluster Manager GUI を使用しても実行できます。詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

注 - cluster status コマンドでもゾーンクラスタのステータスが表示されます。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

始める前に スーパーユーザー以外のユーザーが `status` サブコマンドを使用するには、RBAC の承認 `solaris.cluster.read` が必要です。

- クラスタコンポーネントのステータスを確認します。グローバルクラスタのノードから、次の手順のステップをすべて実行します。

```
phys-schost# cluster status
```

例 1-4 クラスタコンポーネントのステータス確認

次に、`cluster(1CL) status` によって返されるクラスタコンポーネントのステータス情報の例を示します。

```
phys-schost# cluster status
=== Cluster Nodes ===

--- Node Status ---

Node Name                               Status
-----
phys-schost-1                           Online
phys-schost-2                           Online

=== Cluster Transport Paths ===

Endpoint1                               Endpoint2                               Status
-----
phys-schost-1:qfe1                      phys-schost-4:qfe1                      Path online
phys-schost-1:hme1                      phys-schost-4:hme1                      Path online

=== Cluster Quorum ===

--- Quorum Votes Summary ---

          Needed   Present   Possible
          -----
          3         3         4

--- Quorum Votes by Node ---

Node Name           Present   Possible   Status
-----
phys-schost-1       1         1         Online
phys-schost-2       1         1         Online
```


--- Quorum Votes by Device ---

Device Name	Present	Possible	Status
/dev/did/rdisk/d2s2	1	1	Online
/dev/did/rdisk/d8s2	0	1	Offline

=== Cluster Device Groups ===

--- Device Group Status ---

Device Group Name	Primary	Secondary	Status
schost-2	phys-schost-2	-	Degraded

--- Spare, Inactive, and In Transition Nodes ---

Device Group Name	Spare Nodes	Inactive Nodes	In Transition Nodes
schost-2	-	-	-

=== Cluster Resource Groups ===

Group Name	Node Name	Suspended	Status
test-rg	phys-schost-1	No	Offline
	phys-schost-2	No	Online
test-rg	phys-schost-1	No	Offline
	phys-schost-2	No	Error--stop failed
test-rg	phys-schost-1	No	Online
	phys-schost-2	No	Online

=== Cluster Resources ===

Resource Name	Node Name	Status	Message
test_1	phys-schost-1	Offline	Offline
	phys-schost-2	Online	Online
test_1	phys-schost-1	Offline	Offline
	phys-schost-2	Stop failed	Faulted
test_1	phys-schost-1	Online	Online
	phys-schost-2	Online	Online

Device Instance	Node	Status
/dev/did/rdisk/d2	phys-schost-1	Ok
/dev/did/rdisk/d3	phys-schost-1	Ok
	phys-schost-2	Ok

```
/dev/did/rdsk/d4      phys-schost-1      Ok
                      phys-schost-2      Ok

/dev/did/rdsk/d6      phys-schost-2      Ok

=== Zone Clusters ===

--- Zone Cluster Status ---

Name      Node Name  Zone HostName  Status  Zone Status
-----
sczone    schost-1   sczone-1      Online  Running
          schost-2   sczone-2      Online  Running
```

▼ パブリックネットワークのステータスを確認する方法

この手順は、Oracle Solaris Cluster Manager GUI を使用しても実行できます。詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

IP ネットワークマルチパスグループのステータスを確認するには、**clnode(1CL)** コマンドに **status** サブコマンドを付けて使用します。

始める前に スーパーユーザー以外のユーザーがこのサブコマンドを使用するには、RBAC の承認 **solaris.cluster.read** が必要です。

- クラスタコンポーネントのステータスを確認します。グローバルクラスタのノードから、次の手順のステップをすべて実行します。

```
phys-schost# clnode status -m
```

例 1-5 パブリックネットワークのステータスを調べる

次に、**clnode status** コマンドによって返されるクラスタコンポーネントのステータス情報の例を示します。

```
% clnode status -m
--- Node IPMP Group Status ---

Node Name      Group Name    Status    Adapter    Status
-----
phys-schost-1  test-rg      Online    qfe1       Online
phys-schost-2  test-rg      Online    qfe1       Online
```

▼ クラスタ構成を表示する方法

この手順は、Oracle Solaris Cluster Manager GUI を使用しても実行できます。詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

始める前に スーパーユーザー以外のユーザーが `status` サブコマンドを使用するには、RBAC の承認 `solaris.cluster.read` が必要です。

- グローバルクラスタまたはゾーンクラスタの構成を表示します。グローバルクラスタのノードから、次の手順のステップをすべて実行します。

```
% cluster show
```

`cluster show` コマンドをグローバルクラスタの投票ノードから実行すると、そのクラスタに関する詳細な構成情報が表示されます。また、ゾーンクラスタが構成されていれば、ゾーンクラスタの情報も表示されます。

ゾーンクラスタのみに関する構成情報を表示するには、`clzonecluster show` コマンドも使用できます。ゾーンクラスタのプロパティには、ゾーンクラスタ名、IP タイプ、自動ブート、ゾーンパスなどがあります。`show` サブコマンドは、ゾーンクラスタの内部で実行され、そのゾーンクラスタのみが対象になります。ゾーンクラスタノードから `clzonecluster show` コマンドを実行すると、そのゾーンクラスタから認識可能なオブジェクトのみのステータスが得られます。

`cluster` コマンドでより多くの情報を表示するには、冗長オプションを使用します。詳細は、[cluster\(1CL\)](#) のマニュアルページを参照してください。`clzonecluster` の詳細は、[clzonecluster\(1CL\)](#) のマニュアルページを参照してください。

例1-6 グローバルクラスタの構成を表示する

次に、グローバルクラスタの構成情報の例を示します。ゾーンクラスタが構成されている場合は、ゾーンクラスタの情報も表示されます。

```
phys-schost# cluster show
```

```
=== Cluster ===
```

```
Cluster Name:                cluster-1
clusterid:                   0x50C000C4
installmode:                 disabled
heartbeat_timeout:           10000
heartbeat_quantum:           1000
private_netaddr:             172.16.0.0
private_netmask:             255.255.248.0
max_nodes:                   62
num_zoneclusters:            1
max_privatenets:             10
global_fencing:              pathcount
Node List:                   phys-schost-1
Node Zones:                  phys_schost-2:za
```

```
=== Host Access Control ===
```

```
Cluster name:                clustser-1
Allowed hosts:               phys-schost-1, phys-schost-2:za
Authentication Protocol:     sys
```

```
=== Cluster Nodes ===
```

```
Node Name:                   phys-schost-1
Node ID:                     1
Type:                        cluster
Enabled:                     yes
privatehostname:             clusternode1-priv
reboot_on_path_failure:     disabled
globalzoneshares:           3
defaultpsetmin:             1
quorum_vote:                 1
quorum_defaultvote:         1
quorum_resv_key:             0x43CB1E1800000001
Transport Adapter List:      qfe3, hme0
```

```
--- Transport Adapters for phys-schost-1 ---
```

```
Transport Adapter:           qfe3
Adapter State:               Enabled
Adapter Transport Type:      dlpi
Adapter Property(device_name): qfe
Adapter Property(device_instance): 3
Adapter Property(lazy_free): 1
Adapter Property(dlpi_heartbeat_timeout): 10000
Adapter Property(dlpi_heartbeat_quantum): 1000
Adapter Property(nw_bandwidth): 80
Adapter Property(bandwidth): 10
```

```

Adapter Property(ip_address):          172.16.1.1
Adapter Property(netmask):             255.255.255.128
Adapter Port Names:                    0
Adapter Port State(0):                 Enabled

Transport Adapter:                     hme0
Adapter State:                         Enabled
Adapter Transport Type:                dlpi
Adapter Property(device_name):         hme
Adapter Property(device_instance):     0
Adapter Property(lazy_free):           0
Adapter Property(dlpi_heartbeat_timeout): 10000
Adapter Property(dlpi_heartbeat_quantum): 1000
Adapter Property(nw_bandwidth):        80
Adapter Property(bandwidth):           10
Adapter Property(ip_address):          172.16.0.129
Adapter Property(netmask):             255.255.255.128
Adapter Port Names:                    0
Adapter Port State(0):                 Enabled

--- SNMP MIB Configuration on phys-schost-1 ---

SNMP MIB Name:                        Event
State:                               Disabled
Protocol:                            SNMPv2

--- SNMP Host Configuration on phys-schost-1 ---

--- SNMP User Configuration on phys-schost-1 ---

SNMP User Name:                       foo
Authentication Protocol:              MD5
Default User:                         No

Node Name:                            phys-schost-2:za
Node ID:                              2
Type:                                 cluster
Enabled:                              yes
privatehostname:                      clusternode2-priv
reboot_on_path_failure:               disabled
globalzoneshares:                     1
defaulttpsetmin:                       2
quorum_vote:                           1
quorum_defaultvote:                   1
quorum_resv_key:                       0x43CB1E1800000002
Transport Adapter List:                hme0, qfe3

--- Transport Adapters for phys-schost-2 ---

Transport Adapter:                     hme0
Adapter State:                         Enabled
Adapter Transport Type:                dlpi
Adapter Property(device_name):         hme
Adapter Property(device_instance):     0
Adapter Property(lazy_free):           0
Adapter Property(dlpi_heartbeat_timeout): 10000
Adapter Property(dlpi_heartbeat_quantum): 1000
Adapter Property(nw_bandwidth):        80

```

```

Adapter Property(bandwidth):          10
Adapter Property(ip_address):         172.16.0.130
Adapter Property(netmask):            255.255.255.128
Adapter Port Names:                   0
Adapter Port State(0):                Enabled

Transport Adapter:                    qfe3
Adapter State:                        Enabled
Adapter Transport Type:               dlpi
Adapter Property(device_name):        qfe
Adapter Property(device_instance):    3
Adapter Property(lazy_free):          1
Adapter Property(dlpi_heartbeat_timeout): 10000
Adapter Property(dlpi_heartbeat_quantum): 1000
Adapter Property(nw_bandwidth):       80
Adapter Property(bandwidth):          10
Adapter Property(ip_address):         172.16.1.2
Adapter Property(netmask):            255.255.255.128
Adapter Port Names:                   0
Adapter Port State(0):                Enabled

--- SNMP MIB Configuration on phys-schost-2 ---

SNMP MIB Name:                        Event
State:                                Disabled
Protocol:                             SNMPv2

--- SNMP Host Configuration on phys-schost-2 ---

--- SNMP User Configuration on phys-schost-2 ---

=== Transport Cables ===

Transport Cable:                      phys-schost-1:qfe3,switch2@1
Cable Endpoint1:                     phys-schost-1:qfe3
Cable Endpoint2:                     switch2@1
Cable State:                         Enabled

Transport Cable:                      phys-schost-1:hme0,switch1@1
Cable Endpoint1:                     phys-schost-1:hme0
Cable Endpoint2:                     switch1@1
Cable State:                         Enabled

Transport Cable:                      phys-schost-2:hme0,switch1@2
Cable Endpoint1:                     phys-schost-2:hme0
Cable Endpoint2:                     switch1@2
Cable State:                         Enabled

Transport Cable:                      phys-schost-2:qfe3,switch2@2
Cable Endpoint1:                     phys-schost-2:qfe3
Cable Endpoint2:                     switch2@2
Cable State:                         Enabled

=== Transport Switches ===

Transport Switch:                     switch2
Switch State:                         Enabled
Switch Type:                          switch

```

```

Switch Port Names:                1 2
Switch Port State(1):             Enabled
Switch Port State(2):             Enabled

Transport Switch:                  switch1
Switch State:                      Enabled
Switch Type:                       switch
Switch Port Names:                 1 2
Switch Port State(1):              Enabled
Switch Port State(2):              Enabled

=== Quorum Devices ===

Quorum Device Name:                d3
Enabled:                           yes
Votes:                             1
Global Name:                       /dev/did/rdisk/d3s2
Type:                              scsi
Access Mode:                       scsi2
Hosts (enabled):                   phys-schost-1, phys-schost-2

Quorum Device Name:                qs1
Enabled:                           yes
Votes:                             1
Global Name:                       qs1
Type:                              quorum_server
Hosts (enabled):                   phys-schost-1, phys-schost-2
Quorum Server Host:                10.11.114.83
Port:                              9000

=== Device Groups ===

Device Group Name:                  testdg3
Type:                              SVM
failback:                          no
Node List:                         phys-schost-1, phys-schost-2
preferenced:                       yes
numsecondaries:                    1
diskset name:                      testdg3

=== Registered Resource Types ===

Resource Type:                     SUNW.LogicalHostname:2
RT_description:                    Logical Hostname Resource Type
RT_version:                        2
API_version:                       2
RT_basedir:                        /usr/cluster/lib/rgm/rt/hafoip
Single_instance:                   False
Proxy:                             False
Init_nodes:                        All potential masters
Installed_nodes:                   <All>
Failover:                          True
Pkglist:                           SUNWscu
RT_system:                         True

Resource Type:                     SUNW.SharedAddress:2

```

```

RT_description:      HA Shared Address Resource Type
RT_version:          2
API_version:         2
RT_basedir:          /usr/cluster/lib/rgm/rt/hascip
Single_instance:     False
Proxy:               False
Init_nodes:          <Unknown>
Installed_nodes:     <All>
Failover:            True
Pkglist:              SUNWscu
RT_system:           True

```

```

Resource Type:       SUNW.HAStoragePlus:4
RT_description:      HA Storage Plus
RT_version:          4
API_version:         2
RT_basedir:          /usr/cluster/lib/rgm/rt/hastorageplus
Single_instance:     False
Proxy:               False
Init_nodes:          All potential masters
Installed_nodes:     <All>
Failover:            False
Pkglist:              SUNWscu
RT_system:           False

```

```

Resource Type:       SUNW.haderby
RT_description:      haderby server for Oracle Solaris Cluster
RT_version:          1
API_version:         7
RT_basedir:          /usr/cluster/lib/rgm/rt/haderby
Single_instance:     False
Proxy:               False
Init_nodes:          All potential masters
Installed_nodes:     <All>
Failover:            False
Pkglist:              SUNWscderby
RT_system:           False

```

```

Resource Type:       SUNW.sctelemetry
RT_description:      sctelemetry service for Oracle Solaris Cluster
RT_version:          1
API_version:         7
RT_basedir:          /usr/cluster/lib/rgm/rt/sctelemetry
Single_instance:     True
Proxy:               False
Init_nodes:          All potential masters
Installed_nodes:     <All>
Failover:            False
Pkglist:              SUNWsc telemetry
RT_system:           False

```

=== Resource Groups and Resources ===

```

Resource Group:      HA_RG
RG_description:      <Null>
RG_mode:              Failover
RG_state:             Managed
Failback:             False

```



```

Nodelist:                                phys-schost-1 phys-schost-2

--- Resources for Group HA_RG ---

Resource:                                HA_R
  Type:                                  SUNW.HAStoragePlus:4
  Type_version:                          4
  Group:                                  HA_RG
  R_description:                          SCSLM_HA_RG
  Resource_project_name:                  True
  Enabled{phys-schost-1}:                  True
  Enabled{phys-schost-2}:                  True
  Monitored{phys-schost-1}:                True
  Monitored{phys-schost-2}:                True

Resource Group:                           cl-db-rg
  RG_description:                          <Null>
  RG_mode:                                 Failover
  RG_state:                                Managed
  Failback:                                False
  Nodelist:                                phys-schost-1 phys-schost-2

--- Resources for Group cl-db-rg ---

Resource:                                cl-db-rs
  Type:                                  SUNW.haderby
  Type_version:                          1
  Group:                                  cl-db-rg
  R_description:                          default
  Resource_project_name:                  True
  Enabled{phys-schost-1}:                  True
  Enabled{phys-schost-2}:                  True
  Monitored{phys-schost-1}:                True
  Monitored{phys-schost-2}:                True

Resource Group:                           cl-tlmtry-rg
  RG_description:                          <Null>
  RG_mode:                                 Scalable
  RG_state:                                Managed
  Failback:                                False
  Nodelist:                                phys-schost-1 phys-schost-2

--- Resources for Group cl-tlmtry-rg ---

Resource:                                cl-tlmtry-rs
  Type:                                  SUNW.sctelemetry
  Type_version:                          1
  Group:                                  cl-tlmtry-rg
  R_description:                          default
  Resource_project_name:                  True
  Enabled{phys-schost-1}:                  True
  Enabled{phys-schost-2}:                  True
  Monitored{phys-schost-1}:                True
  Monitored{phys-schost-2}:                True

=== DID Device Instances ===

DID Device Name:                          /dev/did/rdisk/d1

```

```

Full Device Path:          phys-schost-1:/dev/rdisk/c0t2d0
Replication:              none
default_fencing:         global

DID Device Name:          /dev/did/rdsk/d2
Full Device Path:        phys-schost-1:/dev/rdisk/c1t0d0
Replication:             none
default_fencing:         global

DID Device Name:          /dev/did/rdsk/d3
Full Device Path:        phys-schost-2:/dev/rdisk/c2t1d0
Full Device Path:        phys-schost-1:/dev/rdisk/c2t1d0
Replication:             none
default_fencing:         global

DID Device Name:          /dev/did/rdsk/d4
Full Device Path:        phys-schost-2:/dev/rdisk/c2t2d0
Full Device Path:        phys-schost-1:/dev/rdisk/c2t2d0
Replication:             none
default_fencing:         global

DID Device Name:          /dev/did/rdsk/d5
Full Device Path:        phys-schost-2:/dev/rdisk/c0t2d0
Replication:             none
default_fencing:         global

DID Device Name:          /dev/did/rdsk/d6
Full Device Path:        phys-schost-2:/dev/rdisk/c1t0d0
Replication:             none
default_fencing:         global

=== NAS Devices ===

Nas Device:               nas_filer1
Type:                     sun
User ID:                  root

Nas Device:               nas2
Type:                     sun
User ID:                  llai

```

例1-7 ゾーンクラスタの構成を表示する

次の例は、ゾーンクラスタ構成のプロパティーのリストを示しています。

```

% clzonecluster show
=== Zone Clusters ===

Zone Cluster Name:          sczone
zonename:                  sczone
zonepath:                  /zones/sczone
autoboot:                  TRUE
ip-type:                   shared
enable_priv_net:           TRUE

--- Solaris Resources for sczone ---

```

```

Resource Name:          net
  address:              172.16.0.1
  physical:             auto

Resource Name:          net
  address:              172.16.0.2
  physical:             auto

Resource Name:          fs
  dir:                  /gz/db_qfs/CrsHome
  special:              CrsHome
  raw:
  type:                 samfs
  options:              []

Resource Name:          fs
  dir:                  /gz/db_qfs/CrsData
  special:              CrsData
  raw:
  type:                 samfs
  options:              []

Resource Name:          fs
  dir:                  /gz/db_qfs/OraHome
  special:              OraHome
  raw:
  type:                 samfs
  options:              []

Resource Name:          fs
  dir:                  /gz/db_qfs/OraData
  special:              OraData
  raw:
  type:                 samfs
  options:              []

--- Zone Cluster Nodes for sczone ---

Node Name:              sczone-1
  physical-host:        sczone-1
  hostname:             lzzone-1

Node Name:              sczone-2
  physical-host:        sczone-2
  hostname:             lzzone-2

```

clnasdevice show サブコマンドまたは Oracle Solaris Cluster Manager. を使用して、グローバルまたはゾーンクラスタ用に構成された NAS デバイスを表示することもできます。詳細は、[clnasdevice\(1CL\)](#)のマニュアルページを参照してください。

▼ 基本的なクラスタ構成を検証する方法

cluster(ICL) コマンドで **check** サブコマンドを使用して、グローバルクラスタが正しく機能するために必要な基本構成を検証します。チェックにエラーがない場合、**cluster check** はシェルプロンプトに戻ります。チェックにエラーがある場合、**cluster check** が指定したディレクトリかデフォルトの出力ディレクトリにレポートを生成します。**cluster check** を複数のノードに対して実行すると、**cluster check** は、ノードごとのレポートと複数ノードチェックのレポートを生成します。**cluster list-checks** コマンドを使用して、使用可能なすべてのクラスタチェックの一覧を表示させることもできます。

Oracle Solaris Cluster 3.3 5/11 リリース以降、新しいタイプのチェックを行うよう **cluster check** コマンドが拡張されています。コマンドを使用することにより、ユーザーの対話型操作なしに実行される基本検査だけでなく、対話型検査、および機能検査も実行することができます。基本検査は、**-keyword** オプションが指定されていない場合に実行されます。

- 対話型検査を実行する場合、ユーザーは検査で判定できない情報を入力しなければなりません。検査の実行時には、ファームウェアバージョン番号などの必要な情報を入力するよう促されます。1つ以上の対話型検査を指定するには、**-k interactive** キーワードを使用します。
- 機能検査では、クラスタの特定の機能または動作を検査します。検査の実行時には、フェイルオーバー先となるノードや検査の開始または続行の確認などの情報を入力するよう促されます。機能検査を指定するには、**-k functional** 検査 *id* キーワードを使用します。機能検査は1回につき1つだけ実行できます。

注-一部の機能検査ではクラスタサービスの中断が必要になるので、検査の詳細説明を読み、最初にクラスタの稼働を停止する必要があるかどうか判断したうえで、機能検査を開始してください。この情報を表示するには、次のコマンドを使用します。

```
% cluster list-checks -v -C checkID
```

cluster check コマンドを詳細モードで **-v** フラグを使用して実行して、進捗情報を表示することができます。

注-**cluster check** は、デバイス、ボリューム管理コンポーネント、または Oracle Solaris Cluster 構成を変更するような管理手順を行なったあとに実行してください。

clzonecluster(ICL) コマンドをグローバルクラスタの投票ノードで実行すると、ゾーンクラスタが正しく機能するために必要な構成を検証する一連のチェック

が実行されます。すべてのチェックでエラーがなかった場合、`clzonecluster verify` はシェルプロンプトに戻ります(その場合は、ゾーンクラスタを安全にインストールできます)。エラーがあった場合は、エラーがあったグローバルクラスタノードに関して `clzonecluster verify` から報告があります。`clzonecluster verify` を複数のノードに対して実行すると、ノードごとのレポートと、複数ノードチェックのレポートが生成されます。ゾーンクラスタ内では、`verify` サブコマンドは指定できません。

- 1 グローバルクラスタのアクティブなメンバーノードで、スーパーユーザーになります。グローバルクラスタのノードから、次の手順のステップをすべて実行します。

```
phys-schost# su
```

- 2 最新のチェックがあることを確認します。

[My Oracle Support](#) の「パッチと更新」タブを参照します。詳細検索を使用して、「製品」として「Solaris Cluster」を選択し、「説明」フィールドに「チェック」と指定して、チェックが含まれている Oracle Solaris Cluster パッチを見つけます。まだクラスタにインストールされていないパッチをすべて適用します。

- 3 基本の妥当性検査を実行します。

```
# cluster check -v -o outputdir
```

-v 冗長モード

-o outputdir outputdir サブディレクトリに出力をリダイレクトします。

このコマンドによって、すべての使用可能な基本検査が実行されます。クラスタ機能には影響はありません。

- 4 インタラクティブな妥当性検査を実行します。

```
# cluster check -v -k interactive -o outputdir
```

-k interactive 実行するインタラクティブな妥当性検査を指定します。

このコマンドで、すべての使用可能なインタラクティブ検査が実行され、クラスタについて必要な情報の入力が求められます。クラスタ機能には影響はありません。

- 5 機能の妥当性検査を実行します。

- a. 非冗長モードですべての使用可能な機能検査一覧が表示されます。

```
# cluster list-checks -k functional
```

- b. どの機能検査が、本稼働環境でクラスタの可用性またはサービスを中断する可能性がある処理を実行するかを判断してください。

たとえば、機能検査によって、ノードパニックまたは他のノードへのフェイルオーバーがトリガーされる可能性があります。

```
# cluster list-checks -v -C checkID
```

-C checkID 特定の検査を指定します。

- c. クラスタの機能を中断するような機能検査を実行する場合、クラスタが本稼働状態から除外されるようにします。

- d. 機能検査を開始します。

```
# cluster check -v -k functional -C checkid -o outputdir
```

-k functional 実行する機能の妥当性検査を指定します。

検査の実行に必要な情報を確認し、実行に必要な情報または操作を求めるプロンプトに入力を行います。

- e. 実行する残りの機能検査ごとに、[手順c](#)と[手順d](#)を繰り返します。

注-記録を保存するために、実行する検査ごとに固有の *outputdir* サブディレクトリ名を指定します。*outputdir* 名を再利用する場合、新しい検査の出力によって、再利用した *outputdir* サブディレクトリの既存の内容が上書きされます。

6. ゾーンクラスタの構成を検証して、ゾーンクラスタがインストール可能かどうかを確認します。

```
phys-schost# clzonecluster verify zoneclustername
```

7. 今後の診断に活用できるように、クラスタ構成を記録しておきます。

『[Oracle Solaris Cluster ソフトウェアのインストール](#)』の「[クラスタ構成の診断データを記録する方法](#)」を参照してください。

例 1-8 グローバルクラスタ構成の基本検証 (エラーがない場合)

次の例は、cluster check を詳細モードで phys-schost-1 および phys-schost-2 ノードに対して実行し、エラーが発見されなかった場合を示しています。

```
phys-schost# cluster check -v -h phys-schost-1,
phys-schost-2
```

```
cluster check: Requesting explorer data and node report from phys-schost-1.
cluster check: Requesting explorer data and node report from phys-schost-2.
cluster check: phys-schost-1: Explorer finished.
cluster check: phys-schost-1: Starting single-node checks.
cluster check: phys-schost-1: Single-node checks finished.
```

```
cluster check: phys-schost-2: Explorer finished.
cluster check: phys-schost-2: Starting single-node checks.
cluster check: phys-schost-2: Single-node checks finished.
cluster check: Starting multi-node checks.
cluster check: Multi-node checks finished
#
```

例 1-9 インタラクティブな妥当性検査のリスト

クラスタで実行するために使用できるすべてインタラクティブな妥当性検査の例を以下に示します。出力例に、使用できる検査の例を示します。実際に使用できる検査は、構成によって異なります。

```
# cluster list-checks -k interactive
```

Some checks might take a few moments to run (use -v to see progress)...

```
I6994574 : (Moderate) Fix for GLDv3 interfaces on cluster transport vulnerability applied?
```

例 1-10 機能の妥当性検査の実行

まず、次の例は機能検査の詳細なリストを示します。検査 F6968101 の詳細な説明が表示されます。この説明で、検査によってクラスタサービスが中断されることがわかります。クラスタは稼働状態ではなくなります。機能検査が実行され、`func.test.F6968101.12Jan2011` サブディレクトリに詳細な出力が記録されます。出力例に、使用できる検査の例を示します。実際に使用できる検査は、構成によって異なります。

```
# cluster list-checks -k functional
```

```
F6968101 : (Critical) Perform resource group switchover
F6984120 : (Critical) Induce cluster transport network failure - single adapter.
F6984121 : (Critical) Perform cluster shutdown
F6984140 : (Critical) Induce node panic
...
```

```
# cluster list-checks -v -C F6968101
```

```
F6968101: (Critical) Perform resource group switchover
Keywords: SolarisCluster3.x, functional
Applicability: Applicable if multi-node cluster running live.
Check Logic: Select a resource group and destination node. Perform
'/usr/cluster/bin/clresourcegroup switch' on specified resource group
either to specified node or to all nodes in succession.
Version: 1.2
Revision Date: 12/10/10
```

Take the cluster out of production

```
# cluster check -k functional -C F6968101 -o func.test.F6968101.12Jan2011
```

```
F6968101
```

```
initializing...
initializing xml output...
loading auxiliary data...
```

```

starting check run...
  pschost1, pschost2, pschost3, pschost4:      F6968101.... starting:
Perform resource group switchover

=====

>>> Functional Check <<<

'Functional' checks exercise cluster behavior. It is recommended that you
do not run this check on a cluster in production mode.' It is recommended
that you have access to the system console for each cluster node and
observe any output on the consoles while the check is executed.

If the node running this check is brought down during execution the check
must be rerun from this same node after it is rebooted into the cluster in
order for the check to be completed.

Select 'continue' for more details on this check.

    1) continue
    2) exit

choice: 1

=====

>>> Check Description <<<
...
    Follow onscreen directions

```

例 1-11 グローバルクラスタ構成の検証(エラーがある場合)

次の例は、suncluster という名前のクラスタのノード phys-schost-2 にマウントポイント /global/phys-schost-1 がないことを示しています。レポートは、出力ディレクトリ /var/cluster/logs/cluster_check/<timestamp> に作成されます。

```

phys-schost# cluster check -v -h phys-schost-1,
phys-schost-2 -o
/var/cluster/logs/cluster_check/Dec5/

cluster check: Requesting explorer data and node report from phys-schost-1.
cluster check: Requesting explorer data and node report from phys-schost-2.
cluster check: phys-schost-1: Explorer finished.
cluster check: phys-schost-1: Starting single-node checks.
cluster check: phys-schost-1: Single-node checks finished.
cluster check: phys-schost-2: Explorer finished.

```



```

cluster check: phys-schost-2: Starting single-node checks.
cluster check: phys-schost-2: Single-node checks finished.
cluster check: Starting multi-node checks.
cluster check: Multi-node checks finished.
cluster check: One or more checks failed.
cluster check: The greatest severity of all check failures was 3 (HIGH).
cluster check: Reports are in /var/cluster/logs/cluster_check/<Dec5>.
#
# cat /var/cluster/logs/cluster_check/Dec5/cluster_check-results.suncluster.txt
...
=====
= ANALYSIS DETAILS =
=====
-----
CHECK ID : 3065
SEVERITY : HIGH
FAILURE : Global filesystem /etc/vfstab entries are not consistent across
all Oracle Solaris Cluster 3.x nodes.
ANALYSIS : The global filesystem /etc/vfstab entries are not consistent across
all nodes in this cluster.
Analysis indicates:
FileSystem '/global/phys-schost-1' is on 'phys-schost-1' but missing from 'phys-schost-2'.
RECOMMEND: Ensure each node has the correct /etc/vfstab entry for the
filesystem(s) in question.
...
#

```

▼ グローバルマウントポイントを確認する方法

cluster(1CL) コマンドには、クラスタファイルシステムとそのグローバルマウントポイントに構成エラーがないか、`/etc/vfstab` ファイルを調べるチェックが含まれています。

注 - `cluster check` は、デバイスやボリューム管理コンポーネントに影響を及ぼすような変更をクラスタ構成に加えたあとで実行してください。

- 1 グローバルクラスタのアクティブなメンバーノードで、スーパーユーザーになります。

グローバルクラスタのノードから、次の手順のステップをすべて実行します。

```
% su
```

- 2 グローバルクラスタ構成を検証します。

```
phys-schost# cluster check
```

例 1-12 グローバルマウントポイントの確認

次の例は、suncluster という名前のクラスタのノード phys-schost-2 にマウントポイント /global/schost-1 がないことを示しています。レポートは、出力ディレクトリ、/var/cluster/logs/cluster_check/<timestamp>/ に送信されています。

```
phys-schost# cluster check -v1 -h phys-schost-1,phys-schost-2 -o /var/cluster//logs/cluster_check/Dec5/
```

```
cluster check: Requesting explorer data and node report from phys-schost-1.
cluster check: Requesting explorer data and node report from phys-schost-2.
cluster check: phys-schost-1: Explorer finished.
cluster check: phys-schost-1: Starting single-node checks.
cluster check: phys-schost-1: Single-node checks finished.
cluster check: phys-schost-2: Explorer finished.
cluster check: phys-schost-2: Starting single-node checks.
cluster check: phys-schost-2: Single-node checks finished.
cluster check: Starting multi-node checks.
cluster check: Multi-node checks finished.
cluster check: One or more checks failed.
cluster check: The greatest severity of all check failures was 3 (HIGH).
cluster check: Reports are in /var/cluster/logs/cluster_check/Dec5.
#
# cat /var/cluster/logs/cluster_check/Dec5/cluster_check-results.suncluster.txt

...
=====
= ANALYSIS DETAILS =
=====
-----
CHECK ID : 3065
SEVERITY : HIGH
FAILURE : Global filesystem /etc/vfstab entries are not consistent across
all Oracle Solaris Cluster 3.x nodes.
ANALYSIS : The global filesystem /etc/vfstab entries are not consistent across
all nodes in this cluster.
Analysis indicates:
FileSystem '/global/phys-schost-1' is on 'phys-schost-1' but missing from 'phys-schost-2'.
RECOMMEND: Ensure each node has the correct /etc/vfstab entry for the
filesystem(s) in question.
...
#
# cat /var/cluster/logs/cluster_check/Dec5/cluster_check-results.phys-schost-1.txt

...
=====
= ANALYSIS DETAILS =
=====
-----
CHECK ID : 1398
SEVERITY : HIGH
FAILURE : An unsupported server is being used as an Oracle Solaris Cluster 3.x node.
ANALYSIS : This server may not been qualified to be used as an Oracle Solaris Cluster 3.x node.
Only servers that have been qualified with Oracle Solaris Cluster 3.x are supported as
Oracle Solaris Cluster 3.x nodes.
RECOMMEND: Because the list of supported servers is always being updated, check with
your Oracle representative to get the latest information on what servers
```

are currently supported and only use a server that is supported with Oracle Solaris Cluster 3.x.
 ...
 #

▼ Oracle Solaris Cluster のコマンドログの内容を表示する方法

/var/cluster/logs/commandlog ASCII テキストファイルには、クラスタ内で実行されている選択済みの Oracle Solaris Cluster コマンドのレコードが含まれています。コマンドのロギングは、ユーザーがクラスタを設定したときに自動的に開始され、ユーザーがクラスタをシャットダウンしたときに終了します。コマンドは、実行中およびクラスタモードでブートされたすべてのノード上でロギングされます。

クラスタの構成や現在の状態を表示するようなコマンドは、このファイルに記録されません。

次のような、クラスタの現在の状態の構成や変更を行うコマンドは、このファイルに記録されます。

- claccess
- cldevice
- cldevicegroup
- clinterconnect
- clnasdevice
- clnode
- clquorum
- clreslogicalhostname
- clresource
- clresourcegroup
- clresourcetype
- clressharedaddress
- clsetup
- clsnmp host
- clsnmpmib
- clnsmputer
- cltelemetryattribute
- cluster
- clzonecluster
- scdidadm

commandlog ファイル内のレコードには次の要素を含めることができます。

- 日付とタイムスタンプ
- コマンドの実行元であるホストの名前

- コマンドのプロセス ID
- コマンドを実行したユーザーのログイン名
- ユーザーが実行したコマンド (すべてのオプションとオペランドを含む)

注-すぐに特定し、シェル内でコピー、貼り付け、および実行ができるように、コマンドのオプションは `commandlog` ファイル内では引用符で囲まれています。

- 実行されたコマンドの終了ステータス

注-あるコマンドが未知の結果を伴って異常終了した場合、Oracle Solaris Cluster ソフトウェアは `commandlog` ファイル内には終了ステータスを「表示しません」。

`commandlog` ファイルはデフォルトでは、週に 1 回定期的にアーカイブされます。`commandlog` ファイルのアーカイブポリシーを変更するには、クラスタ内の各ノード上で `crontab` コマンドを使用します。詳細は、[crontab\(1\)](#) のマニュアルページを参照してください。

Oracle Solaris Cluster ソフトウェアは任意の時点で、アーカイブ済みの `commandlog` ファイルを、クラスタノードごとに最大 8 個保持します。現在の週の `commandlog` ファイルの名前は `commandlog` です。最新の完全な週のファイルの名前は `commandlog.0` です。もっとも古い完全な週のファイルの名前は `commandlog.7` です。

- 1 回につき 1 つの画面で、現在の週の `commandlog` ファイルの内容を表示します。

```
phys-schost# more /var/cluster/logs/commandlog
```

例 1-13 Oracle Solaris Cluster のコマンドログの内容の表示

次の例に、`more` コマンドにより表示される `commandlog` ファイルの内容を示します。

```
more -lines10 /var/cluster/logs/commandlog
11/11/2006 09:42:51 phys-schost-1 5222 root START - clsetup
11/11/2006 09:43:36 phys-schost-1 5758 root START - clrg add "app-sa-1"
11/11/2006 09:43:36 phys-schost-1 5758 root END 0
11/11/2006 09:43:36 phys-schost-1 5760 root START - clrg set -y
"RG_description=Department Shared Address RG" "app-sa-1"
11/11/2006 09:43:37 phys-schost-1 5760 root END 0
11/11/2006 09:44:15 phys-schost-1 5810 root START - clrg online "app-sa-1"
11/11/2006 09:44:15 phys-schost-1 5810 root END 0
11/11/2006 09:44:19 phys-schost-1 5222 root END -20988320
12/02/2006 14:37:21 phys-schost-1 5542 jbloggs START - clrg -c -g "app-sa-1"
-y "RG_description=Joe Bloggs Shared Address RG"
12/02/2006 14:37:22 phys-schost-1 5542 jbloggs END 0
```

Oracle Solaris Cluster と RBAC

この章では、役割に基づくアクセス制御 (RBAC) について Oracle Solaris Cluster に関連する範囲で説明します。次のトピックについて述べます。

- 53 ページの「RBAC の設定と Oracle Solaris Cluster での使用」
- 54 ページの「Oracle Solaris Cluster RBAC の権利プロファイル」
- 55 ページの「Oracle Solaris Cluster 管理権利プロファイルによる RBAC 役割の作成と割り当て」
- 59 ページの「ユーザーの RBAC プロパティーの変更」

RBAC の設定と Oracle Solaris Cluster での使用

次の表を参考に、RBAC の設定と使用について確認するドキュメントを選んでください。RBAC を作成して、Oracle Solaris Cluster ソフトウェアで使用するための手順については、この章で後述します。

タスク	手順
RBAC の詳細を調べる	『Solaris のシステム管理: セキュリティーサービス』の第 8 章「役割と特権の使用 (概要)」
RBAC の設定、要素の管理、RBAC の使用など	『Solaris のシステム管理: セキュリティーサービス』の第 9 章「役割に基づくアクセス制御の使用 (タスク)」
RBAC の要素とツールの詳細を調べる	『Solaris のシステム管理: セキュリティーサービス』の第 10 章「役割によるアクセス制御 (参照)」

Oracle Solaris Cluster RBAC の権利プロファイル

コマンド行で実行する Oracle Solaris Cluster Manager および選択した Oracle Solaris Cluster のコマンドとオプションは、承認のために RBAC を使用します。RBAC の承認を必要とする Oracle Solaris Cluster のコマンドとオプションは、次の承認レベルを 1 つ以上必要とします。Oracle Solaris Cluster RBAC の権利プロファイルは、グローバルクラスタ内の投票ノードと非投票ノードの両方に適用されます。

- solaris.cluster.read

一覧表示、表示、およびその他の読み取り操作の承認。
- solaris.cluster.admin

クラスタオブジェクトの状態を変更する承認。
- solaris.cluster.modify

クラスタオブジェクトのプロパティーを変更する承認。

Oracle Solaris Cluster コマンドにより必要とされる RBAC の承認については、コマンドのマニュアルページを参照してください。

RBAC の権利プロファイルには 1 つ以上の RBAC の承認が含まれます。これらの権利プロファイルをユーザーまたは役割に割り当てることで、Oracle Solaris Cluster に対するさまざまなレベルのアクセス権をユーザーや役割に与えることができます。次に、Oracle Solaris Cluster ソフトウェアに含まれる権利プロファイルを示します。

注 - 次の表に示す RBAC の権利プロファイルは、以前の Oracle Solaris Cluster リリースで定義された古い RBAC の承認を引き続きサポートします。

権利プロファイル	含まれる承認	役割に許可されたアクセス権
Oracle Solaris Cluster コマンド	なし。ただし、 <code>euclid=0</code> を指定して実行される Oracle Solaris Cluster コマンドのリストが含まれます。	<div>すべての Oracle Solaris Cluster コマンドの次のサブコマンドを含めて、クラスタを構成および管理するために使用する一部の Oracle Solaris Cluster コマンドの実行。</div> <div><div>■ list</div><div>■ show</div><div>■ status</div></div> <div>scha_control(1HA)</div> <div>scha_resource_get(1HA)</div> <div>scha_resource_setstatus(1HA)</div> <div>scha_resourcegroup_get(1HA)</div> <div>scha_resourcetype_get(1HA)</div>

権利プロファイル	含まれる承認	役割に許可されたアクセス権
基本 Oracle Solaris ユーザー	この既存の Oracle Solaris 権利プロファイルには、Oracle Solaris の承認のほか、次のものが含まれます。 <code>solaris.cluster.read</code>	Oracle Solaris Cluster コマンドの一覧表示、表示、およびその他の読み取り操作の実行と、Oracle Solaris Cluster Manager GUI へのアクセス。
Cluster Operation	この権利プロファイルは Oracle Solaris Cluster に固有で、次の承認が含まれています。 <code>solaris.cluster.read</code> <code>solaris.cluster.admin</code>	一覧表示、表示、エクスポート、ステータス、およびその他の読み取り操作の実行と、Oracle Solaris Cluster Manager GUI へのアクセス。 クラスタオブジェクトの状態の変更。
システム管理者	この既存の Oracle Solaris 権利プロファイルには、Cluster 管理プロファイルに含まれるものと同じ承認が入っています。	Cluster Management 役割 ID に許可された作業と、その他のシステム管理作業を行えます。
Cluster Management	この権利プロファイルには、Cluster Operation プロファイルに含まれるものと同じ承認のほか、以下の承認が含まれます。 <code>solaris.cluster.modify</code>	Cluster Operation 役割 ID が実行できるのと同じオペレーションおよびクラスタオブジェクトのプロパティの変更を実行します。

Oracle Solaris Cluster 管理権利プロファイルによる RBAC 役割の作成と割り当て

このタスクを使用して、Oracle Solaris Cluster の管理権利プロファイルを持つ新しい RBAC の役割を作成し、この新しい役割にユーザーを割り当てます。

▼ 管理役割ツールを使用して役割を作成する方法

始める前に 役割を作成するには、プライマリ管理者の権利プロファイルが割り当てられている役割になるか、`root`ユーザーとして実行する必要があります。

1 管理役割ツールを起動します。

管理役割ツールを実行するには、『[Solaris のシステム管理: セキュリティサービス](#)』の「[Solaris 管理コンソールで役割を引き受ける方法](#)」で説明されているように、Solaris Management Console を起動します。ユーザーツールコレクションを開き、「管理役割」アイコンをクリックします。

2 「管理役割を追加」ウィザードを起動します。

「アクション」メニューから「管理役割の追加」を選択し、役割を構成するための「管理役割を追加」ウィザードを起動します。

3 クラスタ管理の権利プロファイルを割り当てる役割を設定します。

「次へ」ボタンと「戻る」ボタンを使用して、ダイアログボックス間を移動します。「次へ」ボタンは、すべての必須フィールドに記入しないとアクティブにならないことに注意してください。最後のダイアログボックスでは、入力したデータの確認が可能で、「戻る」ボタンを使用してエントリを変更するか、「完了」をクリックして新しい役割を保存できます。次のリストで、ダイアログボックスのフィールドとボタンについてまとめます。

役割名	役割の短い名前。
完全な名前	長いバージョンの名前。
説明	役割の説明。
役割 ID 番号	役割の UID。自動的に増えます。
役割シェル	役割が使用できるプロファイルシェル。管理者の C シェル、管理者の Bourne シェル、または管理者の Korn シェル。
役割メーリングリストを作成	この役割に割り当てられているユーザー用のメーリングリストを作成します。
使用可能な権利/付与された権利	<p>役割の権利プロファイルの割り当てまたは削除を行います。</p> <p>同じコマンドの複数入力、システムによって防止されないことに注意してください。権利プロファイルで最初に出現するコマンドに割り当てられている属性が優先され、それ以後に出現するものはすべて無視されます。上下の矢印を使用して順序を変更します。</p>
サーバー	ホームディレクトリのサーバー。
パス	ホームディレクトリのパス。
追加	この役割になることができるユーザーを追加します。同じスコープに属している必要があります。
削除	この役割に割り当てられているユーザーを削除します。

注- このプロファイルは、役割に割り当てられているプロファイルのリストの最初に配置する必要があります。

- 4 新しく作成された役割に、**Fmv426;**の機能や **Oracle Solaris Cluster** のコマンドを使用する必要があるユーザーを追加します。
useradd(1M) コマンドを使用して、ユーザーアカウントをシステムに追加します。-p オプションは、ユーザーのアカウントに役割を割り当てます。
- 5 「完了」をクリックします。
- 6 端末ウィンドウを開き、**root** になります。
- 7 ネームサービスキャッシュデーモンを起動して停止します。
 新しい役割は、ネームサービスキャッシュデーモンを再起動するまで有効になりません。root になったあと、次のテキストを入力します。

```
# /etc/init.d/nscd stop
# /etc/init.d/nscd start
```

▼ コマンド行から役割を作成する方法

- 1 スーパーユーザーになるか、RBAC の承認 **solaris.cluster.admin** を提供する役割になります。
- 2 次のいずれかの役割の作成方法を選択します。
 - ローカルスコープの役割の場合は、**roleadd(1M)** コマンドを使用して、新しいローカルの役割およびその属性を指定します。
 - または、ローカルスコープの役割について **user_attr(4)** ファイルを編集し、**type=role** でユーザーを追加します。
 この方法は緊急時にのみ使用します。
 - ネームサービスの役割の場合は、**smrole(1M)** コマンドを使用して新しい役割およびその属性を指定します。
 このコマンドは、スーパーユーザー、またはその他の役割を作成できる役割による認証を必要とします。**smrole** は、すべてのネームサービスに適用できます。このコマンドは、Solaris Management Console サーバーのクライアントとして動作します。

3 ネームサービスキャッシュデーモンを起動して停止します。

新しい役割は、ネームサービスキャッシュデーモンを再起動するまで有効になりません。root として、次のテキストを入力します。

```
# /etc/init.d/nscd stop
# /etc/init.d/nscd start
```

例 2-1 smrole コマンドを使用してカスタムの Operator 役割を作成する

次のコマンドシーケンスは、smrole コマンドを使用して役割を作成します。この例では、新しい Operator 役割が作成され、標準の Operator 権利プロファイルと Media Restore 権利プロファイルが割り当てられます。

```
% su primaryadmin
# /usr/sadm/bin/smrole add -H myHost -- -c "Custom Operator" -n oper2 -a johnDoe \
-d /export/home/oper2 -F "Backup/Restore Operator" -p "Operator" -p "Media Restore"
```

Authenticating as user: primaryadmin

Type /? for help, pressing <enter> accepts the default denoted by []
Please enter a string value for: password :: <type primaryadmin password>

Loading Tool: com.sun.admin.usermgr.cli.role.UserMgrRoleCli from myHost
Login to myHost as user primaryadmin was successful.
Download of com.sun.admin.usermgr.cli.role.UserMgrRoleCli from myHost was successful.

Type /? for help, pressing <enter> accepts the default denoted by []
Please enter a string value for: password :: <type oper2 password>

```
# /etc/init.d/nscd stop
# /etc/init.d/nscd start
```

新しく作成した役割およびその他の役割を表示するには、次のように smrole コマンドに list オプションを指定します。

```
# /usr/sadm/bin/smrole list --
```

Authenticating as user: primaryadmin

Type /? for help, pressing <enter> accepts the default denoted by []
Please enter a string value for: password :: <type primaryadmin password>

Loading Tool: com.sun.admin.usermgr.cli.role.UserMgrRoleCli from myHost
Login to myHost as user primaryadmin was successful.
Download of com.sun.admin.usermgr.cli.role.UserMgrRoleCli from myHost was successful.

root	0	Super-User
primaryadmin	100	Most powerful role
sysadmin	101	Performs non-security admin tasks
oper2	102	Custom Operator

ユーザーの RBAC プロパティの変更

ユーザーアカウントツールかコマンド行のいずれかを使用すると、ユーザーの RBAC プロパティを変更できます。ユーザーの RBAC プロパティを変更するには、次のいずれかの手順を選択します。

- [59 ページの「ユーザーアカウントツールを使用してユーザーの RBAC プロパティを変更する方法」](#)
- [60 ページの「コマンド行からユーザーの RBAC プロパティを変更する方法」](#)

▼ ユーザーアカウントツールを使用してユーザーの RBAC プロパティを変更する方法

始める前に ユーザーのプロパティを変更するには、root ユーザーとしてユーザーツールコレクションを実行するか、プライマリ管理者の権利プロファイルが割り当てられている役割になる必要があります。

- 1 ユーザーアカウントツールを起動します。
ユーザーアカウントツールを実行するには、[『Solaris のシステム管理: セキュリティサービス』の「Solaris 管理コンソールで役割を引き受ける方法」](#)で説明されているように、Solaris Management Console を起動します。ユーザーツールコレクションを開き、「ユーザーアカウント」アイコンをクリックします。
ユーザーアカウントツールが起動すると、既存のユーザーアカウントのアイコンが表示ペインに表示されます。
- 2 変更する「ユーザーアカウント」アイコンをクリックし、「アクション」メニューから「プロパティ」を選択するか、ユーザーアカウントのアイコンをダブルクリックします。
- 3 次のように、変更するプロパティのダイアログボックスで適切なタブをクリックします。
 - ユーザーに割り当てられている役割を変更するには、「役割」タブをクリックして、変更する役割の割り当てを「使用可能な役割」または「割り当てられた役割」のうち適切な列に移動します。
 - ユーザーに割り当てられている権利プロファイルを変更するには、「権利」タブをクリックして、それを有効な権利または割り当てられた権利のどちらか、適切な列に移動します。

注-権利プロファイルは、ユーザーに直接割り当てないようにします。特権のあるアプリケーションを実行する場合はユーザーがその役割になることが必要になる方法をお勧めします。この方法では、ユーザーによる特権の濫用が防止されます。

▼ コマンド行からユーザーの **RBAC** プロパティを変更する方法

- 1 スーパーユーザーになるか、RBAC 承認 **solaris.cluster.modify** を提供する役割になります。
- 2 次のように適切なコマンドを選択します。
 - ローカルスコープで定義されているユーザーに割り当てられた承認、役割、または権利プロファイルを変更するには、**usermod(1M)** コマンドを使用します。
 - また同じくローカルスコープに定義されたユーザーに割り当てられている承認、役割、または権利プロファイルを変更する場合は、**user_attr** ファイルを編集することもできます。
この方法は緊急時にのみ使用します。
 - ネームサービスで定義されているユーザーに割り当てられた承認、役割、または権利プロファイルを変更するには、**smuser(1M)** コマンドを使用します。
このコマンドは、スーパーユーザー、またはユーザーファイルを変更できる役割による認証を必要とします。**smuser** は、すべてのネームサービスに適用できます。**smuser** は、Solaris Management Console サーバーのクライアントとして動作します。

クラスタの停止とブート

この章では、グローバルクラスタ、ゾーンクラスタ、および個々のノードの停止方法とブート方法について説明します。非大域ゾーンのブートについては、『[Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン](#)』の第 18 章「非大域ゾーンの計画と構成(タスク)」を参照してください。

- 61 ページの「クラスタの停止とブートの概要」
- 71 ページの「クラスタ内の 1 つのノードの停止とブート」
- 85 ページの「満杯の `/var` ファイルシステムを修復する」

この章の関連手順の詳細な説明については、82 ページの「非クラスタモードでノードをブートする方法」と表 3-2 を参照してください。

クラスタの停止とブートの概要

Oracle Solaris Cluster の `cluster shutdown` コマンドは、グローバルクラスタサービスを正しい順序で停止し、グローバルクラスタ全体をクリーンに停止します。`cluster shutdown` コマンドは、グローバルクラスタの場所を移動するときに使用できます。また、アプリケーションエラーによってデータが破損した場合に、グローバルクラスタを停止するときにも使用できます。`clzonecluster halt` コマンドは、特定のノード上のゾーンクラスタ、または構成済みのすべてのノード上のゾーンクラスタ全体を停止します(ゾーンクラスタ内で `cluster shutdown` コマンドを使用することもできます)。詳細は、[cluster\(1CL\)](#) のマニュアルページを参照してください。

この章の手順の `phys-schost#` は、グローバルクラスタプロンプトを表します。`clzonecluster` の対話型シェルスプロンプトは `clzc:schost>` です。

注- グローバルクラスタ全体を正しく停止するには、`cluster shutdown` コマンドを使用します。Oracle Solaris の `shutdown` コマンドは `clnode(1CL)` `evacuate` コマンドとともに使用して、個々のノードをシャットダウンします。詳細については、[63 ページの「クラスタを停止する方法」](#) または [71 ページの「クラスタ内の 1 つのノードの停止とブート」](#) を参照してください。

`cluster shutdown` と `clzonecluster halt` コマンドは、それぞれグローバルクラスタまたはゾーンクラスタ内のすべてのノードを停止します。その処理は次のように行われます。

1. 実行中のすべてのリソースグループをオフラインにする。
2. グローバルクラスタまたはゾーンクラスタのすべてのクラスタファイルシステムをマウント解除する。
3. `cluster shutdown` コマンドが、グローバルクラスタまたはゾーンクラスタ上のアクティブなデバイスサービスを停止する。
4. `cluster shutdown` コマンドが `init 0` を実行して、クラスタ上のすべてのノードを OpenBoot PROM `ok` プロンプトの状態にする (SPARC ベースのシステムの場合) か、または GRUB メニューの「Press any key to continue」メッセージの状態にする (x86 ベースのシステムの場合)。GRUB メニューについては、[『Oracle Solaris の管理: 基本管理』の「GRUB を使用して x86 システムをブートする \(タスクマップ\)」](#) で詳しく説明しています。`clzonecluster halt` コマンドが `zoneadm -z zoneclustername halt` コマンドを実行して、ゾーンクラスタのゾーンを停止します (ただし、シャットダウンは行いません)。

注- 必要であれば、ノードを非クラスタモードで (つまり、ノードがクラスタメンバーシップを取得しないように) ブートできます。非クラスタモードは、クラスタソフトウェアをインストールしたり、特定の管理手順を実行する際に役立ちます。詳細は、[82 ページの「非クラスタモードでノードをブートする方法」](#) を参照してください。

表 3-1 タスクリスト: クラスタの停止とブート

タスク	手順
クラスタの停止。	63 ページの「クラスタを停止する方法」
すべてのノードを起動してクラスタを起動 クラスタメンバーシップを取得できるように、ノードにはクラスタインターコネクトとの動作中の接続が必要です。	65 ページの「クラスタをブートする方法」
クラスタのリブート	67 ページの「クラスタをリブートする方法」

▼ クラスタを停止する方法

グローバルクラスタ、1つのゾーンクラスタ、またはすべてのゾーンクラスタを停止できます。



注意-グローバルクラスタノードやゾーンクラスタノードを停止する場合に、`send brk` をクラスタコンソール上で使用しないでください。この機能はクラスタ内ではサポートされません。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 グローバルクラスタまたはゾーンクラスタで **Oracle Real Application Clusters (RAC)** が実行されている場合は、停止するクラスタ上のデータベースのインスタンスをすべて停止します。
停止の手順については、Oracle RAC 製品のドキュメントを参照してください。
- 2 クラスタ内の任意のノードで、スーパーユーザーになるか、**RBAC** の承認 `solaris.cluster.admin` を提供する役割になります。グローバルクラスタのノードから、次の手順のステップをすべて実行します。
- 3 グローバルクラスタ、1つのゾーンクラスタ、またはすべてのゾーンクラスタを停止します。

- グローバルクラスタを停止します。この操作を行うと、すべてのゾーンクラスタも停止します。

```
phys-schost# cluster shutdown -g0 -y
```

- 特定のゾーンクラスタを停止します。

```
phys-schost# clzonecluster halt zoneclustername
```

- すべてのゾーンクラスタを停止します。

```
phys-schost# clzonecluster halt +
```

ゾーンクラスタ内で `cluster shutdown` コマンドを使用して、特定のゾーンクラスタを停止することもできます。

- 4 **SPARC** ベースのシステムの場合は、グローバルクラスタまたはゾーンクラスタ上のすべてのノードが **ok** プロンプトの状態になったことを確認します。**x86** ベースのシステムの場合は、すべてのノードが **GRUB** メニューの状態になったことを確認します。
SPARC ベースのシステムの場合はすべてのノードが **ok** プロンプトになるまで、**x86** ベースのシステムの場合はすべてのノードが **Boot Subsystem** の状態になるまで、どのノードの電源も切らないでください。
 - クラスタ内でまだ稼働および実行中の別のグローバルクラスタノードから、1つ以上のグローバルクラスノードのステータスを確認します。
 phys-schost# **cluster status -t node**
 - **status** サブコマンドを使用して、ゾーンクラスタが停止したことを確認します。
 phys-schost# **clzonecluster status**
- 5 必要であれば、グローバルクラスタのノードの電源を切ります。

例 3-1 ゾーンクラスタの停止

次の例では、*sparse-sczone* というゾーンクラスタをシャットダウンしています。

```
phys-schost# clzonecluster halt sparse-sczone
Waiting for zone halt commands to complete on all the nodes of the zone cluster "sparse-sczone"...
Sep  5 19:06:01 schost-4 cl_runtime: NOTICE: Membership : Node 2 of cluster 'sparse-sczone' died.
Sep  5 19:06:01 schost-4 cl_runtime: NOTICE: Membership : Node 4 of cluster 'sparse-sczone' died.
Sep  5 19:06:01 schost-4 cl_runtime: NOTICE: Membership : Node 3 of cluster 'sparse-sczone' died.
Sep  5 19:06:01 schost-4 cl_runtime: NOTICE: Membership : Node 1 of cluster 'sparse-sczone' died.
phys-schost#
```

例 3-2 SPARC: グローバルクラスタの停止

次に、正常なグローバルクラスタの動作を停止して、すべてのノードを停止し、**ok** プロンプトが表示されたときのコンソールの出力例を示します。ここでは、**-g 0** オプションで停止の猶予期間をゼロに設定し、**-y** オプションで、確認プロンプトに対して自動的に **yes** と応答するよう指定しています。停止メッセージは、グローバルクラスタ内のほかのノードのコンソールにも表示されます。

```
phys-schost# cluster shutdown -g0 -y
Wed Mar 10 13:47:32 phys-schost-1 cl_runtime:
WARNING: CMM monitoring disabled.
phys-schost-1#
INIT: New run level: 0
The system is coming down. Please wait.
System services are now being stopped.
/etc/rc0.d/K05initrgm: Calling clnode evacuate
The system is down.
syncing file systems... done
Program terminated
ok
```


例 3-3 x86: グローバルクラスタの停止

次に、正常なグローバルクラスタの動作を停止して、すべてのノードを停止したときのコンソールの出力例を示します。この例では、すべてのノードで `ok` プロンプトが表示されません。ここでは、`-g 0` オプションで停止の猶予期間をゼロに設定し、`-y` オプションで、確認プロンプトに対して自動的に `yes` と応答するよう指定しています。停止メッセージは、グローバルクラスタ内のほかのノードのコンソールにも表示されます。

```
phys-schost# cluster shutdown -g0 -y
May  2 10:32:57 phys-schost-1 cl_runtime:
WARNING: CMM: Monitoring disabled.
root@phys-schost-1#
INIT: New run level: 0
The system is coming down. Please wait.
System services are now being stopped.
/etc/rc0.d/K05initrgm: Calling clnode evacuate
failfasts already disabled on node 1
Print services already stopped.
May  2 10:33:13 phys-schost-1 syslogd: going down on signal 15
The system is down.
syncing file systems... done
Type any key to continue
```

参照 停止したグローバルクラスタまたはゾーンクラスタを再起動するには、[65 ページ](#)の「[クラスタをブートする方法](#)」を参照してください。

▼ クラスタをブートする方法

この手順では、ノードが停止されているグローバルクラスタまたはゾーンクラスタを起動する方法について説明します。グローバルクラスタノードに対して、`ok` プロンプト (SPARC システムの場合) または「Press any key to continue」メッセージ (GRUB ベースの x86 システムの場合) が表示されています。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

注- ゾーンクラスタを作成するために、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』の「[ゾーンクラスタの構成](#)」の手順を実行します。

- 1 各ノードをクラスタモードでブートします。グローバルクラスタのノードから、次の手順のステップをすべて実行します。

- SPARC ベースのシステム上で、次のコマンドを実行します。

```
ok boot
```

- x86 ベースのシステム上で、次のコマンドを実行します。

GRUB メニューが表示された時点で、適切な Oracle Solaris エントリを選択し、Enter キーを押します。次のような GRUB メニューが表示されます。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-----+
| Solaris 10 /sol_10_x86                               |
| Solaris failsafe                                     |
|                                                       |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

GRUB ベースのブートの詳細については、『[Oracle Solaris の管理: 基本管理](#)』の「[GRUB を使用して x86 システムをブートする \(タスクマップ\)](#)」を参照してください。

注-クラスタメンバーシップを取得できるように、ノードにはクラスタインターコネクトとの動作中の接続が必要です。

- ゾーンクラスタが1つの場合は、ゾーンクラスタ全体をブートできます。

```
phys-schost# clzonecluster boot zoneclustername
```

- ゾーンクラスタが複数ある場合は、すべてのゾーンクラスタをブートできます。*zoneclustername* の代わりに + を使用してください。

- 2 ノードが問題なくブートし、オンラインであることを確認します。

cluster(1CL) ステータスコマンドは、グローバルクラスタノードのステータスを報告します。

```
phys-schost# cluster status -t node
```

clzonecluster(1CL) ステータスコマンドをグローバルクラスタノードから実行すると、ゾーンクラスタノードの状態が報告されます。

```
phys-schost# clzonecluster status
```

注- ノードの `/var` ファイルシステムが満杯になると、そのノード上では Oracle Solaris Cluster が再起動できなくなる可能性があります。この問題が発生した場合は、[86 ページの「満杯の `/var` ファイルシステムを修復する方法」](#)を参照してください。

例 3-4 SPARC: グローバルクラスタのブート

次に、ノード `phys-schost-1` をブートしてグローバルクラスタに結合させたときのコンソールの出力例を示します。グローバルクラスタ内のほかのノードのコンソールにも同様のメッセージが表示されます。ゾーンクラスタの自動ブートプロパティが `true` に設定されている場合は、そのマシン上のグローバルクラスタノードがブートすると、ゾーンクラスタノードも自動的にブートされます。

グローバルクラスタノードがリブートすると、そのマシン上のゾーンクラスタノードがすべて停止します。同じマシン上に、自動起動プロパティが `true` に設定されたゾーンクラスタノードがある場合は、グローバルクラスタノードが再起動するとゾーンクラスタノードも再起動されます。

```
ok boot
Rebooting with command: boot
...
Hostname: phys-schost-1
Booting as part of a cluster
NOTICE: Node phys-schost-1 with votecount = 1 added.
NOTICE: Node phys-schost-2 with votecount = 1 added.
NOTICE: Node phys-schost-3 with votecount = 1 added.
...
NOTICE: Node phys-schost-1: attempting to join cluster
...
NOTICE: Node phys-schost-2 (incarnation # 937690106) has become reachable.
NOTICE: Node phys-schost-3 (incarnation # 937690290) has become reachable.
NOTICE: cluster has reached quorum.
NOTICE: node phys-schost-1 is up; new incarnation number = 937846227.
NOTICE: node phys-schost-2 is up; new incarnation number = 937690106.
NOTICE: node phys-schost-3 is up; new incarnation number = 937690290.
NOTICE: Cluster members: phys-schost-1 phys-schost-2 phys-schost-3.
...
```

▼ クラスタをリブートする方法

グローバルクラスタを停止するために `cluster shutdown` コマンドを実行してから、各ノード上で `boot` コマンドを使用してグローバルクラスタをブートします。ゾーンクラスタを停止するために `clzonecluster halt` コマンドを使用してから、`clzonecluster boot` コマンドを使用してゾーンクラスタをブートします。`clzonecluster reboot` コマンドを使用することもできます。詳細は、[cluster\(1CL\)](#)、[boot\(1M\)](#)、および [clzonecluster\(1CL\)](#) のマニュアルページを参照してください。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 **Oracle RAC** が動作しているクラスタの場合は、停止するクラスタ上のデータベースのすべてのインスタンスを停止します。

停止の手順については、Oracle RAC 製品のドキュメントを参照してください。

- 2 クラスタ内の任意のノードで、スーパーユーザーになるか、**RBAC** の承認 `solaris.cluster.admin` を提供する役割になります。グローバルクラスタのノードから、次の手順のステップをすべて実行します。

- 3 クラスタを停止します。

- グローバルクラスタを停止します。

```
phys-schost# cluster shutdown -g0 -y
```

- ゾーンクラスタがある場合は、グローバルクラスタノードからゾーンクラスタを停止します。

```
phys-schost# clzonecluster halt zoneclustername
```

各ノードが停止します。ゾーンクラスタ内で `cluster shutdown` コマンドを使用して、ゾーンクラスタを停止することもできます。

注-クラスタメンバーシップを取得できるように、ノードにはクラスタインターコネクタとの動作中の接続が必要です。

- 4 各ノードをブートします。

停止中に構成を変更した場合以外は、どのような順序でノードをブートしてもかまいません。停止中に構成を変更した場合は、最新の構成情報を持つノードを最初に起動する必要があります。

- SPARC ベースのシステムのグローバルクラスタノードの場合は、次のコマンドを実行します。

```
ok boot
```

- x86 ベースのシステムのグローバルクラスタノードの場合は、次のコマンドを実行します。

GRUB メニューが表示された時点で、適切な Oracle Solaris OS エントリを選択し、Enter キーを押します。次のような GRUB メニューが表示されます。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-----+
| Solaris 10 /sol_10_x86          |
| Solaris failsafe                |
|                                 |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

注-クラスタメンバーシップを取得できるように、ノードにはクラスタインターコネクトとの動作中の接続が必要です。

GRUB ベースのブートの詳細については、[『Oracle Solaris の管理: 基本管理』の「GRUBを使用してx86システムをブートする\(タスクマップ\)」](#)を参照してください。

- ゾーンクラスタの場合は、グローバルクラスタの1つのノードで次のコマンドを入力して、ゾーンクラスタをブートします。

```
phys-schost# clzonecluster boot zoneclustername
```

クラスタコンポーネントがブートすると、ブートされたノードのコンソールにメッセージが表示されます。

5 ノードが問題なくブートし、オンラインであることを確認します。

- **clnode status** コマンドを実行すると、グローバルクラスタ上のノードのステータスが報告されます。

```
phys-schost# clnode status
```

- **clzonecluster status** コマンドをグローバルクラスタノード上で実行すると、ゾーンクラスタノードのステータスが報告されます。

```
phys-schost# clzonecluster status
```

ゾーンクラスタ内で **cluster status** コマンドを実行して、ノードのステータスを確認することもできます。

注-ノードの /var ファイルシステムが満杯になると、そのノード上では Oracle Solaris Cluster が再起動できなくなる可能性があります。この問題が発生した場合は、[86 ページの「満杯の /var ファイルシステムを修復する方法」](#)を参照してください。

例 3-5 ゾーンクラスタのリブート

次の例は、*sparse-sczone* というゾーンクラスタを停止してブートする方法を示しています。clzonecluster reboot コマンドを使用することもできます。

```

phys-schost# clzonecluster halt sparse-sczone
Waiting for zone halt commands to complete on all the nodes of the zone cluster "sparse-sczone"...
Sep  5 19:17:46 schost-4 cl_runtime: NOTICE: Membership : Node 4 of cluster 'sparse-sczone' died.
Sep  5 19:17:46 schost-4 cl_runtime: NOTICE: Membership : Node 2 of cluster 'sparse-sczone' died.
Sep  5 19:17:46 schost-4 cl_runtime: NOTICE: Membership : Node 1 of cluster 'sparse-sczone' died.
Sep  5 19:17:46 schost-4 cl_runtime: NOTICE: Membership : Node 3 of cluster 'sparse-sczone' died.
phys-schost#
phys-schost# clzonecluster boot sparse-sczone
Waiting for zone boot commands to complete on all the nodes of the zone cluster "sparse-sczone"...
phys-schost# Sep  5 19:18:23 schost-4 cl_runtime: NOTICE: Membership : Node 1 of cluster
'sparse-sczone' joined.
Sep  5 19:18:23 schost-4 cl_runtime: NOTICE: Membership : Node 2 of cluster 'sparse-sczone' joined.
Sep  5 19:18:23 schost-4 cl_runtime: NOTICE: Membership : Node 3 of cluster 'sparse-sczone' joined.
Sep  5 19:18:23 schost-4 cl_runtime: NOTICE: Membership : Node 4 of cluster 'sparse-sczone' joined.

phys-schost#
phys-schost# clzonecluster status

=== Zone Clusters ===

--- Zone Cluster Status ---

Name           Node Name   Zone HostName   Status   Zone Status
-----
sparse-sczone  schost-1    sczone-1        Online   Running
                schost-2    sczone-2        Online   Running
                schost-3    sczone-3        Online   Running
                schost-4    sczone-4        Online   Running
phys-schost#

```

例3-6 SPARC: グローバルクラスタのリブート

次に、正常なグローバルクラスタの動作を停止して、すべてのノードを停止し、ok プロンプトが表示され、グローバルクラスタが再起動したときのコンソールの出力例を示します。ここでは、**-g 0** オプションで停止の猶予期間をゼロに設定し、**-y** オプションで、確認プロンプトに対して自動的に **yes** と応答するよう指定しています。停止メッセージは、グローバルクラスタ内のほかのノードのコンソールにも表示されます。

```

phys-schost# cluster shutdown -g0 -y
Wed Mar 10 13:47:32 phys-schost-1 cl_runtime:
WARNING: CMM monitoring disabled.
phys-schost-1#
INIT: New run level: 0
The system is coming down. Please wait.
...
The system is down.
syncing file systems... done
Program terminated
ok boot
Rebooting with command: boot
...
Hostname: phys-schost-1
Booting as part of a cluster
...

```

```

NOTICE: Node phys-schost-1: attempting to join cluster
...
NOTICE: Node phys-schost-2 (incarnation # 937690106) has become reachable.
NOTICE: Node phys-schost-3 (incarnation # 937690290) has become reachable.
NOTICE: cluster has reached quorum.
...
NOTICE: Cluster members: phys-schost-1 phys-schost-2 phys-schost-3.
...
NOTICE: Node phys-schost-1: joined cluster
...
The system is coming up. Please wait.
checking ufs filesystems
...
reservation program successfully exiting
Print services started.
volume management starting.
The system is ready.
phys-schost-1 console login:
NOTICE: Node phys-schost-1: joined cluster
...
The system is coming up. Please wait.
checking ufs filesystems
...
reservation program successfully exiting
Print services started.
volume management starting.
The system is ready.
phys-schost-1 console login:

```

クラスタ内の1つのノードの停止とブート

グローバルクラスタノード、ゾーンクラスタノード、または非大域ゾーンをシャットダウンできます。ここでは、グローバルクラスタノードとゾーンクラスタノードを停止する手順を説明します。

グローバルクラスタノードを停止するには、`clnode evacuate` コマンドを Oracle Solaris の `shutdown` コマンドとともに使用します。`cluster shutdown` コマンドは、グローバルクラスタ全体を停止する場合にのみ使用します。

ゾーンクラスタノードでは、`clzonecluster halt` コマンドをグローバルクラスタで使用して、1つのゾーンクラスタノードまたはゾーンクラスタ全体を停止します。`clnode evacuate` コマンドと `shutdown` コマンドを使用してゾーンクラスタノードを停止することもできます。

非大域ゾーンのシャットダウンとブートについては、『Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン』の第20章「非大域ゾーンのインストール、ブート、停止、アンインストール、および複製(タスク)」を参照してください。`clnode(1CL)`、`shutdown(1M)`、および `clzonecluster(1CL)` のマニュアルページも参照してください。

この章の手順の `phys-schost#` は、グローバルクラスタプロンプトを表します。`clzonecluster` の対話型シェルスプロンプトは `clzc:schost>` です。

表 3-2 タスクマップ: ノードの停止とブート

タスク	ツール	手順
ノードの停止。	グローバルクラスタノードの場合は、 <code>clnode(1CL)</code> <code>evacuate</code> および <code>shutdown</code> を使用します。 ゾーンクラスタノードの場合は、 <code>clzonecluster(1CL)</code> <code>halt</code> を使用します。	72 ページの「ノードを停止する方法」
ノードの起動。 クラスタメンバーシップを取得できるように、ノードにはクラスタインターコネクトとの動作中の接続が必要です。	グローバルクラスタノードの場合は、 <code>boot</code> または <code>b</code> を使用します。 ゾーンクラスタノードの場合は、 <code>clzonecluster(1CL)</code> <code>boot</code> を使用します。	76 ページの「ノードをブートする方法」
クラスタ上のノードをいったん停止してから再起動。 クラスタメンバーシップを取得できるように、ノードにはクラスタインターコネクトとの動作中の接続が必要です。	グローバルクラスタノードの場合は、 <code>clnode</code> <code>evacuate</code> および <code>shutdown</code> を使用してから、 <code>boot</code> または <code>b</code> を使用します。 ゾーンクラスタノードの場合は、 <code>clzonecluster(1CL)</code> <code>reboot</code> を使用します。	78 ページの「ノードをリブートする方法」
ノードがクラスタメンバーシップを取得しないようにノードをブート。	グローバルクラスタノードの場合は、 <code>clnode</code> <code>evacuate</code> および <code>shutdown</code> コマンドを使用してから、 <code>boot -x</code> を使用 (SPARC または x86 の GRUB メニューエントリ編集 で)。 基になるグローバルクラスタが非クラスタモードでブートされる場合は、ゾーンクラスタノードも自動的に非クラスタモードになります。	82 ページの「非クラスタモードでノードをブートする方法」

▼ ノードを停止する方法

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。



注意 - グローバルクラスタやゾーンクラスタ上のノードを停止する場合に、`send brk` をクラスタコンソール上で使用しないでください。この機能はクラスタ内ではサポートされません。

- 1 Oracle RAC が動作しているクラスタの場合は、停止するクラスタ上のデータベースのすべてのインスタンスを停止します。

停止の手順については、Oracle RAC 製品のドキュメントを参照してください。

- 2 停止するクラスタノードで、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.admin` を提供する役割になります。グローバルクラスタのノードから、次の手順のステップをすべて実行します。

- 3 特定のゾーンクラスタメンバーを停止する場合は、手順 4 から 6 をスキップし、グローバルクラスタノードから次のコマンドを実行します。

```
phys-schost# clzonecluster halt -n physical-name zoneclustername
```

特定のゾーンクラスタノードを指定すると、そのノードのみが停止します。halt コマンドは、デフォルトではすべてのノード上のゾーンクラスタを停止します。

- 4 すべてのリソースグループ、リソース、およびデバイスグループを、停止するノードから別のグローバルクラスタノードに切り替えます。

停止するグローバルクラスタノードで、次のようにコマンドを入力します。clnode evacuate コマンドは、すべてのリソースグループおよびデバイスグループ (すべての非大域ゾーンを含む) を、指定したノードから次に優先されるノードに切り替えます。(ゾーンクラスタノード内で clnode evacuate を実行することもできます)。

```
phys-schost# clnode evacuate node
```

node リソースグループとデバイスグループを切り替えるノードを指定します。

- 5 ノードを停止します。

停止するグローバルクラスタノードを指定します。

```
phys-schost# shutdown -g0 -y -i0
```

SPARC ベースのシステムではグローバルクラスタノードが ok プロンプトを表示し、x86 ベースのシステムでは GRUB メニューで「Press any key to continue」というメッセージが表示されていることを確認します。

- 6 必要であればノードの電源を切ります。

例 3-7 SPARC: グローバルクラスタノードの停止

次の例に、ノード `phys-schost-1` がシャットダウンされた場合のコンソール出力を示します。ここでは、`-g0` オプションで猶予期間をゼロに設定し、`-y` オプションで、確認プロンプトに対して自動的に `yes` と応答するよう指定しています。このノードの停止メッセージは、グローバルクラスタ内のほかのノードのコンソールにも表示されます。

```
phys-schost# clnode evacuate phys-schost-1
phys-schost# shutdown -g0 -y
Wed Mar 10 13:47:32 phys-schost-1 cl_runtime:
WARNING: CMM monitoring disabled.
phys-schost-1#
INIT: New run level: 0
The system is coming down. Please wait.
Notice: rgmd is being stopped.
Notice: rpc.pmfd is being stopped.
Notice: rpc.fed is being stopped.
umount: /global/.devices/node@1 busy
umount: /global/phys-schost-1 busy
The system is down.
syncing file systems... done
Program terminated
ok
```

例 3-8 x86: グローバルクラスタノードの停止

次の例に、ノード `phys-schost-1` が停止した場合のコンソール出力を示します。ここでは、`-g0` オプションで猶予期間をゼロに設定し、`-y` オプションで、確認プロンプトに対して自動的に `yes` と応答するよう指定しています。このノードの停止メッセージは、グローバルクラスタ内のほかのノードのコンソールにも表示されます。

```
phys-schost# clnode evacuate phys-schost-1
phys-schost# shutdown -g0 -y
Shutdown started.    Wed Mar 10 13:47:32 PST 2004

Changing to init state 0 - please wait
Broadcast Message from root (console) on phys-schost-1 Wed Mar 10 13:47:32...
THE SYSTEM phys-schost-1 IS BEING SHUT DOWN NOW !!!
Log off now or risk your files being damaged

phys-schost-1#
INIT: New run level: 0
The system is coming down. Please wait.
System services are now being stopped.
/etc/rc0.d/K05initrgm: Calling clnode evacuate
failfasts disabled on node 1
Print services already stopped.
Mar 10 13:47:44 phys-schost-1 syslogd: going down on signal 15
umount: /global/.devices/node@2 busy
umount: /global/.devices/node@1 busy
```

```
The system is down.
syncing file systems... done
WARNING: CMM: Node being shut down.
Type any key to continue
```

例3-9 ゾーンクラスタノードの停止

次の例は、`clzonecluster halt` を使用して *sparse-sczone* というゾーンクラスタ上のノードを停止する方法を示しています (ゾーンクラスタノード内で `clnode evacuate` コマンドと `shutdown` コマンドを実行することもできます)。

```
phys-schost# clzonecluster status
```

```
=== Zone Clusters ===
```

```
--- Zone Cluster Status ---
```

Name	Node Name	Zone HostName	Status	Zone Status
sparse-sczone	schost-1	sczone-1	Online	Running
	schost-2	sczone-2	Online	Running
	schost-3	sczone-3	Online	Running
	schost-4	sczone-4	Online	Running

```
phys-schost#
phys-schost# clzonecluster halt -n schost-4 sparse-sczone
Waiting for zone halt commands to complete on all the nodes of the zone cluster "sparse-sczone"...
Sep  5 19:24:00 schost-4 cl_runtime: NOTICE: Membership : Node 3 of cluster 'sparse-sczone' died.
phys-host#
phys-host# clzonecluster status
```

```
=== Zone Clusters ===
```

```
--- Zone Cluster Status ---
```

Name	Node Name	Zone HostName	Status	Zone Status
sparse-sczone	schost-1	sczone-1	Online	Running
	schost-2	sczone-2	Online	Running
	schost-3	sczone-3	Offline	Installed
	schost-4	sczone-4	Online	Running

```
phys-schost#
```

参照 停止したグローバルクラスタノードを再起動するには、[76 ページの「ノードをブートする方法」](#)を参照してください。

▼ ノードをブートする方法

グローバルクラスタまたはゾーンクラスタの他のアクティブなノードを停止またはリブートする場合は、ブートするノードのマルチユーザーサーバーのマイルストーンがオンラインになるまで待ちます。

ログインプロンプトが表示されてからでなければ、そのノードは、停止またはリブートするクラスタ内の他のノードからサービスを引き継ぎません。非大域ゾーンのブートについては、『[Oracle Solaris の管理: Oracle Solaris コンテナ-リソース管理と Oracle Solaris ゾーン](#)』の第20章「非大域ゾーンのインストール、ブート、停止、アンインストール、および複製(タスク)」を参照してください。

注- ノードの起動は、定足数の構成によって変わる場合があります。2ノードのクラスタでは、クラスタの定足数の合計数が3つになるように定足数デバイスを構成する必要があります(各ノードごとに1つと定足数デバイスに1つ)。この場合、最初のノードを停止しても、2つ目のノードは定足数を保持しており、唯一のクラスタメンバーとして動作します。1番目のノードをクラスタノードとしてクラスタに復帰させるには、2番目のノードが稼動中で必要な数のクラスタ定足数(2つ)が存在している必要があります。

Oracle Solaris Cluster をゲストドメインで実行している場合は、制御ドメインまたは I/O ドメインをリブートすると、ドメインが停止するなど、実行中のゲストドメインが影響を受ける場合があります。制御ドメインまたは I/O ドメインをリブートする前に、ほかのノードとの負荷のバランスを取り直し、Oracle Solaris Cluster を実行しているゲストドメインを停止するようにしてください。

制御ドメインまたは I/O ドメインがリブートされると、ハートビートはゲストドメインによって送受信されません。これにより、スプリットブレインとクラスタ再構成が発生します。制御ドメインまたは I/O ドメインはリブート中のため、ゲストドメインは共有デバイスにアクセスできません。その他のクラスタノードでは、このゲストドメインは共有デバイスから切り離されます。制御ドメインまたは I/O ドメインのリブートが終了すると、ゲストドメインで I/O が再開されますが、ゲストドメインはクラスタ再構成の一部として共有ディスクから切り離されているため、共有ストレージとの間の I/O により、ゲストドメインでパニックが発生します。冗長性のためにゲストが2つの I/O ドメインを使用している場合、I/O ドメインを1つずつリブートすると、この問題を解決できます。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

注- クラスタメンバーシップを取得できるように、ノードにはクラスタインターコネクトとの動作中の接続が必要です。

- 1 停止したグローバルクラスタノードやゾーンクラスタノードを起動するために、そのノードを起動します。グローバルクラスタのノードから、次の手順のステップをすべて実行します。

- SPARC ベースのシステム上で、次のコマンドを実行します。

```
ok boot
```

- x86 ベースのシステム上で、次のコマンドを実行します。

GRUB メニューが表示された時点で、適切な Oracle Solaris エントリを選択し、Enter キーを押します。次のような GRUB メニューが表示されます。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
```

```
+-----+
| Solaris 10 /sol_10_x86          |
| Solaris failsafe                |
|                                 |
+-----+
```

```
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

クラスタコンポーネントがブートすると、ブートされたノードのコンソールにメッセージが表示されます。

- ゾーンクラスタがある場合は、ブートするノードを指定できます。

```
phys-schost# clzonecluster boot -n node zoneclustername
```

- 2 ノードが問題なくブートし、オンラインであることを確認します。

- **cluster status** コマンドを実行すると、グローバルクラスタノードのステータスが報告されます。

```
phys-schost# cluster status -t node
```

- **clzonecluster status** コマンドをグローバルクラスタ上のノードから実行すると、すべてのゾーンクラスタノードのステータスが報告されます。

```
phys-schost# clzonecluster status
```

ホストのノードがクラスタモードでブートされる場合は、ゾーンクラスタノードもクラスタモードのみでブートできます。

注 - ノードの `/var` ファイルシステムが満杯になると、そのノード上では Oracle Solaris Cluster が再起動できなくなる可能性があります。この問題が発生した場合は、[86 ページの「満杯の `/var` ファイルシステムを修復する方法」](#)を参照してください。

例 3-10 SPARC: グローバルクラスタノードのブート

次に、ノード `phys-schost-1` をブートしてグローバルクラスタに結合させたときのコンソールの出力例を示します。

```
ok boot
Rebooting with command: boot
...
Hostname: phys-schost-1
Booting as part of a cluster
...
NOTICE: Node phys-schost-1: attempting to join cluster
...
NOTICE: Node phys-schost-1: joined cluster
...
The system is coming up. Please wait.
checking ufs filesystems
...
reservation program successfully exiting
Print services started.
volume management starting.
The system is ready.
phys-schost-1 console login:
```

▼ ノードをリブートする方法

グローバルクラスタまたはゾーンクラスタの他のアクティブなノードを停止またはリブートするには、リブートするノードのマルチユーザーサーバーのマイルストーンがオンラインになるまで待ちます。

ログインプロンプトが表示されてからでなければ、そのノードは、停止またはリブートするクラスタ内の他のノードからサービスを引き継げません。非大域ゾーンのリブートについては、『[Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン](#)』の第 20 章「[非大域ゾーンのインストール、ブート、停止、アンインストール、および複製 \(タスク\)](#)」を参照してください。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。



注意 - リソースのメソッドのいずれかがタイムアウトして強制終了できなかった場合、リソースの `Failover_mode` プロパティが `HARD` に設定されているときに限り、ノードがリブートされます。 `Failover_mode` プロパティがそれ以外の値に設定されている場合、ノードはリブートされません。

- 1 グローバルクラスタノードまたはゾーンクラスタノードで **Oracle RAC** を実行している場合は、停止するノード上のデータベースのすべてのインスタンスを停止します。停止の手順については、Oracle RAC 製品のドキュメントを参照してください。
- 2 停止するノードで、スーパーユーザーになるか、**RBAC** の承認 `solaris.cluster.admin` を提供する役割になりますグローバルクラスタのノードから、次の手順のステップをすべて実行します。
- 3 **clnode evacuate** および **shutdown** コマンドを使用して、グローバルクラスタノードを停止します。グローバルクラスタのノード上で実行する **clzonecluster halt** コマンドで、ゾーンクラスタを停止します。(**clnode evacuate** コマンドと **shutdown** コマンドもゾーンクラスタ内で動作します)。

グローバルクラスタの場合は、停止するノードで次のコマンドを入力します。 **clnode evacuate** コマンドは、すべてのデバイスグループを、指定のノードから次に優先されるノードに切り替えます。このコマンドはまた、すべてのリソースグループを、指定したノード上の大域ゾーンまたは非大域ゾーンから、ほかのノード上の次に優先される大域ゾーンまたは非大域ゾーンに切り替えます。

注 - ノードを個別にシャットダウンするには、 **shutdown -g0 -y -i6** コマンドを使用します。複数のノードを一括してシャットダウンするには、 **shutdown -g0 -y -i0** コマンドを使用してノードを停止します。すべてのノードが停止した後は、すべてのノードに対して **boot** コマンドを使用して、すべてのノードをブートしてクラスタに戻します。

- SPARC ベースのシステムの場合、ノードを個別にリブートするには、次のコマンドを実行します。

```
phys-schost# clnode evacuate node
phys-schost# shutdown -g0 -y -i6
```

- x86 ベースのシステムの場合、ノードを個別にリブートするには、次のコマンドを実行します。

```
phys-schost# clnode evacuate node
phys-schost# shutdown -g0 -y -i6
```

GRUB メニューが表示された時点で、適切な Oracle Solaris エントリを選択し、Enter キーを押します。次のような GRUB メニューが表示されます。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-----+
| Solaris 10 /sol_10_x86                |
| Solaris failsafe                      |
|                                     |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

- 停止し、リブートするゾーンクラスタノードを指定します。

```
phys-schost# clzonecluster reboot - node zoneclustername
```

注-クラスタメンバーシップを取得できるように、ノードにはクラスタインターコネクトとの動作中の接続が必要です。

4 ノードが問題なくブートし、オンラインであることを確認します。

- グローバルクラスタノードがオンラインであることを確認します。

```
phys-schost# cluster status -t node
```

- ゾーンクラスタノードがオンラインであることを確認します。

```
phys-schost# clzonecluster status
```

例 3-11 SPARC: グローバルクラスタノードのリブート

次の例に、ノード `phys-schost-1` がリブートした場合のコンソール出力を示します。このノードの停止時および起動時の通知メッセージは、グローバルクラスタ内のほかのノードのコンソールに表示されます。

```
phys-schost# clnode evacuate phys-schost-1
phys-schost# shutdown -g0 -y -i6
Shutdown started.    Wed Mar 10 13:47:32 phys-schost-1 cl_runtime:
```

```
WARNING: CMM monitoring disabled.
phys-schost-1#
INIT: New run level: 6
The system is coming down. Please wait.
System services are now being stopped.
Notice: rgmd is being stopped.
Notice: rpc.pmfd is being stopped.
Notice: rpc.fed is being stopped.
umount: /global/.devices/node@1 busy
umount: /global/phys-schost-1 busy
The system is down.
syncing file systems... done
rebooting...
```



```
Resetting ...
'''
Sun Ultra 1 SBus (UltraSPARC 143MHz), No Keyboard
OpenBoot 3.11, 128 MB memory installed, Serial #5932401.
Ethernet address 8:8:20:99:ab:77, Host ID: 8899ab77.
...
Rebooting with command: boot
...
Hostname: phys-schost-1
Booting as part of a cluster
...
NOTICE: Node phys-schost-1: attempting to join cluster
...
NOTICE: Node phys-schost-1: joined cluster
...
The system is coming up. Please wait.
The system is ready.
phys-schost-1 console login:
```

例 3-12 x86: グローバルクラスタノードのリブート

次の例に、ノード `phys-schost-1` をリブートする際のコンソール出力を示します。このノードの停止時および起動時の通知メッセージは、グローバルクラスタ内のほかのノードのコンソールに表示されます。

```
phys-schost# clnode evacuate phys-schost-1
phys-schost # shutdown -g0 -i6 -y

GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-----+
| Solaris 10 /sol_10_x86                               |
| Solaris failsafe                                     |
|                                                       |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
Hostname: phys-schost-1
Booting as part of a cluster
...
NOTICE: Node phys-schost-1: attempting to join cluster
...
NOTICE: Node phys-schost-1: joined cluster
...
The system is coming up. Please wait.
checking ufs filesystems
...
reservation program successfully exiting
Print services started.
volume management starting.
The system is ready.
phys-schost-1 console login:
```

例 3-13 ゾーンクラスタノードのリブート

次の例は、ゾーンクラスタ上のノードをリブートする方法を示しています。

```
phys-schost# clzonecluster reboot -n schost-4 sparse-sczone
Waiting for zone reboot commands to complete on all the nodes of the zone cluster
"sparse-sczone"...
Sep  5 19:40:59 schost-4 cl_runtime: NOTICE: Membership : Node 3 of cluster
'sparse-sczone' died.
phys-schost# Sep  5 19:41:27 schost-4 cl_runtime: NOTICE: Membership : Node 3 of cluster
'sparse-sczone' joined.

phys-schost#
phys-schost# clzonecluster status

=== Zone Clusters ===

--- Zone Cluster Status ---
Name           Node Name      Zone HostName  Status  Zone Status
-----
sparse-sczone  schost-1       sczone-1       Online  Running
                schost-2       sczone-2       Online  Running
                schost-3       sczone-3       Online  Running
                schost-4       sczone-4       Online  Running

phys-schost#
```

▼ 非クラスタモードでノードをブートする方法

グローバルクラスタノードは、非クラスタモードでブートできます(その場合は、ノードがクラスタメンバーシップに参加しません)。非クラスタモードは、クラスタソフトウェアをインストールしたり、ノードにパッチを適用するなどの管理手順を実行したりするときに役立ちます。ゾーンクラスタノードは、その基になるグローバルクラスタノードの状態と異なる状態ではブートできません。グローバルクラスタノードが、非クラスタモードでブートすると、ゾーンクラスタノードも自動的に非クラスタモードになります。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 非クラスタモードで起動するクラスタ上で、スーパーユーザーになるか、RBAC の承認 **solaris.cluster.admin** を提供する役割になります。グローバルクラスタのノードから、次の手順のステップをすべて実行します。

- 2 グローバルクラスタのノードで **clzonecluster halt** コマンドを実行し、ゾーンクラスタノードをシャットダウンします。**clnode evacuate** および **shutdown** コマンドを使用して、グローバルクラスタノードを停止します。

clnode evacuate コマンドは、すべてのデバイスグループを、指定のノードから次に優先されるノードに切り替えます。このコマンドはまた、すべてのリソースグループを、指定したノード上の大域ゾーンまたは非大域ゾーンから、ほかのノード上の次に優先される大域ゾーンまたは非大域ゾーンに切り替えます。

- 特定のグローバルクラスタをシャットダウンします。

```
phys-schost# clnode evacuate node
```

```
phys-schost# shutdown -g0 -y
```

- グローバルクラスタノードから特定のゾーンクラスタノードを停止します。

```
phys-schost# clzonecluster halt -n node zoneclustername
```

ゾーンクラスタ内で **clnode evacuate** コマンドと **shutdown** コマンドを使用することもできます。

- 3 **Oracle Solaris** ベースのシステムではグローバルクラスタノードが **ok** プロンプトを表示し、**x86** ベースのシステムでは **GRUB** メニューで「**Press any key to continue**」というメッセージが表示されていることを確認します。

- 4 非クラスタモードでグローバルクラスタノードをブートします。

- **SPARC** ベースのシステム上で、次のコマンドを実行します。

```
ok boot -xs
```

- **x86** ベースのシステム上で、次のコマンドを実行します。

- a. **GRUB** メニューで矢印キーを使用して該当する **Oracle Solaris** エントリを選択し、**e** と入力してコマンドを編集します。

次のような **GRUB** メニューが表示されます。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
```

```
+-----+
| Solaris 10 /sol_10_x86          |
| Solaris failsafe                 |
|                                 |
+-----+
```

```
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

GRUB ベースのブートの詳細については、『[Oracle Solaris の管理: 基本管理](#)』の「[GRUB を使用して x86 システムをブートする \(タスクマップ\)](#)」を参照してください。

- b. ブートパラメータの画面で、矢印キーを使用してカーネルエントリを選択し、**e**を入力してエントリを編集します。

次のような GRUB ブートパラメータの画面が表示されます。

```
GNU GRUB version 0.95 (615K lower / 2095552K upper memory)
+-----+
| root (hd0,0,a)                                |
| kernel /platform/i86pc/multiboot              |
| module /platform/i86pc/boot_archive           |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.
```

- c. コマンドに **-x** を追加して、システムが非クラスタモードでブートするように指定します。

```
[ Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time exits. ]
```

```
grub edit> kernel /platform/i86pc/multiboot -x
```

- d. **Enter** キーを押して変更を受け入れ、ブートパラメータの画面に戻ります。
画面には編集されたコマンドが表示されます。

```
GNU GRUB version 0.95 (615K lower / 2095552K upper memory)
+-----+
| root (hd0,0,a)                                |
| kernel /platform/i86pc/multiboot -x          |
| module /platform/i86pc/boot_archive         |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.-
```

- e. **b** と入力して、ノードを非クラスタモードでブートします。

注-カーネルブートパラメータコマンドへのこの変更は、システムをブートすると無効になります。次にノードをリブートする際には、ノードはクラスタモードでブートします。クラスタモードではなく、非クラスタモードでブートするには、これらの手順を再度実行して、カーネルブートパラメータコマンドに **-x** オプションを追加します。

例 3-14 SPARC: 非クラスタモードでグローバルクラスタノードをブートする

次に、ノード `phys-schost-1` を停止し、非クラスタモードで再起動した場合のコンソール出力の例を示します。ここでは、`-g0` オプションで猶予期間をゼロに設定し、`-y` オプションで、確認プロンプトに対して自動的に `yes` と応答するよう指定し、`-i0` で実行レベル 0 (ゼロ) で起動します。このノードの停止メッセージは、グローバルクラスタ内のほかのノードのコンソールにも表示されます。

```
phys-schost# clnode evacuate phys-schost-1
phys-schost# cluster shutdown -g0 -y
Shutdown started.    Wed Mar 10 13:47:32 phys-schost-1 cl_runtime:

WARNING: CMM monitoring disabled.
phys-schost-1#
...
rg_name = schost-sa-1 ...
offline node = phys-schost-2 ...
num of node = 0 ...
phys-schost-1#
INIT: New run level: 0
The system is coming down. Please wait.
System services are now being stopped.
Print services stopped.
syslogd: going down on signal 15
...
The system is down.
syncing file systems... done
WARNING: node phys-schost-1 is being shut down.
Program terminated

ok boot -x
...
Not booting as part of cluster
...
The system is ready.
phys-schost-1 console login:
```

満杯の /var ファイルシステムを修復する

Oracle Solaris ソフトウェアと Oracle Solaris Cluster ソフトウェアはどちらも、`/var/adm/messages` ファイルにエラーメッセージを書き込みます。このため時間が経過すると、`/var` ファイルシステムが満杯になる可能性があります。クラスタノードの `/var` ファイルシステムが満杯になると、そのノード上では Oracle Solaris Cluster が再起動できなくなる可能性があります。また、そのノードにログインできなくなる可能性もあります。

▼ 満杯の /var ファイルシステムを修復する方法

/var ファイルシステムが満杯になったことがノードによって報告され、Oracle Solaris Cluster サービスが引き続き実行されているときは、次の手順で、満杯になったファイルシステムを整理してください。詳細については、『[Solaris のシステム管理 \(上級編\)](#)』の「[システムメッセージの表示](#)」を参照してください。

- 1 満杯の /var ファイルシステムが存在するクラスタノードでスーパーユーザーになります。
- 2 満杯のファイルシステムを整理します。
たとえば、ファイルシステムにある重要ではないファイルを削除します。

データ複製のアプローチ

この章では、Oracle Solaris Cluster ソフトウェアで利用できるデータ複製技術について説明します。データ複製は、プライマリストレージデバイスからバックアップデバイス (セカンダリデバイス) へのデータのコピーとして定義されます。プライマリデバイスに障害が発生した場合も、セカンダリデバイスからデータを使用できます。データ複製を使用すると、クラスタの高可用性と耐障害性を確保できます。

Oracle Solaris Cluster ソフトウェアは、次のデータ複製タイプをサポートします。

- クラスタ間 - 障害回復には、Oracle Solaris Cluster Geographic Edition を使用します
- クラスタ内 - キャンパスクラスタ内でホストベースのミラーリングの代替として使用します

データ複製を実行するには、複製するオブジェクトと同じ名前のデバイスグループが必要です。デバイスは、一度に 1 つのデバイスグループのみに属することができるため、デバイスを含む Oracle Solaris Cluster デバイスグループがすでにある場合、そのデバイスを新しいデバイスグループに追加する前にそのグループを削除する必要があります。Solaris Volume Manager、ZFS、または raw ディスクデバイスグループの作成および管理については、第 5 章の [122 ページ](#) の「[デバイスグループの管理](#)」を参照してください。

クラスタに最適なサービスを提供する複製アプローチを選択するには、ホストベースとストレージベースのデータ複製を両方とも理解しておく必要があります。障害回復でデータ複製を管理するための Oracle Solaris Cluster Geographic Edition の使用の詳細は、『[Oracle Solaris Cluster Geographic Edition Overview](#)』を参照してください。

この章で説明する内容は次のとおりです。

- [88 ページ](#) の「[データ複製についての理解](#)」
- [90 ページ](#) の「[クラスタ内でのストレージベースのデータ複製の使用](#)」

データ複製についての理解

Oracle Solaris Cluster は、データ複製で次のアプローチをサポートしています。

- ホストベースのデータ複製は、ソフトウェアを使用して、地理的に離れたクラスタとの間でリアルタイムにディスクボリュームを複製します。リモートミラー複製を使用すると、プライマリクラスタのマスターボリュームのデータを、地理的に離れたセカンダリクラスタのマスターボリュームに複製できます。リモートミラービットマップは、プライマリディスク上のマスターボリュームと、セカンダリディスク上のマスターボリュームの差分を追跡します。クラスタ間(およびクラスタとクラスタ内にないホストとの間)の複製に使用されるホストベースの複製ソフトウェアには、Sun StorageTek Availability Suite 4 などがあります。

ホストベースのデータ複製は、特別なストレージレイではなくホストリソースを使用するため、より費用のかからないデータ複製ソリューションです。Oracle Solaris OS を実行する複数のホストが共有ボリュームにデータを書き込むことができるように構成されているデータベース、アプリケーション、またはファイルシステムはサポートされていません (Oracle 9iRAC や Oracle Parallel Server など)。2 つのクラスタ間でのホストベースのデータ複製の使用に関する詳細は、『[Oracle Solaris Cluster Geographic Edition Data Replication Guide for Sun StorageTek Availability Suite](#)』を参照してください。Oracle Solaris Cluster Geographic Edition を使用しないホストベースの複製の例については、付録 A の 337 ページの「[Availability Suite ソフトウェアを使用したホストベースのデータ複製の構成](#)」を参照してください。

- ストレージベースのデータ複製では、ストレージコントローラ上でソフトウェアを使用することで、データ複製の作業をクラスタノードからストレージデバイスに移動します。このソフトウェアはノードの処理能力を一部解放し、クラスタのリクエストに対応します。クラスタ内またはクラスタ間でデータを複製できるストレージベースのソフトウェアの例には、Hitachi TrueCopy、HITACHI Universal Replicator、EMC SRDF が含まれます。ストレージベースのデータ複製は、キャンパスクラスタ構成において特に重要になる場合があり、必要なインフラストラクチャを簡素化できます。キャンパスクラスタ環境でストレージベースのデータ複製を使用する方法の詳細は、90 ページの「[クラスタ内でのストレージベースのデータ複製の使用](#)」を参照してください。

2 つ以上のクラスタ間でのストレージベースの複製の使用と、プロセスを自動化する Oracle Solaris Cluster Geographic Edition 製品の詳細については、『[Oracle Solaris Cluster Geographic Edition Data Replication Guide for Hitachi TrueCopy and Universal Replicator](#)』および『[Oracle Solaris Cluster Geographic Edition Data Replication Guide for EMC Symmetrix Remote Data Facility](#)』を参照してください。このタイプのクラスタ構成の完全な例については、付録 A の 337 ページの「[Availability Suite ソフトウェアを使用したホストベースのデータ複製の構成](#)」も参照してください。

サポートされるデータ複製方式

Oracle Solaris Cluster ソフトウェアは、クラスタ間またはクラスタ内のデータ複製方式として以下をサポートしています。

1. クラスタ間の複製 - 障害回復の場合、クラスタ間のデータ複製を実行するためにホストベース、ストレージベース、またはアプリケーションベースの複製を使用できます。一般的には、3つの方法の組み合わせではなく、1つの複製方法を選択します。Oracle Solaris Cluster Geographic Edition ソフトウェアでは、すべての種類の複製を管理できます。

- ホストベースの複製
 - Sun StorageTek Availability Suite、Oracle Solaris 10 OS 以降

Oracle Solaris Cluster Geographic Edition ソフトウェアを使用しないでホストベースの複製を使用する場合は、付録 A 「例」 の337 ページの「[Availability Suite](#) ソフトウェアを使用したホストベースのデータ複製の構成」を参照してください。

- ストレージベースの複製
 - Hitachi TrueCopy および Hitachi Universal Replicator、Oracle Solaris Cluster Geographic Edition 経由
 - EMC Symmetrix Remote Data Facility (SRDF)、Oracle Solaris Cluster Geographic Edition 経由
 - Oracle 社製 Sun ZFS Storage Appliance。詳細は、『[Oracle Solaris Cluster Geographic Edition Overview](#)』の「[Data Replication](#)」を参照してください。

Oracle Solaris Cluster Geographic Edition ソフトウェアを使用せずにストレージベースの複製を使用する場合は、複製ソフトウェアのドキュメントを参照してください。

- アプリケーションベースのレプリケーション
 - Oracle Data Guard
 - MySQL

障害回復の詳細については、『[Oracle Solaris Cluster Geographic Edition Data Replication Guide for Oracle Data Guard](#)』を参照してください。

2. クラスタ内の複製 - この方式は、ホストベースのミラー化の代替として使用されます。
 - ストレージベースの複製
 - Hitachi TrueCopy および Hitachi Universal Replicator
 - EMC Symmetrix Remote Data Facility (SRDF)

クラスタ内でのストレージベースのデータ複製の使用

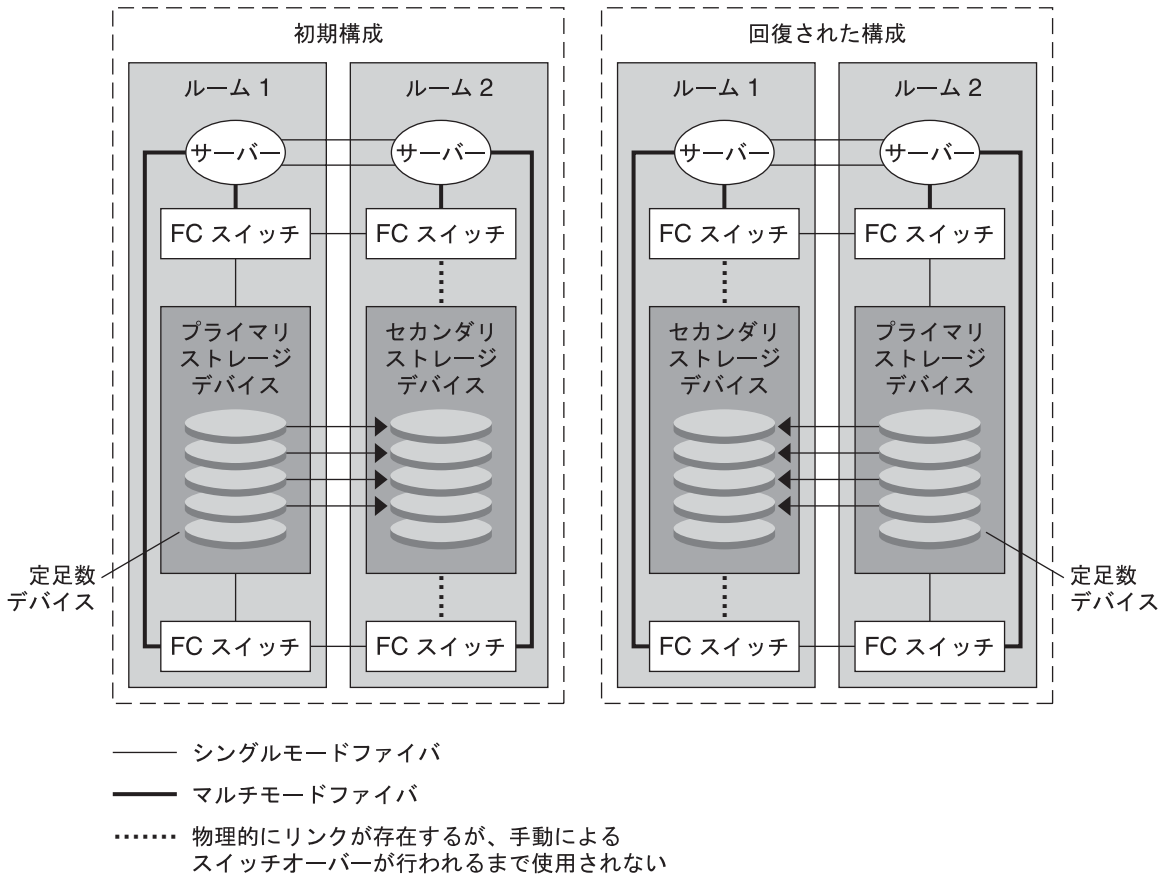
ストレージベースのデータ複製は、ストレージデバイスにインストールされているソフトウェアを使用して、クラスタまたはキャンパスクラスタ内の複製を管理します。このようなソフトウェアは、特定のストレージデバイスに固有で、障害回復には使用されません。ストレージベースのデータ複製を構成する際には、ストレージデバイスに付属するドキュメントを参照してください。

使用するソフトウェアに応じて、ストレージベースのデータ複製を使用して自動と手動のいずれかのフェイルオーバーを使用できます。Oracle Solaris Cluster では、Hitachi TrueCopy、Hitachi Universal Replicator、および EMC SRDF ソフトウェアを使用した複製で手動と自動の両方のフェイルオーバーをサポートしています。

このセクションでは、キャンパスクラスタで使用されるストレージベースのデータ複製について説明します。[図 4-1](#) に、2つのストレージアレイ間でデータが複製される、2ルーム構成の例を示します。この構成では、最初のルームにプライマリストレージアレイがあり、これが両方のルームのノードにデータを提供します。また、プライマリストレージアレイは、複製するデータをセカンダリストレージアレイに提供します。

注 - [図 4-1](#) は、複製されていないボリューム上に定足数デバイスがあることを示しています。複製されたボリュームを定足数デバイスとして使用することはできません。

図 4-1 ストレージベースのデータ複製を装備した2ルーム構成



Hitachi TrueCopy または Hitachi Universal Replicator を使用したストレージベースのデータ複製は、Oracle Solaris Cluster 環境では使用するアプリケーションの種類に応じて、同期または非同期で実行できます。キャンパスクラスタで自動フェイルオーバーを実行する場合は、TrueCopy を同期的に使用します。EMC SRDF を使用したストレージベースの同期複製は、Oracle Solaris Cluster でサポートされますが、非同期複製は EMC SRDF でサポートされません。

EMC SRDF のドミノモードまたは適応型コピーモードを使用しないでください。ドミノモードでは、ターゲットが使用可能でない場合に、ローカルおよびターゲット SRDF ボリュームをホストで使用できなくなります。適応型コピーモードは、一般的にデータ移行およびデータセンター移行に使用され、障害回復には推奨されません。

リモートストレージデバイスとの通信が失われた場合は、never または async の Fence_level を指定して、プライマリクラスタ上で実行されているアプリケーション

がブロックされないようにしてください。data または status の Fence_level を指定すると、リモートストレージデバイスに更新がコピーできない場合に、プライマリストレージデバイスが更新を拒否します。

クラスタ内でストレージベースのデータ複製を使用する際の要件と制限

データの整合性を確保するためには、マルチパスと適切な RAID パッケージを使用してください。次のリストには、ストレージベースのデータ複製を使用するクラスタ構成を実装するための考慮事項が含まれています。

- ノード間の距離は、Oracle Solaris Cluster Fibre Channel とインターコネクティブラストラクチャーにより制限されます。現在の制限とサポートされる技術の詳細については、Oracle のサービスプロバイダにお問い合わせください。
- 複製されたボリュームを、定足数デバイスとして構成しないでください。共有の複製されていないボリュームにある定足数デバイスを見つけるか、定足数サーバーを使用します。
- データのプライマリコピーのみがクラスタノードに認識されるようにします。それ以外の場合、ボリュームマネージャーはデータのプライマリコピーとセカンダリコピーの両方に同時にアクセスしようとする場合があります。データコピーの可視性の制御については、ストレージアレイに付属するドキュメントを参照してください。
- EMC SRDF、Hitachi TrueCopy、および Hitachi Universal Replicator では、複製されるデバイスのグループを定義できます。各複製デバイスグループには、同じ名前の Oracle Solaris Cluster デバイスグループが必要です。
- EMC SRDF を使用し、並列または直列接続された RDF デバイスを備える、3つのサイトまたは3つのデータセンターがある構成の場合、参加するすべてのクラスタノードで、Solutions Enabler の SYMCLI オプションファイルに次のエントリを追加する必要があります。

```
SYMAPI_2SITE_CLUSTER_DG=device-group:rdg-group-number
```

このエントリにより、クラスタソフトウェアで、2つの SRDF 同期サイト間でのアプリケーションの移動を自動化できます。エントリ内の *rdg-group-number* は、ホストのローカル Symmetrix を 2 番目のサイトの Symmetrix に接続する RDF グループを表します。

3つのデータセンター構成の詳細については、『[Oracle Solaris Cluster Geographic Edition Overview](#)』の「[Three-Data-Center \(3DC\) Topologies](#)」を参照してください。

- 特定のアプリケーション固有のデータは、非同期データ複製には適さない場合があります。アプリケーションの動作に関する知識を使って、ストレージデバイス間でアプリケーション固有のデータを複製する最善の方法を決定します。
- クラスタを自動フェイルオーバー用に構成する場合は、同期複製を使用します。

複製されたボリュームの自動フェイルオーバー用にクラスタを構成する手順については、[97 ページの「ストレージベースの複製されたデバイスの管理」](#)を参照してください。

- Oracle Real Application Clusters (RAC) は、クラスタ内で複製を行う場合、SRDF、Hitachi TrueCopy、および Hitachi Universal Replicator をサポートしません。現在プライマリ複製ではない複製に接続されたノードには、書き込みアクセス権はありません。クラスタのすべてのノードからの直接書き込みアクセス権が必要なスケラブルアプリケーションは、複製されるデバイスでサポートできません。
- Oracle Solaris Cluster ソフトウェア用の複数所有者 Solaris Volume Manager はサポートされていません。
- EMC SRDF でドミノモードまたは適応型コピーモードを使用しないでください。詳細は、[90 ページの「クラスタ内でのストレージベースのデータ複製の使用」](#)を参照してください。
- Hitachi TrueCopy または Hitachi Universal Replicator では、データモードまたはステータスモードを使用しないでください。詳細は、[90 ページの「クラスタ内でのストレージベースのデータ複製の使用」](#)を参照してください。

クラスタ内でストレージベースのデータ複製を使用する際の手動回復に関する懸念事項

すべてのキャンパスクラスタと同じように、ストレージベースのデータ複製を使用するクラスタでは、通常、1つの障害が発生した場合は介入の必要はありません。ただし、手動フェイルオーバーを使用していて、プライマリストレージデバイスを保持するルームが失われた場合 ([図 4-1](#) を参照)、2 ノードクラスタでは問題が発生します。残ったノードは定足数デバイスを予約できず、またクラスタメンバーとしてブートできません。このような状況では、クラスタで次の手動介入が必要になります。

1. クラスタメンバーとしてブートするよう、Oracle のサービスプロバイダが残りのノードを再構成する必要があります。
2. ユーザーまたは Oracle のサービスプロバイダが、セカンダリストレージデバイスの複製されてないボリュームを定足数デバイスとして構成する必要があります。
3. セカンダリストレージデバイスをプライマリストレージとして使用できるよう、ユーザーまたは Oracle のサービスプロバイダが残りのノードを構成する必要があります。このような再構成には、ボリュームマネージャーボリュームの再構築、データの復元、ストレージボリュームとアプリケーションの関連付けの変更が含まれます。

ストレージベースのデータ複製を使用する際のベストプラクティス

ストレージベースのデータ複製のために、Hitachi TrueCopy または Hitachi Universal Replicator ソフトウェアを使用するデバイスグループを設定するときは、次のベストプラクティスに従ってください。

- プライマリサイトで障害が発生した場合のデータ損失の可能性を回避するため、同期複製を使用します。
- Oracle Solaris Cluster のグローバルデバイスグループと、horcm 構成ファイルで定義される TrueCopy 複製グループとの間に、1 対 1 の関係が存在するようにしてください。これにより、両方のグループが 1 つの単位でノード間を移動できます。
- グローバルファイルシステムのボリュームとフェイルオーバーファイルシステムのボリュームは、制御方法が異なるため、同一の複製されたデバイスグループ内に混在させることはできません。グローバルファイルシステムは DCS (Device Configuration System) によって制御されますが、フェイルオーバーファイルシステムのボリュームは HAS+ によって制御されます。それぞれのプライマリは異なるノードである可能性があり、どちらのノードが複製のプライマリであるかについて競合が発生します。
- すべての RAID マネージャーインスタンスが常に稼働中である必要があります。

EMC SRDF ソフトウェアをストレージベースのデータ複製のために使用する場合は、静的デバイスではなく動的デバイスを使用してください。静的デバイスでは複製プライマリを変更するのに数分かかり、フェイルオーバー時間に影響を与えることがあります。

グローバルデバイス、ディスクパス監視、およびクラスタファイルシステムの管理

この章では、グローバルデバイス、ディスクパス監視、およびクラスタファイルシステムの管理手順について説明します。

- 95 ページの「グローバルデバイスとグローバルな名前空間の管理の概要」
- 97 ページの「ストレージベースの複製されたデバイスの管理」
- 121 ページの「クラスタファイルシステムの管理の概要」
- 122 ページの「デバイスグループの管理」
- 148 ページの「ストレージデバイス用の SCSI プロトコル設定の管理」
- 153 ページの「クラスタファイルシステムの管理」
- 160 ページの「ディスクパス監視の管理」

この章の関連手順の詳細は、表 5-4 を参照してください。

グローバルデバイス、グローバル名前空間、デバイスグループ、ディスクパス監視、およびクラスタファイルシステムに関連する概念については、『[Oracle Solaris Cluster Concepts Guide](#)』を参照してください。

グローバルデバイスとグローバルな名前空間の管理の概要

Oracle Solaris Cluster デバイスグループの管理は、クラスタにインストールされているボリュームマネージャーによって異なります。Solaris Volume Manager は「クラスタ対応」であるため、Solaris Volume Manager の `metaset(1M)` コマンドを使用してデバイスグループを追加、登録、および削除します。

注- グローバルクラスタの非投票ノードからグローバルデバイスに直接アクセスすることはできません。

Oracle Solaris Cluster ソフトウェアは、クラスタ内のディスクデバイスやテープデバイスごとに、raw ディスクデバイスグループを自動的に作成します。ただし、クラスタデバイスグループはグローバルデバイスとしてアクセスされるまでオフラインのままです。デバイスグループやボリューム管理ソフトウェアのディスクグループを管理する際は、グループのプライマリノードであるクラスタから実行する必要があります。

グローバルな名前空間はインストール中に自動的に設定され、Oracle Solaris OS のリブート中に自動的に更新されるため、通常、グローバルデバイス名前空間を管理する必要はありません。ただし、グローバルな名前空間を更新する必要がある場合は、任意のクラスタノードから `cldevice populate` コマンドを実行できます。このコマンドにより、その他のすべてのクラスタノードだけでなく、今後クラスタに参加する可能性があるノードでもグローバルな名前空間を更新できます。

Solaris Volume Manager のグローバルデバイスのアクセス権

グローバルデバイスのアクセス権に加えた変更は、Solaris Volume Manager およびディスクデバイスのクラスタのすべてのノードには自動的に伝達されません。グローバルデバイスのアクセス権を変更する場合は、クラスタ内のすべてのノードで手作業でアクセス権を変更する必要があります。たとえば、グローバルデバイス `/dev/global/dsk/d3s0` のアクセス権を 644 に変更する場合は、クラスタ内のすべてのノード上で次のコマンドを実行します。

```
# chmod 644 /dev/global/dsk/d3s0
```

グローバルデバイスでの動的再構成

クラスタ内のディスクデバイスやテープデバイス上で動的再構成 (DR) を実行するときには、いくつかの問題を考慮する必要があります。

- Oracle Solaris Cluster の動的再構成 (DR) のサポートには、Oracle Solaris の DR 機能に述べられている必要条件、手順、および制限がすべて適用されます。ただし、オペレーティングシステムの休止操作は除きます。したがって、Oracle Solaris Cluster ソフトウェアで DR 機能を使用する前に、Oracle Solaris の DR 機能についての説明を参照してください。特に、DR の切り離し操作中に、ネットワークに接続されていない入出力デバイスに影響する問題について確認してください。
- Oracle Solaris Cluster は、プライマリノードのアクティブなデバイス上で DR 削除操作を実行できません。DR 操作を実行できるのは、プライマリノードのアクティブでないデバイスか、セカンダリノードの任意のデバイス上だけです。
- DR 操作が終了すると、クラスタのデータアクセスが前と同じように続けられます。

- Oracle Solaris Cluster は、定足数デバイスの使用に影響を与える DR 操作を拒否します。詳細については、171 ページの「定足数デバイスへの動的再構成」を参照してください。



注意 - セカンダリノードに対して DR 操作を行なっているときに現在のプライマリノードに障害が発生すると、クラスタの可用性が損なわれます。新しいセカンダリノードが提供されるまで、プライマリノードにはフェイルオーバーする場所がありません。

グローバルデバイス上で DR 操作を実行するには、次の手順をその順番どおりに行います。

表 5-1 タスクマップ: ディスクデバイスとテープデバイスでの動的再構成

タスク	説明
1. アクティブなデバイスグループに影響するような DR 操作を現在のプライマリノードに実行する必要がある場合、DR 削除操作をデバイス上で実行する前に、プライマリノードとセカンダリノードの切替えを実行	145 ページの「デバイスグループのプライマリノードを切り替える」
2. 削除するデバイス上で DR 削除操作を実行	

ストレージベースの複製されたデバイスの管理

ストレージベースの複製を使用して複製されるデバイスが含まれるように、Oracle Solaris Cluster デバイスグループを構成できます。Oracle Solaris Cluster ソフトウェアは、ストレージベースの複製用に Hitachi TrueCopy および EMC Symmetrix Remote Data Facility ソフトウェアをサポートしています。

Hitachi TrueCopy または EMC Symmetrix Remote Data Facility ソフトウェアでデータを複製する前に、ストレージベースの複製のドキュメントによく目を通し、ストレージベースの複製製品と最新のパッチをシステムにインストールしておいてください。ストレージベースの複製ソフトウェアのインストールについては、製品のドキュメントを参照してください。

ストレージベースの複製ソフトウェアは、デバイスのペアを複製として構成する際、一方のデバイスをプライマリ複製、もう一方のデバイスをセカンダリ複製とします。一方のノードのセットに接続されたデバイスが、常にプライマリ複製になります。もう一方のノードのセットに接続されたデバイスは、セカンダリ複製になります。

Oracle Solaris Cluster 構成では、複製が属する Oracle Solaris Cluster デバイスグループが移動されると、常に、プライマリ複製が自動的に移動されます。そのため、Oracle Solaris Cluster 構成下では、プライマリ複製を直接移動しないでください。その代わりに、テイクオーバーは関連する Oracle Solaris Cluster デバイスグループを移動することによって行うようにしてください。



注意 - 作成する Oracle Solaris Cluster デバイスグループ (Solaris Volume Manager、または raw ディスク) の名前は、複製されたデバイスグループと同じ名前にする必要があります。

ここでは、次の手順について説明します。

- 98 ページの「Hitachi TrueCopy で複製されたデバイスの管理」
- 109 ページの「EMC Symmetrix Remote Data Facility で複製されたデバイスの管理」

Hitachi TrueCopy で複製されたデバイスの管理

次の表に、Hitachi TrueCopy ストレージベースの複製されたデバイスを設定するために実行する必要のあるタスクを示します。

表 5-2 タスクマップ: Hitachi TrueCopy ストレージベースの複製されたデバイスの管理

タスク	手順
ストレージデバイスとノードに TrueCopy ソフトウェアをインストールする	Hitachi ストレージデバイスに付属するドキュメントを参照してください。
Hitachi 複製グループを構成する	98 ページの「Hitachi TrueCopy 複製グループを構成する方法」
DID デバイスを構成する	100 ページの「Hitachi TrueCopy を使用して DID デバイスを複製用に構成する方法」
複製されたグループを登録する	129 ページの「デバイスグループを追加および登録する方法 (Solaris Volume Manager)」
構成を確認する	102 ページの「HITACHI TrueCopy で複製されたグローバルデバイスグループ構成を確認する方法」

▼ Hitachi TrueCopy 複製グループを構成する方法

始める前に

まず、プライマリクラスタの共有ディスクに Hitachi TrueCopy デバイスグループを構成します。この構成情報は、HITACHI アレイにアクセスするクラスタの各ノードで、`/etc/horcm.conf` ファイルに指定します。`/etc/horcm.conf` ファイルを構成する方法の詳細については、Sun StorEdge SE 9900 V シリーズのコマンドおよび制御インタフェースのユーザーおよびリファレンスガイドを参照してください。



注意 - 作成する Oracle Solaris Cluster デバイスグループ (Solaris Volume Manager、ZFS、または raw ディスク) の名前は、複製されたデバイスグループと同じ名前にする必要があります。

- 1 ストレージアレイに接続されたすべてのノードで、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.modify` を提供する役割になります。
- 2 `/etc/services` ファイルに `horcm` エントリを追加します。

```
horcm 9970/udp
```

 新しいエントリのポート番号とプロトコル名を指定します。
- 3 `/etc/horcm.conf` ファイルに Hitachi TrueCopy デバイスグループの構成情報を指定します。
 手順については、TrueCopy ソフトウェアに付属するドキュメントを参照してください。
- 4 すべてのノードで `horcmstart.sh` コマンドを実行し、TrueCopy CCI デーモンを起動します。

```
# /usr/bin/horcmstart.sh
```
- 5 まだ複製ペアを作成していない場合は、この時点で作成します。
`paircreate` コマンドを使用して、必要な防御レベルで複製ペアを作成します。複製ペアの作成手順については、TrueCopy のドキュメントを参照してください。
- 6 複製されたデバイスによって構成された各ノードで、`pairedisplay` コマンドを使用して、データの複製が正しく設定されていることを確認します。`fence_level` が `ASYNC` の Hitachi TrueCopy または Hitachi Universal Replicator デバイスグループは、`ctgid` をシステム上のほかのデバイスグループと共有できません。

```
# pairedisplay -g group-name
Group PairVol(L/R) (Port#,TID,LU),Seq#,LDEV#,P/S,Status,Fence,Seq#,P-LDEV# M
group-name pair1(L) (CL1-C , 0, 9) 54321 58..P-VOL PAIR NEVER ,12345 29 -
group-name pair1(R) (CL1-A , 0, 29)12345 29..S-VOL PAIR NEVER ,----- 58 -
```
- 7 すべてのノードが複製グループをマスターできることを確認します。
 - a. どのノードにプライマリ複製が含まれ、どのノードにセカンダリ複製が含まれているかを判別するには、`pairedisplay` コマンドを使用します。

```
# pairedisplay -g group-name
Group PairVol(L/R) (Port#,TID,LU),Seq#,LDEV#,P/S,Status,Fence,Seq#,P-LDEV# M
group-name pair1(L) (CL1-C , 0, 9) 54321 58..P-VOL PAIR NEVER ,12345 29 -
group-name pair1(R) (CL1-A , 0, 29)12345 29..S-VOL PAIR NEVER ,----- 58 -
```

状態が P-VOL のローカル (L) デバイスがあるノードにはプライマリ複製が含まれ、状態が S-VOL のローカル (L) デバイスがあるノードにはセカンダリ複製が含まれます。

- b. セカンダリ複製を含むノードで **horctakeover** コマンドを実行し、セカンダリノードをマスターにします。

```
# horctakeover -g group-name
```

最初のデータコピーが完了するまで待ってから、次の手順に進みます。

- c. **horctakeover** を実行したノードに、状態が **P-VOL** のローカル (L) デバイスがあることを確認します。

```
# pairedisplay -g group-name
Group PairVol(L/R) (Port#,TID,LU),Seq#,LDEV#,P/S,Status,Fence,Seq#,P-LDEV# M
group-name pair1(L) (CL1-C , 0, 9) 54321 58..S-VOL PAIR NEVER ,12345 29 -
group-name pair1(R) (CL1-A , 0, 29)12345 29..P-VOL PAIR NEVER ,----- 58 -
```

- d. 最初にプライマリ複製があったノードで **horctakeover** コマンドを実行します。

```
# horctakeover -g group-name
```

- e. **pairedisplay** コマンドを実行して、プライマリノードが元の構成に戻ったことを確認します。

```
# pairedisplay -g group-name
Group PairVol(L/R) (Port#,TID,LU),Seq#,LDEV#,P/S,Status,Fence,Seq#,P-LDEV# M
group-name pair1(L) (CL1-C , 0, 9) 54321 58..P-VOL PAIR NEVER ,12345 29 -
group-name pair1(R) (CL1-A , 0, 29)12345 29..S-VOL PAIR NEVER ,----- 58 -
```

次の手順 [100 ページの「Hitachi TrueCopy を使用して DID デバイスを複製用に構成する方法」](#)の手順に従って、複製されたデバイスの構成を続行します。

▼ Hitachi TrueCopy を使用して DID デバイスを複製用に構成する方法

始める前に 複製されるデバイス用にデバイスグループを構成したら、複製されるデバイスが使用するデバイス識別子 (DID) ドライバを構成する必要があります。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 **solaris.cluster.modify** を提供する役割になります。

- 2 すべてのノードで **horcmd** デーモンが実行中であることを確認します。

実行中でなければ、次のコマンドでデーモンが起動されます。デーモンがすでに実行中の場合は、システムによりメッセージが表示されます。

```
# /usr/bin/horcmstart.sh
```

- 3 **pairedisplay** コマンドを使用して、どのノードにセカンダリ複製が含まれているかを判別します。

```
# pairedisplay -g group-name
Group PairVol(L/R) (Port#,TID,LU),Seq#,LDEV#,P/S,Status,Fence,Seq#,P-LDEV# M
group-name pair1(L) (CL1-C , 0, 9) 54321 58..P-VOL PAIR NEVER ,12345 29 -
group-name pair1(R) (CL1-A , 0, 29)12345 29..S-VOL PAIR NEVER ,----- 58 -
```

状態が S-VOL のローカル (L) デバイスがあるノードにはセカンダリ複製が含まれません。

- 4 前の手順で判別されたセカンダリ複製があるノードで、ストレージベースの複製に使用する DID デバイスを構成します。

このコマンドは、デバイス複製ペアの2つの独立した DID インスタンスを組み合わせ、1つの論理的 DID インスタンスにします。1つのインスタンスであることで、どちらの側からもボリューム管理ソフトウェアでデバイスを使用できます。



注意-セカンダリ複製に複数のノードが接続されている場合、これらのノードの1つでのみ、次のコマンドを実行します。

```
# cldevice replicate -D primary-replica-nodename -S secondary replica-nodename
```

primary-replica-nodename

プライマリ複製を含むリモートノードの名前を指定します。

-S

現在のノード以外のソースノードを指定します。

secondary replica-nodename

セカンダリ複製を含むリモートノードの名前を指定します。

注-デフォルトでは、現在のノードがソースノードです。-s オプションは、別のソースノードを指定するのに使用します。

- 5 DID インスタンスが結合されていることを確認します。

```
# cldevice list -v logical_DID_device
```

- 6 **TrueCopy** 複製が設定されていることを確認します。

```
# cldevice show logical_DID_device
```

コマンド出力に、複製タイプは TrueCopy であることが示されます。

- 7 DID の再マッピングによって、複製されたすべてのデバイスが正しく結合されなかった場合は、複製された個々のデバイスを手動で結合します。



注意-DID インスタンスを手動で結合する場合は十分に注意してください。デバイスの再マッピングが不適切だと、データが破損する可能性があります。

- a. セカンダリ複製を含むすべてのノードで、**cldevice combine** コマンドを実行します。

```
# cldevice combine -d destination-instance source-instance
```

-d プライマリ複製に対応するリモート DID インスタンス。
destination-instance

source-instance セカンダリ複製に対応するローカル DID インスタンス。

- b. DID の再マッピングが正しく行われたことを確認します。

```
# cldevice list desination-instance source-instance
```

DID インスタンスの 1 つが表示されないはずです。

- 8 すべてのノード上で、結合されたすべての DID インスタンスの DID デバイスがアクセス可能であることを確認します。

```
# cldevice list -v
```

次の手順 複製されるデバイスグループの構成を完了するために、次の手順を実行します。

- 129 ページの「デバイスグループを追加および登録する方法 (Solaris Volume Manager)」
デバイスグループを登録するときには、TrueCopy 複製グループと同じ名前を指定するようにします。
- 102 ページの「HITACHI TrueCopy で複製されたグローバルデバイスグループ構成を確認する方法」

▼ HITACHI TrueCopy で複製されたグローバルデバイスグループ構成を確認する方法

始める前に グローバルデバイスグループを確認する前に、まずそれを作成する必要があります。Solaris Volume Manager、ZFS、または raw ディスクからデバイスグループを使用できます。詳細は、次を参照してください。

- 129 ページの「デバイスグループを追加および登録する方法 (Solaris Volume Manager)」
- 132 ページの「デバイスグループ (raw ディスク) を追加および登録する方法」
- 132 ページの「複製デバイスグループ (ZFS) の追加と登録方法」



注意 - 作成した Oracle Solaris Cluster デバイスグループ (Solaris Volume Manager または raw ディスク) の名前は、複製されたデバイスグループの名前と同じである必要があります。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 プライマリデバイスグループが、プライマリ複製が含まれるノードと同じノードに対応することを確認します。

```
# paireddisplay -g group-name
# cldevicegroup status -n nodename group-name
```

- 2 デバイスグループに対して複製プロパティが設定されていることを確認します。

```
# cldevicegroup show -n nodename group-name
```

- 3 デバイスに対して複製プロパティが設定されていることを確認します。

```
# usr/cluster/bin/cldevice status [-s state] [-n node[,?]] [+| [disk-device ]]
```

- 4 試験的にスイッチオーバーを実行し、デバイスグループが正しく構成され、複製がノード間を移動できることを確認します。

デバイスグループがオフラインの場合は、オンラインにします。

```
# cldevicegroup switch -n nodename group-name
```

-n nodename デバイスグループの切り替え先のノード。このノードが新しいプライマリになります。

- 5 次のコマンドの出力を比較し、スイッチオーバーが成功したことを確認します。

```
# paireddisplay -g group-name
# cldevicegroup status -n nodename group-name
```

例: Oracle Solaris Cluster の TrueCopy 複製グループの構成

この例では、クラスタの TrueCopy 複製を設定するのに必要な Oracle Solaris Cluster 固有の手順を完了します。この例では、次のタスクが実行済みであることを前提とします。

- Hitachi の LUN を設定済み
- ストレージデバイスとクラスタノードに TrueCopy ソフトウェアがインストール済み

■ クラスタノードで複製ペアを構成済み

複製ペアを構成する手順については、[98 ページの「Hitachi TrueCopy 複製グループを構成する方法」](#)を参照してください。

この例には、TrueCopy を使用する 3 ノードのクラスタが含まれます。クラスタは 2 つのリモートサイトにまたがっており、一方のサイトに 2 つのノードがあり、他方のサイトには 1 つのノードがあります。それぞれのサイトに独自の Hitachi ストレージデバイスがあります。

次の例は、各ノードの TrueCopy /etc/horcm.conf 構成ファイルを示しています。

例 5-1 ノード 1 の TrueCopy 構成ファイル

```
HORCM_DEV
#dev_group    dev_name    port#    TargetID    LU#    MU#
VG01          pair1      CL1-A    0           29
VG01          pair2      CL1-A    0           30
VG01          pair3      CL1-A    0           31
HORCM_INST
#dev_group    ip_address  service
VG01          node-3      horcm
```

例 5-2 ノード 2 の TrueCopy 構成ファイル

```
HORCM_DEV
#dev_group    dev_name    port#    TargetID    LU#    MU#
VG01          pair1      CL1-A    0           29
VG01          pair2      CL1-A    0           30
VG01          pair3      CL1-A    0           31
HORCM_INST
#dev_group    ip_address  service
VG01          node-3      horcm
```

例 5-3 ノード 3 の TrueCopy 構成ファイル

```
HORCM_DEV
#dev_group    dev_name    port#    TargetID    LU#    MU#
VG01          pair1      CL1-C    0           09
VG01          pair2      CL1-C    0           10
VG01          pair3      CL1-C    0           11
HORCM_INST
#dev_group    ip_address  service
VG01          node-1      horcm
VG01          node-2      horcm
```

上記の例では、2 つのサイト間で 3 つの LUN が複製されます。LUN はすべて VG01 という名前の複製グループに属しています。pairdisplay コマンドにより、この情報が確認され、ノード 3 にプライマリ複製があることが示されます。

例5-4 ノード1でのpairdisplayコマンドの出力

```
# pairdisplay -g VG01
Group PairVol(L/R) (Port#,TID,LU),Seq#,LDEV#.P/S,Status,Fence, Seq#,P-LDEV# M
VG01 pair1(L) (CL1-A , 0, 29)61114 29..S-VOL PAIR NEVER ,----- 58 -
VG01 pair1(R) (CL1-C , 0, 9)20064 58..P-VOL PAIR NEVER ,61114 29 -
VG01 pair2(L) (CL1-A , 0, 30)61114 30..S-VOL PAIR NEVER ,----- 59 -
VG01 pair2(R) (CL1-C , 0, 10)20064 59..P-VOL PAIR NEVER ,61114 30 -
VG01 pair3(L) (CL1-A , 0, 31)61114 31..S-VOL PAIR NEVER ,----- 60 -
VG01 pair3(R) (CL1-C , 0, 11)20064 60..P-VOL PAIR NEVER ,61114 31 -
```

例5-5 ノード2でのpairdisplayコマンドの出力

```
# pairdisplay -g VG01
Group PairVol(L/R) (Port#,TID,LU),Seq#,LDEV#.P/S,Status,Fence, Seq#,P-LDEV# M
VG01 pair1(L) (CL1-A , 0, 29)61114 29..S-VOL PAIR NEVER ,----- 58 -
VG01 pair1(R) (CL1-C , 0, 9)20064 58..P-VOL PAIR NEVER ,61114 29 -
VG01 pair2(L) (CL1-A , 0, 30)61114 30..S-VOL PAIR NEVER ,----- 59 -
VG01 pair2(R) (CL1-C , 0, 10)20064 59..P-VOL PAIR NEVER ,61114 30 -
VG01 pair3(L) (CL1-A , 0, 31)61114 31..S-VOL PAIR NEVER ,----- 60 -
VG01 pair3(R) (CL1-C , 0, 11)20064 60..P-VOL PAIR NEVER ,61114 31 -
```

例5-6 ノード3でのpairdisplayコマンドの出力

```
# pairdisplay -g VG01
Group PairVol(L/R) (Port#,TID,LU),Seq#,LDEV#.P/S,Status,Fence, Seq#,P-LDEV# M
VG01 pair1(L) (CL1-C , 0, 9)20064 58..P-VOL PAIR NEVER ,61114 29 -
VG01 pair1(R) (CL1-A , 0, 29)61114 29..S-VOL PAIR NEVER ,----- 58 -
VG01 pair2(L) (CL1-C , 0, 10)20064 59..P-VOL PAIR NEVER ,61114 30 -
VG01 pair2(R) (CL1-A , 0, 30)61114 30..S-VOL PAIR NEVER ,----- 59 -
VG01 pair3(L) (CL1-C , 0, 11)20064 60..P-VOL PAIR NEVER ,61114 31 -
VG01 pair3(R) (CL1-A , 0, 31)61114 31..S-VOL PAIR NEVER ,----- 60 -
```

使用中のディスクを確認するには、次の例に示すように、pairdisplayコマンドの-fdオプションを使用します。

例5-7 ノード1でのpairdisplayコマンドの出力:使用されているディスクの表示

```
# pairdisplay -fd -g VG01
Group PairVol(L/R) Device File ,Seq#,LDEV#.P/S,Status,Fence,Seq#,P-LDEV# M
VG01 pair1(L) c6t500060E800000000000000E0EBA0000001Dd0s2 61114 29..S-VOL PAIR NEVER ,----- 58 -
VG01 pair1(R) c5t500060E8000000000000004E600000003Ad0s2 20064 58..P-VOL PAIR NEVER ,61114 29 -
VG01 pair2(L) c6t500060E800000000000000E0EBA0000001Ed0s2 61114 30..S-VOL PAIR NEVER ,----- 59 -
VG01 pair2(R) c5t500060E8000000000000004E600000003Bd0s2 0064 59..P-VOL PAIR NEVER ,61114 30 -
VG01 pair3(L) c6t500060E800000000000000E0EBA0000001Fd0s2 61114 31..S-VOL PAIR NEVER ,----- 60 -
VG01 pair3(R) c5t500060E8000000000000004E600000003Cd0s2 20064 60..P-VOL PAIR NEVER ,61114 31 -
```

例5-8 ノード2でのpairdisplayコマンドの出力:使用されているディスクの表示

```
# pairdisplay -fd -g VG01
Group PairVol(L/R) Device File ,Seq#,LDEV#.P/S,Status,Fence,Seq#,P-LDEV# M
VG01 pair1(L) c5t500060E800000000000000E0EBA0000001Dd0s2 61114 29..S-VOL PAIR NEVER ,----- 58 -
VG01 pair1(R) c5t500060E8000000000000004E600000003Ad0s2 20064 58..P-VOL PAIR NEVER ,61114 29 -
VG01 pair2(L) c5t500060E800000000000000E0EBA0000001Ed0s2 61114 30..S-VOL PAIR NEVER ,----- 59 -
VG01 pair2(R) c5t500060E8000000000000004E600000003Bd0s2 20064 59..P-VOL PAIR NEVER ,61114 30 -
```

例 5-8 ノード 2 での `pairdisplay` コマンドの出力: 使用されているディスクの表示 (続き)

VG01 pair3(L)	c5t50060E80000000000000EEBA0000001Fd0s2	61114	31..S-VOL PAIR NEVER	,-----	60	-
VG01 pair3(R)	c5t50060E8000000000000004E600000003Cd0s2	20064	60..P-VOL PAIR NEVER	,61114	31	-

例 5-9 ノード 3 での `pairdisplay` コマンドの出力: 使用されているディスクの表示

```
# pairdisplay -fd -g VG01
Group PairVol(L/R) Device_File                               ,Seq#,LDE# ,P/S,Status,Fence ,Seq#,P-LDE# ,M
VG01 pair1(L) c5t50060E80000000000000004E600000003Ad0s2 20064 58..S-VOL PAIR NEVER ,61114 29 -
VG01 pair1(R) c6t500060E8000000000000000EEBA0000001Dd0s2 61114 29..S-VOL PAIR NEVER ,---- 58 -
VG01 pair2(L) c5t50060E80000000000000004E600000003Bd0s2 20064 59..P-VOL PAIR NEVER ,61114 30 -
VG01 pair2(R) c6t500060E8000000000000000EEBA0000001Ed0s2 61114 30..S-VOL PAIR NEVER ,---- 59 -
VG01 pair3(L) c5t50060E80000000000000004E600000003Cd0s2 20064 60..P-VOL PAIR NEVER ,61114 31 -
VG01 pair3(R) c6t500060E8000000000000000EEBA0000001Fd0s2 61114 31..S-VOL PAIR NEVER ,---- 60 -
```

これらの例は、次のディスクが使用中であることを示しています。

- ノード 1 上:
 - c6t500060E80000000000000000EEBA0000001Dd0s2
 - c6t500060E80000000000000000EEBA0000001Ed0s2
 - c6t500060E80000000000000000EEBA0000001Fd0s
- ノード 2 上:
 - c5t500060E80000000000000000EEBA0000001Dd0s2
 - c5t500060E80000000000000000EEBA0000001Ed0s2
 - c5t500060E80000000000000000EEBA0000001Fd0s2
- ノード 3 上:
 - c5t50060E800000000000000004E600000003Ad0s2
 - c5t50060E800000000000000004E600000003Bd0s2
 - c5t50060E800000000000000004E600000003Cd0s2

これらのディスクに対応する DID デバイスを表示するには、次の例に示すように、`cldevice list` コマンドを使用します。

例 5-10 使用されているディスクに対応する DID の表示

```
# cldevice list -v
```

DID	Device	Full Device Path
1		node-1:/dev/rdsck/c0t0d0 /dev/did/rdsck/d1
2		node-1:/dev/rdsck/c0t6d0 /dev/did/rdsck/d2
11		node-1:/dev/rdsck/c6t500060E8000000000000EBA00000020d0 /dev/did/rdsck/d11
11		node-2:/dev/rdsck/c5t500060E8000000000000EBA00000020d0 /dev/did/rdsck/d11
12		node-1:/dev/rdsck/c6t500060E8000000000000EBA0000001Fd0 /dev/did/rdsck/d12
12		node-2:/dev/rdsck/c5t500060E8000000000000EBA0000001Fd0 /dev/did/rdsck/d12
13		node-1:/dev/rdsck/c6t500060E8000000000000EBA0000001Ed0 /dev/did/rdsck/d13
13		node-2:/dev/rdsck/c5t500060E8000000000000EBA0000001Ed0 /dev/did/rdsck/d13
14		node-1:/dev/rdsck/c6t500060E8000000000000EBA0000001Dd0 /dev/did/rdsck/d14

例 5-10 使用されているディスクに対応する DID の表示 (続き)

```

14      node-2:/dev/rdisk/c5t500060E800000000000000E600000001Dd0 /dev/did/rdsk/d14
18      node-3:/dev/rdisk/c0t0d0 /dev/did/rdsk/d18
19      node-3:/dev/rdisk/c0t6d0 /dev/did/rdsk/d19
20      node-3:/dev/rdisk/c5t50060E8000000000000004E6000000013d0 /dev/did/rdsk/d20
21      node-3:/dev/rdisk/c5t50060E8000000000000004E600000003Dd0 /dev/did/rdsk/d21
22      node-3:/dev/rdisk/c5t50060E8000000000000004E600000003Cd0 /dev/did/rdsk/d2223
23      node-3:/dev/rdisk/c5t50060E8000000000000004E600000003Bd0 /dev/did/rdsk/d23
24      node-3:/dev/rdisk/c5t50060E8000000000000004E600000003Ad0 /dev/did/rdsk/d24

```

複製されたデバイスの各ペアの DID インスタンスを結合する場合は、`cldevice list` により、DID インスタンス 12 と 22、インスタンス 13 と 23、インスタンス 14 と 24 を結合します。ノード 3 にはプライマリ複製があるため、ノード 1 またはノード 2 のいずれかから `cldevice -T` コマンドを実行します。必ず、セカンダリ複製があるノードのインスタンスを結合します。このコマンドは、両方のノードではなく、1 つのノードからのみ実行します。

次の例は、ノード 1 でコマンドを実行して DID インスタンスを結合した場合の出力を示しています。

例 5-11 DID インスタンスの結合

```

# cldevice replicate -D node-3
Remapping instances for devices replicated with node-3...
VG01 pair1 L node-1:/dev/rdisk/c6t500060E800000000000000E600000001Dd0
VG01 pair1 R node-3:/dev/rdisk/c5t50060E8000000000000004E600000003Ad0
Combining instance 14 with 24
VG01 pair2 L node-1:/dev/rdisk/c6t500060E800000000000000E600000001Ed0
VG01 pair2 R node-3:/dev/rdisk/c5t50060E8000000000000004E600000003Bd0
Combining instance 13 with 23
VG01 pair3 L node-1:/dev/rdisk/c6t500060E800000000000000E600000001Fd0
VG01 pair3 R node-3:/dev/rdisk/c5t50060E8000000000000004E600000003Cd0
Combining instance 12 with 22

```

`cldevice list` の出力を確認すると、両方のサイトの LUN に同じ DID インスタンスがあります。次の例に示すように、同じ DID インスタンスがあるそれぞれの複製ペアは、単一の DID デバイスのように見えます。

例 5-12 結合された DID の表示

```

# cldevice list -v
DID Device      Full Device Path
-----
1      node-1:/dev/rdisk/c0t0d0 /dev/did/rdsk/d1
2      node-1:/dev/rdisk/c0t6d0 /dev/did/rdsk/d2
11     node-1:/dev/rdisk/c6t500060E800000000000000E6000000020d0 /dev/did/rdsk/d11
11     node-2:/dev/rdisk/c5t50060E8000000000000000E6000000020d0 /dev/did/rdsk/d11
18     node-3:/dev/rdisk/c0t0d0 /dev/did/rdsk/d18
19     node-3:/dev/rdisk/c0t6d0 /dev/did/rdsk/d19
20     node-3:/dev/rdisk/c5t50060E8000000000000004E6000000013d0 /dev/did/rdsk/d20

```

例 5-12 結合された DID の表示 (続き)

```

21 node-3:/dev/rdisk/c5t50060E8000000000000004E600000003Dd0 /dev/did/rdsk/d21
22 node-1:/dev/rdisk/c6t50060E800000000000000EEBA0000001Fd0 /dev/did/rdsk/d1222
22 node-2:/dev/rdisk/c5t50060E800000000000000EEBA0000001Fd0 /dev/did/rdsk/d12
22 node-3:/dev/rdisk/c5t50060E8000000000000004E600000003Cd0 /dev/did/rdsk/d22
23 node-1:/dev/rdisk/c6t50060E800000000000000EEBA0000001Ed0 /dev/did/rdsk/d13
23 node-2:/dev/rdisk/c5t50060E800000000000000EEBA0000001Ed0 /dev/did/rdsk/d13
23 node-3:/dev/rdisk/c5t50060E8000000000000004E600000003Bd0 /dev/did/rdsk/d23
24 node-1:/dev/rdisk/c6t50060E800000000000000EEBA0000001Dd0 /dev/did/rdsk/d24
24 node-2:/dev/rdisk/c5t50060E800000000000000EEBA0000001Dd0 /dev/did/rdsk/d24
24 node-3:/dev/rdisk/c5t50060E8000000000000004E600000003Ad0 /dev/did/rdsk/d24

```

次の手順では、ボリュームマネージャーデバイスグループを作成します。このコマンドは、プライマリ複製があるノードから発行します。この例ではノード 3 です。次の例に示すように、デバイスグループに複製グループと同じ名前を付けます。

例 5-13 Solaris ボリュームマネージャーデバイスグループの作成

```

# metaset -s VG01 -ah phys-deneb-3
# metaset -s VG01 -ah phys-deneb-1
# metaset -s VG01 -ah phys-deneb-2
# metaset -s VG01 -a /dev/did/rdsk/d22
# metaset -s VG01 -a /dev/did/rdsk/d23
# metaset -s VG01 -a /dev/did/rdsk/d24
# metaset
Set name = VG01, Set number = 1

```

Host	Owner
phys-deneb-3	Yes
phys-deneb-1	
phys-deneb-2	

Drive	Dbase
d22	Yes
d23	Yes
d24	Yes

この時点でデバイスグループは使用可能なので、メタデバイスを作成でき、3つのノードのうち、任意のノードにデバイスグループを移動できます。ただし、スイッチオーバーとフェイルオーバーの効率を高めるため、`cldevicegroup set` を実行し、クラスタ構成内でそのデバイスグループを複製済みとマークします。

例 5-14 スイッチオーバーとフェイルオーバーの効率向上

```

# cldevicegroup sync VG01
# cldevicegroup show VG01
=== Device Groups===

```

Device Group Name	VG01
Type:	SVM

例 5-14 スイッチオーバーとフェイルオーバーの効率向上 (続き)

```
failback: no
Node List: phys-deneb-3, phys-deneb-1, phys-deneb-2
preferenced: yes
numsecondaries: 1
device names: VG01
Replication type: truecopy
```

複製グループの構成はこの手順で完了です。構成に成功したことを確認するには、[102 ページの「HITACHI TrueCopy で複製されたグローバルデバイスグループ構成を確認する方法」](#) の手順を実行します。

EMC Symmetrix Remote Data Facility で複製されたデバイスの管理

次の表に、EMC Symmetrix Remote Data Facility (SRDF) ストレージベースの複製されたデバイスを設定および管理するために実行する必要のあるタスクを示します。

表 5-3 タスクマップ: EMC SRDF ストレージベースの複製されたデバイスの管理

タスク	手順
ストレージデバイスとノードに SRDF ソフトウェアをインストールする	EMC ストレージデバイスに付属するドキュメント。
EMC 複製グループを構成する	109 ページの「EMC SRDF 複製グループを構成する方法」
DID デバイスを構成する	112 ページの「EMC SRDF を使用して DID デバイスを複製用に構成する方法」
複製されたグループを登録する	129 ページの「デバイスグループを追加および登録する方法 (Solaris Volume Manager)」
構成を確認する	113 ページの「EMC SRDF で複製されたグローバルデバイスグループ構成を確認する方法」
キャンパスクラスタのプライマリルームが完全に失敗したあとに手動でデータを回復する	119 ページの「プライマリルームの完全な失敗後に EMC SRDF データを回復する方法」

▼ EMC SRDF 複製グループを構成する方法

始める前に EMC Symmetrix Remote Data Facility (SRDF) 複製グループを構成する前に、すべてのクラスタノードに EMC Solutions Enabler ソフトウェアをインストールする必要があります。

す。まず、クラスタの共有ディスクに EMC SRDF デバイスグループを構成します。EMC SRDF デバイスグループを構成する方法の詳細は、EMC SRDF 製品のドキュメントを参照してください。

EMC SRDF を使用する場合は、静的デバイスではなく、動的デバイスを使用します。静的デバイスでは複製プライマリを変更するのに数分かかり、フェイルオーバー時間に影響を与えることがあります。



注意 - 作成する Oracle Solaris Cluster デバイスグループ (Solaris Volume Manager、または raw ディスク) の名前は、複製されたデバイスグループと同じ名前にしてください。

- 1 ストレージアレイに接続されたすべてのノードで、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.modify` を提供する役割になります。
- 2 並列 SRDF または直列接続されたデバイスを使用し、サイトが 3 つまたはデータセンターが 3 つある実装の場合、`SYMAPI_2SITE_CLUSTER_DG` パラメータを設定します。参加しているすべてのクラスタノードで、Solutions Enabler のオプションファイルに次のエントリを追加します。

```
SYMAPI_2SITE_CLUSTER_DG=:rdf-group-number
```

`device-group` デバイスグループの名前を指定します。

`rdf-group-number` ホストのローカル Symmetrix を 2 番目のサイトの Symmetrix に接続する RDF グループを指定します。

このエントリにより、クラスタソフトウェアで、2 つの SRDF 同期サイト間でのアプリケーションの移動を自動化できます。

3 つのデータセンター構成の詳細については、『[Oracle Solaris Cluster Geographic Edition Overview](#)』の「[Three-Data-Center \(3DC\) Topologies](#)」を参照してください。

- 3 複製データで構成された各ノードで、シンメトリックスデバイス構成を検出します。

これには数分かかることがあります。

```
# /usr/symcli/bin/symcfg discover
```

- 4 まだ複製ペアを作成していない場合は、この時点で作成します。
複製ペアを作成するには、`symrdf` コマンドを使用します。複製ペアの作成方法の手順については、SRDF のドキュメントを参照してください。

注- サイトが3つまたはデータセンターが3つある実装で並列 RDF デバイスを使用する場合は、すべての `symrdf` コマンドに次のパラメータを追加します。

```
-rdfg rdf-group-number
```

`symrdf` コマンドに RDF グループ番号を指定することで、`symrdf` の操作が正しい RDF グループに向けられるようにします。

- 5 複製されたデバイスによって構成された各ノードで、データの複製が正しく設定されていることを確認します。

```
# /usr/symcli/bin/symdg show group-name
```

- 6 デバイスグループのスワップを実行します。

- a. プライマリ複製とセカンダリ複製が同期していることを確認します。

```
# /usr/symcli/bin/symrdf -g group-name verify -synchronized
```

- b. どのノードにプライマリ複製が含まれ、どのノードにセカンダリ複製が含まれているかを判別するには、`symdg show` コマンドを使用します。

```
# /usr/symcli/bin/symdg show group-name
```

RDF1 デバイスのノードにはプライマリ複製が含まれ、RDF2 デバイス状態のノードにはセカンダリ複製が含まれます。

- c. セカンダリ複製を有効にします。

```
# /usr/symcli/bin/symrdf -g group-name failover
```

- d. RDF1 デバイスと RDF2 デバイスをスワップします。

```
# /usr/symcli/bin/symrdf -g group-name swap -refresh R1
```

- e. 複製ペアを有効にします。

```
# /usr/symcli/bin/symrdf -g group-name establish
```

- f. プライマリノードとセカンダリ複製が同期していることを確認します。

```
# /usr/symcli/bin/symrdf -g group-name verify -synchronized
```

- 7 最初にプライマリ複製があったノードで、手順5のすべてを繰り返します。

次の手順 EMC SRDF で複製されたデバイス用にデバイスグループを構成したら、複製されたデバイスが使用するデバイス識別子 (DID) ドライバを構成する必要があります。

▼ EMC SRDF を使用して DID デバイスを複製用に構成する方法

この手順では、複製されたデバイスが使用するデバイス識別名 (DID) ドライバを構成します。

始める前に `phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.modify` を提供する役割になります。
- 2 構成される RDF1 デバイスおよび RDF2 デバイスに対応する DID デバイスを判別します。

```
# /usr/symcli/bin/symdg show group-name
```

注 - システムに Oracle Solaris デバイスのパッチ全体が表示されない場合は、環境変数 `SYMCLI_FULL_PDEVNAME` を 1 に設定して、`symdg -show` コマンドを再入力します。

- 3 Oracle Solaris デバイスに対応する DID デバイスを判別します。
- 4 一致した DID デバイスのペアごとに、複製された 1 つの DID デバイスにインスタンスを結合します。RDF2/セカンダリ側から次のコマンドを実行します。

```
# cldevice combine -t srdf -g replication-device-group \
  -d destination-instance source-instance
```

注 - SRDF データ複製デバイスでは、`-t` オプションはサポートされていません。

<code>-t replication-type</code>	複製タイプを指定します。EMC SRDF の場合、 SRDF を入力します。
<code>-g replication-device-group</code>	<code>symdg show</code> コマンドで表示されるデバイスグループの名前を指定します。
<code>-d destination-instance</code>	RDF1 デバイスに対応する DID インスタンスを指定します。
<code>source-instance</code>	RDF2 デバイスに対応する DID インスタンスを指定します。

注-間違ったDID デバイスを結合した場合は、`scdidadm` コマンドで `-b` オプションを使用して、2つのDID デバイスの結合を取り消します。

```
# scdidadm -b device
```

`-b device` インスタンスを結合したときに `destination_device` に対応していたDID インスタンス。

- 複製デバイスグループの名前が変更された場合は、**Hitachi TrueCopy** や **SRDF** で追加の手順が必要です。手順1-4が完了したら、該当する追加手順を実行します。

項目	説明
TrueCopy	複製デバイスグループ (および対応するグローバルデバイスグループ) の名前が変更された場合、 <code>cldevice replicate</code> コマンドを再実行し、複製されたデバイスの情報を更新する必要があります。
SRDF	複製デバイスグループと、対応するグローバルデバイスグループの名前が変更された場合は、最初に、 <code>scdidadm -b</code> コマンドを使用して既存の情報を削除し、複製されたデバイス情報を更新する必要があります。最後の手順では、 <code>cldevice combine</code> コマンドを使用して、更新された新しいデバイスを作成します。

- DID インスタンスが結合されていることを確認します。

```
# cldevice list -v device
```

- SRDF 複製が設定されていることを確認します。

```
# cldevice show device
```

- すべてのノード上で、結合されたすべてのDID インスタンスのDID デバイスがアクセス可能であることを確認します。

```
# cldevice list -v
```

次の手順 複製されたデバイスが使用するデバイス識別名 (DID) ドライバを構成したら、EMC SRDF で複製されたグローバルデバイスグループ構成を確認する必要があります。

▼ EMC SRDF で複製されたグローバルデバイスグループ構成を確認する方法

始める前に グローバルデバイスグループを確認する前に、まずそれを作成する必要があります。Solaris Volume Manager、ZFS、またはraw ディスクからデバイスグループを使用できます。詳細は、次を参照してください。

- 129 ページの「デバイスグループを追加および登録する方法 (Solaris Volume Manager)」
- 132 ページの「デバイスグループ (raw ディスク) を追加および登録する方法」

■ 132 ページの「複製デバイスグループ(ZFS)の追加と登録方法」



注意 - 作成する Oracle Solaris Cluster デバイスグループ (Solaris Volume Manager または raw ディスク) の名前は、複製されたデバイスグループと同じ名前にする必要があります。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 プライマリデバイスグループが、プライマリ複製が含まれるノードと同じノードに対応することを確認します。

```
# symdbg -show group-name
# cldevicegroup status -n nodename group-name
```

- 2 試験的にスイッチオーバーを実行し、デバイスグループが正しく構成され、複製がノード間を移動できることを確認します。

デバイスグループがオフラインの場合は、オンラインにします。

```
# cldevicegroup switch -n nodename group-name
-n nodename    デバイスグループの切り替え先のノード。このノードが新しいプライマリになります。
```

- 3 次のコマンドの出力を比較し、スイッチオーバーが成功したことを確認します。

```
# symdbg -show group-name
# cldevicegroup status -n nodename group-name
```

例: Oracle Solaris Cluster の SRDF 複製グループの構成

この例では、クラスタの SRDF 複製を設定するのに必要な Oracle Solaris Cluster 固有の手順を完了します。この例では、次のタスクが実行済みであることを前提とします。

- アレイ間の複製の LUN のペア作成が完了している。
- ストレージデバイスとクラスタノードに SRDF ソフトウェアがインストール済みである。

この例には4ノードクラスタが含まれ、そのうちの2ノードは1つのシンメトリックスに接続され、残りの2ノードはもう1つのシンメトリックスに接続されています。SRDF デバイスグループは、dg1 と呼ばれます。

例5-15 複製ペアの作成

すべてのノードで次のコマンドを実行します。

```
# symcfg discover
! This operation might take up to a few minutes.
# symdev list pd
```

Symmetrix ID: 000187990182

Device Name		Directors			Device			
Sym	Physical	SA	:P	DA :IT	Config	Attribute	Sts	Cap (MB)
0067	c5t600604800001879901*	16D:0	02A:C1		RDF2+Mir	N/Grp'd	RW	4315
0068	c5t600604800001879901*	16D:0	16B:C0		RDF1+Mir	N/Grp'd	RW	4315
0069	c5t600604800001879901*	16D:0	01A:C0		RDF1+Mir	N/Grp'd	RW	4315
...								

RDF1 側のすべてのノードで、次のように入力します。

```
# symdg -type RDF1 create dg1
# symld -g dg1 add dev 0067
```

RDF2 側のすべてのノードで、次のように入力します。

```
# symdg -type RDF2 create dg1
# symld -g dg1 add dev 0067
```

例5-16 データ複製設定の確認

クラスタ内の1つのノードから、次のように入力します。

```
# symdg show dg1
```

Group Name: dg1

Group Type	: RDF1 (RDFA)
Device Group in GNS	: No
Valid	: Yes
Symmetrix ID	: 000187900023
Group Creation Time	: Thu Sep 13 13:21:15 2007
Vendor ID	: EMC Corp
Application ID	: SYMCLI
Number of STD Devices in Group	: 1
Number of Associated GK's	: 0
Number of Locally-associated BCV's	: 0
Number of Locally-associated VDEV's	: 0
Number of Remotely-associated BCV's (STD RDF):	0
Number of Remotely-associated BCV's (BCV RDF):	0
Number of Remotely-assoc'd RBCV's (RBCV RDF) :	0

例 5-16 データ複製設定の確認 (続き)

```
Standard (STD) Devices (1):
{
-----
LdevName          PdevName          Sym   Att. Sts   Cap
Dev              Dev              Dev
-----
DEV001            /dev/rdisk/c5t6006048000018790002353594D303637d0s2 0067    RW    4315
}

Device Group RDF Information
...
# symrdf -g dg1 establish

Execute an RDF 'Incremental Establish' operation for device
group 'dg1' (y/[n]) ? y

An RDF 'Incremental Establish' operation execution is
in progress for device group 'dg1'. Please wait...

Write Disable device(s) on RA at target (R2).....Done.
Suspend RDF link(s).....Done.
Mark target (R2) devices to refresh from source (R1).....Started.
Device: 0067 ..... Marked.
Mark target (R2) devices to refresh from source (R1).....Done.
Merge device track tables between source and target.....Started.
Device: 0067 ..... Merged.
Merge device track tables between source and target.....Done.
Resume RDF link(s).....Started.
Resume RDF link(s).....Done.

The RDF 'Incremental Establish' operation successfully initiated for
device group 'dg1'.

#
# symrdf -g dg1 query

Device Group (DG) Name      : dg1
DG's Type                   : RDF2
DG's Symmetrix ID           : 000187990182

Target (R2) View              Source (R1) View              MODES
-----
Standard      ST              LI      ST
Logical        A              N        A
Device Dev    T  R1 Inv  R2 Inv K  Dev    T  R1 Inv  R2 Inv  RDF Pair
              E  Tracks  Tracks S
-----
DEV001 0067 WD          0          0 RW 0067 RW          0          0 S.. Synchronized
Total
MB(s)      -----
          0.0          0.0          0.0          0.0
```

例 5-16 データ複製設定の確認 (続き)

```
Legend for MODES:

M(ode of Operation): A = Async, S = Sync, E = Semi-sync, C = Adaptive Copy
D(omino)             : X = Enabled, . = Disabled
A(daptive Copy)      : D = Disk Mode, W = WP Mode, . = ACp off

#

例 5-17 使用されているディスクに対応する DID の表示
RDF1 側と RDF2 側に同じ手順を適用します。

dymdg show dg コマンドの出力の PdevName フィールドの下に DID が表示されます。
RDF1 側で、次のように入力します。

# symdg show dg1
Group Name: dg1

Group Type : RDF1 (RDFA)
...
Standard (STD) Devices (1):
{
-----
LdevName      PdevName      Sym      Cap
Dev  Att. Sts  (MB)
-----
DEV001        /dev/rdsk/c5t6006048000018790002353594D303637d0s2 0067      RW      4315
}

Device Group RDF Information
...

対応する DID を取得するには、次のように入力します。

# scdidadm -L | grep c5t6006048000018790002353594D303637d0
217      pmoney1:/dev/rdsk/c5t6006048000018790002353594D303637d0 /dev/did/rdsk/d217
217      pmoney2:/dev/rdsk/c5t6006048000018790002353594D303637d0 /dev/did/rdsk/d217
#

対応する DID の一覧を表示するには、次のように入力します。

# cldevice show d217

=== DID Device Instances ===

DID Device Name: /dev/did/rdsk/d217
Full Device Path: pmoney2:/dev/rdsk/c5t6006048000018790002353594D303637d0
Full Device Path: pmoney1:/dev/rdsk/c5t6006048000018790002353594D303637d0
Replication: none
default_fencing: global
```

例 5-17 使用されているディスクに対応する DID の表示 (続き)

```
#

RDF2 側で、次のように入力します。

dymdg show dg コマンドの出力の PdevName フィールドの下に DID が表示されます。

# symdg show dg1

Group Name:  dg1

Group Type                                :  RDF2      (RDFA)
...
Standard (STD) Devices (1):
{
-----
LdevName          PdevName          Sym      Att.  Sts      Cap
Dev              (MB)
-----
DEV001            /dev/rdsk/c5t6006048000018799018253594D303637d0s2  0067      WD      4315
}

Device Group RDF Information
...
```

対応する DID を取得するには、次のように入力します。

```
# scdidadm -L | grep c5t6006048000018799018253594D303637d0
108      pmoney4:/dev/rdsk/c5t6006048000018799018253594D303637d0 /dev/did/rdsk/d108
108      pmoney3:/dev/rdsk/c5t6006048000018799018253594D303637d0 /dev/did/rdsk/d108
#
```

対応する DID の一覧を表示するには、次のように入力します。

```
# cldevice show d108

=== DID Device Instances ===

DID Device Name:      /dev/did/rdsk/d108
Full Device Path:     pmoney3:/dev/rdsk/c5t6006048000018799018253594D303637d0
Full Device Path:     pmoney4:/dev/rdsk/c5t6006048000018799018253594D303637d0
Replication:          none
default_fencing:      global

#
```

例 5-18 DID インスタンスの結合

RDF2 側から、次のように入力します。

```
# cldevice combine -t srdf -g dg1 -d d217 d108
#
```

例 5-19 結合された DID の表示

クラスタ内の任意のノードから、次のように入力します。

```
# cldevice show d217 d108
cldevice: (C727402) Could not locate instance "108".

=== DID Device Instances ===

DID Device Name:                               /dev/did/rdisk/d217
Full Device Path:                               pmoney1:/dev/rdisk/c5t6006048000018790002353594D303637d0
Full Device Path:                               pmoney2:/dev/rdisk/c5t6006048000018790002353594D303637d0
Full Device Path:                               pmoney4:/dev/rdisk/c5t6006048000018799018253594D303637d0
Full Device Path:                               pmoney3:/dev/rdisk/c5t6006048000018799018253594D303637d0
Replication:                                   srdf
default_fencing:                               global

#
```

▼ プライマリルームの完全な失敗後に EMC SRDF データを回復する方法

この手順では、キャンパスクラスタのプライマリルームが完全に失敗し、プライマリルームがセカンダリルームにフェイルオーバーして、プライマリルームがオンラインに戻ったとき、データ回復を実行します。キャンパスクラスタのプライマリルームは、プライマリノードとストレージサイトです。ルームの完全な失敗には、そのルームのホストとストレージの両方の不具合が含まれます。プライマリルームが失敗した場合、Oracle Solaris Cluster は自動的にセカンダリルームにフェイルオーバーし、セカンダリルームのストレージデバイスを読み書き可能にし、対応するデバイスグループとリソースグループのフェイルオーバーを有効にします。

プライマリルームがオンラインに戻ったら、セカンダリルームに書き込まれた SRDF デバイスグループからデータを手動で回復し、データを再同期できます。この手順では、元のセカンダリルーム(この手順では、セカンダリルームに *phys-campus-2* を使用)のデータを元のプライマリルーム (*phys-campus-1*) と同期して、SRDF デバイスグループを回復します。また、この手順では、SRDF デバイスグループタイプを、*phys-campus-2* では RDF1 に変更し、*phys-campus-1* では RDF2 に変更します。

始める前に 手動でフェイルオーバーを実行する前に、EMC 複製グループおよび DID デバイスを構成し、EMC 複製グループを登録する必要があります。Solaris Volume Manager デバイスグループの作成については、[129 ページの「デバイスグループを追加および登録する方法 \(Solaris Volume Manager\)」](#)を参照してください。

注- これらの説明は、プライマリルームが完全にフェイルオーバーしてからオンラインに戻ったあとに、SRDF データを手動で回復するための 1 つの方法を示しています。その他の方法については、EMC のドキュメントを確認してください。

これらの手順を実行するには、キャンパスクラスタのプライマリルームにログインします。次の手順では、*dg1* は SRDF デバイスグループ名です。失敗した時点では、この手順のプライマリルームは *phys-campus-1* で、セカンダリルームは *phys-campus-2* です。

- 1 キャンパスクラスタのプライマリルームにログインし、スーパーユーザーになるか、**solaris.cluster.modify RBAC** の承認を提供する役割になります。
- 2 プライマリルームから、**symrdf** コマンドを使用して RDF デバイスの複製ステータスに対するクエリを実行し、これらのデバイスに関する情報を表示します。

```
phys-campus-1# symrdf -g dg1 query
```

ヒント-split 状態にあるデバイスグループは同期されません。

- 3 RDF ペア状態が **split** で、デバイスグループタイプが **RDF1** の場合、**SRDF** デバイスグループのフェイルオーバーを強制実行します。
phys-campus-1# **symrdf -g dg1 -force failover**
- 4 RDF デバイスのステータスを表示します。
phys-campus-1# **symrdf -g dg1 query**
- 5 フェイルオーバー後、フェイルオーバーした **RDF** デバイスのデータをスワップできます。
phys-campus-1# **symrdf -g dg1 swap**
- 6 RDF デバイスに関するステータスおよびその他の情報を確認します。
phys-campus-1# **symrdf -g dg1 query**
- 7 プライマリルームで **SRDF** デバイスグループを確立します。
phys-campus-1# **symrdf -g dg1 establish**
- 8 デバイスグループが同期状態にあり、デバイスグループタイプが **RDF2** であることを確認します。
phys-campus-1# **symrdf -g dg1 query**

例 5-20 プライマリサイトフェイルオーバー後の EMC SRDF データの手動回復

この例では、キャンパスクラスタのプライマリルームがフェイルオーバーし、セカンダリルームがテイクオーバーしてデータを記録し、プライマリルームがオンラインに戻ったあとで、EMC SRDF データを手動で回復するために必要な Oracle Solaris Cluster 固有の手順が示されています。この例では、SRDF デバイスグループは *dg1* と呼ばれ、標準論理デバイスは DEV001 です。失敗した時点では、プライマリルームは

phys-campus-1 で、セカンダリルームは *phys-campus-2* です。キャンパスクラスタのプライマリルーム *phys-campus-1* から各手順を実行します。

```
phys-campus-1# symrdf -g dg1 query | grep DEV
DEV001 0012RW 0 0NR 0012RW 2031 0 S.. Split

phys-campus-1# symdg list | grep RDF
dg1 RDF1 Yes 00187990182 1 0 0 0 0

phys-campus-1# symrdf -g dg1 -force failover
...

phys-campus-1# symrdf -g dg1 query | grep DEV
DEV001 0012 WD 0 0 NR 0012 RW 2031 0 S.. Failed Over

phys-campus-1# symdg list | grep RDF
dg1 RDF1 Yes 00187990182 1 0 0 0 0

phys-campus-1# symrdf -g dg1 swap
...

phys-campus-1# symrdf -g dg1 query | grep DEV
DEV001 0012 WD 0 0 NR 0012 RW 0 2031 S.. Suspended

phys-campus-1# symdg list | grep RDF
dg1 RDF2 Yes 000187990182 1 0 0 0 0

phys-campus-1# symrdf -g dg1 establish
...

phys-campus-1# symrdf -g dg1 query | grep DEV
DEV001 0012 WD 0 0 RW 0012 RW 0 0 S.. Synchronized

phys-campus-1# symdg list | grep RDF
dg1 RDF2 Yes 000187990182 1 0 0 0 0
```

クラスタファイルシステムの管理の概要

クラスタファイルシステムの管理のために特別な Oracle Solaris Cluster コマンドは必要ありません。クラスタファイルシステムを管理するには、他の Oracle Solaris ファイルシステムを管理するときと同じように、Oracle Solaris の標準のファイルシステムコマンド (`mount` や `newfs` など) を使用します。クラスタファイルシステムをマウントするには、`mount` コマンドに `-g` オプションを指定します。クラスタファイルシステムはブート時に自動的にマウントすることもできます。クラスタファイルシステムは、グローバルクラスタ内の投票ノードからのみ認識できます。非投票ノードからクラスタファイルシステムのデータにアクセスできるようにする必要がある場合は、[zoneadm\(1M\)](#) または `HASStoragePlus` を使用して、データを非投票ノードにマップします。

注- クラスタファイルシステムがファイルを読み取るとき、ファイルシステムはファイルのアクセス時間を更新しません。

クラスタファイルシステムの制限事項

次に、クラスタファイルシステム管理に適用される制限事項を示します。

- `unlink(1M)` コマンドは、空ではないディレクトリではサポートされていません。
- `lockfs -d` コマンドはサポートされません。回避方法として、`lockfs -n` を使用してください。
- クラスタファイルシステムをマウントし直すとき、`directio` マウントオプションは指定できません。
- ルートファイルシステムに ZFS がサポートされていますが、重要な例外が 1 つあります。グローバルデバイスファイルシステムにブートディスクの専用パーティションを使用する場合、ファイルシステムとして UFS のみを使用してください。グローバルデバイス名前空間には、UFS ファイルシステムで動作しているプロキシファイルシステム (PxFS) が必要です。ただし、グローバルデバイス名前空間の UFS ファイルシステムは、ルート (/) ファイルシステムやほかのルートファイルシステム (/var や /home など) 用の ZFS ファイルシステムと共存できません。また、lofi デバイスを使用してグローバルデバイス名前空間をホストする場合、ルートファイルシステムに対する ZFS の使用に関する制限はありません。

デバイスグループの管理

クラスタの要件の変化により、クラスタ上のデバイスグループの追加、削除、または変更が必要となる場合があります。Oracle Solaris Cluster には、このような変更を行うために使用できる、`clsetup` と呼ばれる対話型インタフェースがあります。`clsetup` は `cluster` コマンドを生成します。生成されるコマンドについては、各説明の後にある例を参照してください。次の表に、デバイスグループを管理するためのタスクを示し、またこのセクションの適切な手順へのリンクを示します。



注意- ほかのノードが有効なクラスタメンバーであり、それらのノードの少なくとも 1 つがディスクセットを持つ場合は、クラスタの外側でブートされるクラスタノードで `metaset -s setname -f -t` を実行しないでください。

注 - Oracle Solaris Cluster ソフトウェアは、クラスタ内のディスクデバイスやテープデバイスごとに、`raw` ディスクデバイスグループを自動的に作成します。ただし、クラスタデバイスグループはグローバルデバイスとしてアクセスされるまでオフラインのままです。

表 5-4 タスクマップ: デバイスグループの管理

タスク	手順
<code>cldevice populate</code> コマンドを使用することにより、再構成のリポートを行わずにグローバルデバイス名前空間を更新する	124 ページの「グローバルデバイス名前空間を更新する方法」
グローバルデバイス名前空間で使用する <code>lofi</code> デバイスのサイズを変更する	125 ページの「グローバルデバイス名前空間で使用する <code>lofi</code> デバイスのサイズを変更する方法」
既存のグローバルデバイス名前空間を移動する	127 ページの「専用パーティションから <code>lofi</code> デバイスにグローバルデバイス名前空間を移行する方法」 128 ページの「 <code>lofi</code> デバイスから専用パーティションにグローバルデバイス名前空間を移行する方法」
<code>metaset</code> コマンドを使用することにより、Solaris Volume Manager ディスクセットを追加し、それらをデバイスグループとして登録する	129 ページの「デバイスグループを追加および登録する方法 (Solaris Volume Manager)」
<code>cldevicegroup</code> コマンドを使用することにより、 <code>raw</code> ディスクデバイスグループを追加および登録する	132 ページの「デバイスグループ (<code>raw</code> ディスク) を追加および登録する方法」
<code>cldevicegroup</code> コマンドを使用することにより、ZFS に名前付きデバイスグループを追加する	132 ページの「複製デバイスグループ (ZFS) の追加と登録方法」
<code>metaset</code> コマンドおよび <code>metaclear</code> コマンドを使用することにより、構成から Solaris Volume Manager デバイスグループを削除する	134 ページの「デバイスグループを削除して登録を解除する方法 (Solaris Volume Manager)」
<code>cldevicegroup</code> 、 <code>metaset</code> 、および <code>clsetup</code> コマンドを使用することにより、すべてのデバイスグループからノードを削除する	134 ページの「すべてのデバイスグループからノードを削除する方法」
<code>metaset</code> コマンドを使用することにより、Solaris Volume Manager デバイスグループからノードを削除する	135 ページの「デバイスグループからノードを削除する方法 (Solaris Volume Manager)」

表 5-4 タスクマップ: デバイスグループの管理 (続き)

タスク	手順
cldevicegroup コマンドを使用することにより、raw ディスクデバイスグループからノードを削除する	137 ページの「raw ディスクデバイスグループからノードを削除する方法」
clsetup を使用して cldevicegroup を生成することにより、デバイスグループのプロパティを変更する	139 ページの「デバイスグループのプロパティを変更する方法」
cldevicegroup show コマンドを使用することにより、デバイスグループとプロパティを表示する	143 ページの「デバイスグループ構成の一覧を表示する方法」
clsetup を使用して cldevicegroup を生成することにより、デバイスグループのセカンダリノードの希望数を変更する	141 ページの「デバイスグループのセカンダリノードの希望数を設定する方法」
cldevicegroup switch コマンドを使用することにより、デバイスグループのプライマリノードを切り替える	145 ページの「デバイスグループのプライマリノードを切り替える」
metaset コマンドまたは vxvg コマンドを使用することにより、デバイスグループを保守状態にする	146 ページの「デバイスグループを保守状態にする方法」

▼ グローバルデバイス名前空間を更新する方法

新しいグローバルデバイスを追加するときに、cldevice populate コマンドを実行して手動でグローバルデバイス名前空間を更新します。

注- コマンドを実行するノードがクラスタのメンバーでない場合は、cldevice populate コマンドを実行しても無効です。また、/global/.devices/node@nodeID ファイルシステムがマウントされていない場合も、コマンドは無効になります。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 solaris.cluster.modify を提供する役割になります。
- 2 クラスタ内の各ノードで、devfsadm(1M) コマンドを実行します。
このコマンドは、すべてのノードで同時に実行できます。
- 3 名前空間を再構成します。

```
# cldevice populate
```

- 4 ディスクセットを作成する前に、各ノードで **cldevice populate** コマンドが完了していることを確認します。

cldevice コマンドは、1つのノードだけで実行された場合でも、すべてのノードでリモートから自身を呼び出します。cldevice populate コマンドが処理を終了したかどうかを確認するには、クラスタの各ノードで次のコマンドを実行します。

```
# ps -ef | grep cldevice populate
```

例 5-21 グローバルデバイス名前空間を更新する

次の例に、cldevice populate コマンドを正しく実行することにより生成される出力を示します。

```
# devfsadm
cldevice populate
Configuring the /dev/global directory (global devices)...
obtaining access to all attached disks
reservation program successfully exiting
# ps -ef | grep cldevice populate
```

▼ グローバルデバイス名前空間で使用する lofi デバイスのサイズを変更する方法

グローバルクラスタの1つ以上のノードのグローバルデバイス名前空間で lofi デバイスを使用する場合は、次の手順を使用してデバイスのサイズを変更します。

- 1 サイズを変更するグローバルデバイス名前空間の lofi デバイスのノードで、スーパーユーザーになるか、RBAC の承認 **solaris.cluster.modify** を提供する役割になります。
- 2 ノードからサービスを退避させ、ノードを非クラスタモードでリブートします。
これは、この手順の実行中にグローバルデバイスがこのノードからサービスを提供されないようにするために行います。手順については、[82 ページの「非クラスタモードでノードをブートする方法」](#)を参照してください。
- 3 グローバルデバイスのファイルシステムをマウント解除し、その lofi デバイスを切り離します。

グローバルデバイスファイルシステムはローカルにマウントされます。

```
phys-schost# umount /global/.devices/node\@'clinfo -n' > /dev/null 2>&1
```

Ensure that the lofi device is detached

```
phys-schost# lofiadm -d /.globaldevices
```

The command returns no output if the device is detached

注--m オプションを使用してファイルシステムがマウントされた場合、`mnttab` ファイルにエントリは追加されません。`umount` コマンドによって次のような警告が報告される場合があります。

```
umount: warning: /global/.devices/node@2 not in mnttab      =====>>>>
not mounted
```

この警告は無視してもかまいません。

- 4 `/.globaldevices` ファイルを削除し、必要なサイズで再作成します。

次の例は、サイズが 200M バイトの新しい `/.globaldevices` ファイルの作成を示しています。

```
phys-schost# rm /.globaldevices
phys-schost# mkfile 200M /.globaldevices
```

- 5 グローバルデバイス名前空間の新しいファイルシステムを作成します。

```
phys-schost# lofiadm -a /.globaldevices
phys-schost# newfs 'lofiadm /.globaldevices' < /dev/null
```

- 6 クラスタモードでノードをブートします。

グローバルデバイスが新しいファイルシステムに生成されました。

```
phys-schost# reboot
```

- 7 サービスを実行するノードに移行します。

グローバルデバイス名前空間を移行する

専用パーティションでグローバルデバイス名前空間を作成するのではなく、ループバックファイルインタフェース (lofi) デバイス上に名前空間を作成できます。この機能は、Oracle Solaris 10 OS とともに事前にインストールされたシステムに Oracle Solaris Cluster ソフトウェアをインストールする場合に便利です。

注- ルートファイルシステムに ZFS がサポートされていますが、重要な例外が 1 つあります。グローバルデバイスファイルシステムにブートディスクの専用パーティションを使用する場合、ファイルシステムとして UFS のみを使用してください。グローバルデバイス名前空間には、UFS ファイルシステムで動作しているプロキシファイルシステム (PxFS) が必要です。ただし、グローバルデバイス名前空間の UFS ファイルシステムは、ルート (/) ファイルシステムやほかのルートファイルシステム (/var や /home など) 用の ZFS ファイルシステムと共存できます。また、lofi デバイスを使用してグローバルデバイス名前空間をホストする場合、ルートファイルシステムに対する ZFS の使用に関する制限はありません。

次の手順は、既存のグローバルデバイス名前空間を専用パーティションから lofi デバイスまたはその逆に移行する方法を説明しています。

- 127 ページの「専用パーティションから lofi デバイスにグローバルデバイス名前空間を移行する方法」
- 128 ページの「lofi デバイスから専用パーティションにグローバルデバイス名前空間を移行する方法」

▼ 専用パーティションから lofi デバイスにグローバルデバイス名前空間を移行する方法

- 1 名前空間の場所を変更したいグローバルクラスタ投票ノードのスーパーユーザーになります。
- 2 ノードからサービスを退避させ、ノードを非クラスタモードでリブートします。
これは、この手順の実行中にグローバルデバイスがこのノードからサービスを提供されないようにするために行います。手順については、[82 ページの「非クラスタモードでノードをブートする方法」](#)を参照してください。
- 3 `/.globaldevices` という名前のファイルがノードに存在しないことを確認します。ファイルが存在する場合は、削除します。
- 4 lofi デバイスを作成します。

```
# mkfile 100m /.globaldevices# lofiadm -a /.globaldevices
# LOFI_DEV='lofiadm /.globaldevices'
# newfs 'echo ${LOFI_DEV} | sed -e 's/lofi/rlofi/g'' < /dev/null# lofiadm -d /.globaldevices
```

- 5 `/etc/vfstab` ファイルで、グローバルデバイス名前空間エントリをコメントアウトします。このエントリには、`/global/.devices/node@nodeID` で始まるマウントパスがあります。
- 6 グローバルデバイスパーティション `/global/.devices/node@nodeID` をアンマウントします。
- 7 `globaldevices` および `scmountdev` SMF サービスを無効にし再度有効にします。

```
# svcadm disable globaldevices# svcadm disable scmountdev
# svcadm enable scmountdev
# svcadm enable globaldevices
```

lofi デバイスは現在 `/.globaldevices` に作成され、グローバルデバイスファイルシステムとしてマウントされています。
- 8 パーティションから lofi デバイスへ移行したいグローバルデバイス名前空間のある他のノードでもこのステップを繰り返します。

- 9 1つのノードから、グローバルデバイス名前空間を生成します。

```
# /usr/cluster/bin/cldevice populate
```

各ノードで、コマンドが処理を完了したことを確認してから、クラスタに対する以降の操作を実行してください。

```
# ps -ef \ grep cldevice populate
```

グローバルデバイス名前空間は、現在 `lofi` デバイスにあります。

- 10 サービスを実行するノードに移行します。

▼ **lofi** デバイスから専用パーティションにグローバルデバイス名前空間を移行する方法

- 1 名前空間の場所を変更したいグローバルクラスタ投票ノードのスーパーユーザーになります。
- 2 ノードからサービスを退避させ、ノードを非クラスタモードでリブートします。
これは、この手順の実行中にグローバルデバイスがこのノードからサービスを提供されないようにするために行います。手順については、[82 ページの「非クラスタモードでノードをブートする方法」](#)を参照してください。

- 3 ノードのローカルディスクで、次の要件を満たす新しいパーティションを作成します。

- サイズが 512 M バイト以上
- UFS ファイルシステムの使用

- 4 グローバルデバイスファイルシステムとしてマウントする新しいパーティションに、`/etc/vfstab` ファイルへのエントリを追加します。

- 現在のノードのノード ID を指定します。

```
# /usr/sbin/clinfo -nnode ID
```

- 次の形式を使用して、`/etc/vfstab` ファイルに新しいエントリを作成します。

```
blockdevice rawdevice /global/.devices/node@nodeID ufs 2 no global
```

たとえば、使用するパーティションが `/dev/did/rdisk/d5s3` の場合、`/etc/vfstab` ファイルに追加する新しいエントリは、`/dev/did/dsk/d5s3 /dev/did/rdisk/d5s3 /global/.devices/node@3 ufs 2 no global` となります。

- 5 グローバルデバイスパーティション `/global/.devices/node@nodeID` のマウントを解除します。

- 6 `/.globaldevices` ファイルに関連付けられた `lofi` デバイスを削除します。

```
# lofiadm -d /.globaldevices
```

- 7 `/.globaldevices` ファイルを削除します。

```
# rm /.globaldevices
```

- 8 `globaldevices` および `scmountdev` SMF サービスを無効にし再度有効にします。

```
# svcadm disable globaldevices# svcadm disable scmountdev
# svcadm enable scmountdev
# svcadm enable globaldevices
```

パーティションは現在グローバルデバイス名前空間ファイルシステムとしてマウントされています。

- 9 `lofi` デバイスからパーティションへ移行したいグローバルデバイス名前空間のある他のノードでもこのステップを繰り返します。

- 10 クラスタモードでブートします。クラスタの 1 つのノードから、`cldevice populate` コマンドを実行し、グローバルデバイス名前空間を生成します。

```
# /usr/cluster/bin/cldevice populate
```

クラスタのすべてのノードで処理が完了したことを確認してから、ノードに対する作業を実行してください。

```
# ps -ef | grep cldevice populate
```

グローバルデバイス名前空間は、現在専用パーティションにあります。

- 11 サービスを実行するノードに移行します。

デバイスグループを追加および登録する

Solaris Volume Manager、ZFS、または raw ディスクのデバイスグループを追加および登録できます。

▼ デバイスグループを追加および登録する方法 (Solaris Volume Manager)

`metaset` コマンドを使用して Solaris Volume Manager ディスクセットを作成し、そのディスクセットを Oracle Solaris Cluster デバイスグループとして登録します。デバイスグループには、ディスクセットを登録するときにディスクセットに割り当てた名前が自動的に割り当てられます。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。



注意 - 作成する Oracle Solaris Cluster デバイスグループ (Solaris Volume Manager、または raw ディスク) の名前は、複製されたデバイスグループと同じ名前にしてください。

- 1 ディスクセットを作成するディスクに接続されたノードのいずれかで、スーパーユーザーになるか、RBAC の承認 **solaris.cluster.modify** を提供する役割になります。
- 2 **Solaris Volume Manager** ディスクセットを追加し、このディスクセットをデバイスグループとして **Oracle Solaris Cluster** に登録します。複数所有者のディスクグループを作成するには、**-M** オプションを使用します。

```
# metaset -s diskset -a -M -h nodelist
```

-s diskset 作成するディスクセットを指定します。

-a -h nodelist ディスクセットをマスターできるノードの一覧を追加します。

-M ディスクグループを複数所有者として指定します。

注 - **metaset** コマンドを実行して設定した、クラスタ上の Solaris Volume Manager デバイスグループは、そのデバイスグループに含まれるノード数に関わらず、デフォルトでセカンダリノードになります。デバイスグループが作成されたあと、**clsetup** ユーティリティを使用することで、セカンダリノードの希望数を変更できます。ディスクのフェイルオーバーの詳細については、[141 ページの「デバイスグループのセカンダリノードの希望数を設定する方法」](#)を参照してください。

- 3 複製されたデバイスグループを構成している場合は、そのデバイスグループの複製プロパティを設定します。

```
# cldevicegroup sync devicegroup
```

- 4 デバイスグループが追加されたことを確認します。
デバイスグループ名は **metaset** に指定したディスクセット名と一致します。

```
# cldevicegroup list
```

- 5 **DID** マッピングの一覧を表示します。

```
# cldevice show | grep Device
```

- ディスクセットをマスターする (またはマスターする可能性がある) クラスタノードによって共有されているドライブを選択します。

- ディスクセットにドライブを追加する際は、`/dev/did/rdisk/dN`の形式の完全なDIDデバイス名を使用してください。

次の例では、DID デバイス `/dev/did/rdisk/d3` のエントリは、ドライブが `phys-schost-1` および `phys-schost-2` によって共有されていることを示しています。

```
=== DID Device Instances ===
DID Device Name:                /dev/did/rdisk/d1
Full Device Path:               phys-schost-1:/dev/rdsk/c0t0d0
DID Device Name:                /dev/did/rdisk/d2
Full Device Path:               phys-schost-1:/dev/rdsk/c0t6d0
DID Device Name:                /dev/did/rdisk/d3
Full Device Path:               phys-schost-1:/dev/rdsk/c1t1d0
Full Device Path:               phys-schost-2:/dev/rdsk/c1t1d0
...
```

- 6 ディスクセットにドライブを追加します。
完全な DID パス名を使用します。

```
# metaset -s setname -a /dev/did/rdisk/dN
-s setname   デバイスグループ名と同じである、ディスクセット名を指定します。
-a          ディスクセットにドライブを追加します。
```

注 - ディスクセットにドライブを追加するときは、下位デバイス名 (`cNtXdY`) は使用しないでください。下位レベルデバイス名はローカル名であり、クラスタ全体で一意ではないため、この名前を使用するとディスクセットがスイッチオーバーできなくなる可能性があります。

- 7 新しいディスクセットとドライブのステータスを検査します。

```
# metaset -s setname
```

例 5-22 Solaris Volume Manager デバイスグループの追加

次の例は、ディスクドライブ `/dev/did/rdsk/d1` および `/dev/did/rdsk/d2` を持つディスクセットおよびデバイスグループの作成を示し、デバイスグループが作成されたことを確認しています。

```
# metaset -s dg-schost-1 -a -h phys-schost-1

# cldevicegroup list
dg-schost-1
metaset -s dg-schost-1 -a /dev/did/rdsk/d1 /dev/did/rdsk/d2
```

▼ デバイスグループ (raw ディスク) を追加および登録する方法

Oracle Solaris Cluster ソフトウェアでは、他のボリュームマネージャーに加え、raw ディスクデバイスグループを使用できます。Oracle Solaris Cluster を最初に構成する際、クラスタ内の raw デバイスごとにデバイスグループが自動的に構成されます。ここで説明する手順を使用して、これらの自動作成されたデバイスグループを Oracle Solaris Cluster ソフトウェアでできるように再構成します。

次の理由のため、raw ディスクタイプの新しいデバイスグループを作成します。

- 複数の DID をデバイスグループに追加したい
- デバイスグループの名前を変更する必要がある
- cldg コマンドの -v オプションを使用せずにデバイスグループのリストを作成したい



注意-複製したデバイスにデバイスグループを作成する場合は、作成するデバイスグループ名 (Solaris Volume Manager、または raw ディスク) は複製したデバイスグループの名前と同じにする必要があります。

- 1 使用する各デバイスを特定し、事前に規定されたデバイスグループの構成を解除します。

次のコマンドは、d7 および d8 に対する定義済みのデバイスグループを除去します。

```
paris-1# cldevicegroup disable dsk/d7 dsk/d8
paris-1# cldevicegroup offline dsk/d7 dsk/d8
paris-1# cldevicegroup delete dsk/d7 dsk/d8
```

- 2 必要なデバイスを含む、新しい raw ディスクデバイスグループを作成します。

次のコマンドは、グローバルデバイスグループ rawdg を作成します。このデバイスグループに d7 および d8 が収められます。

```
paris-1# cldevicegroup create -n phys-paris-1,phys-paris-2 -t rawdisk
-d d7,d8 rawdg
paris-1# /usr/cluster/lib/dcs/cldg show rawdg -d d7 rawdg
paris-1# /usr/cluster/lib/dcs/cldg show rawdg -d d8 rawdg
```

▼ 複製デバイスグループ (ZFS) の追加と登録方法

ZFS を複製するには、名前付きデバイスグループを作成し、zpool に属するディスクをリストする必要があります。デバイスは、一度に 1 つのデバイスグループのみに属することができるため、デバイスを含む Oracle Solaris Cluster デバイスグループがすでにある場合、そのデバイスを新しい ZFS デバイスグループに追加する前にそのグループを削除する必要があります。

作成する Oracle Solaris Cluster デバイスグループ (Solaris Volume Manager、または raw ディスク) の名前は、複製されたデバイスグループと同じ名前にしてください。



注意 - サードパーティーのデータ複製テクノロジーを使用した ZFS の全面的なサポートは保留中です。ZFS サポートの更新情報については、最新の Oracle Solaris Cluster リリースノートを参照してください。

- 1 **zpool** のデバイスに対応するデフォルトデバイスグループを削除してください。
たとえば、2つのデバイス `/dev/did/dsk/d2` と `/dev/did/dsk/d13` を含む `mypool` と呼ばれる **zpool** がある場合、`d2` と `d13` と呼ばれる2つのデフォルトデバイスグループを削除する必要があります。


```
# cldevicegroup offline dsk/d2 dsk/d13
# cldevicegroup delete dsk/d2 dsk/d13
```
- 2 手順1で削除したデバイスグループの **DID** に対応する **DID** を使って名前付きデバイスグループを作成します。


```
# cldevicegroup create -n pnode1,pnode2 -d d2,d13 -t rawdisk mypool
```

このアクションでは、`mypool`(`zpool` と同じ名前) と呼ばれるデバイスグループが作成され、raw デバイス `/dev/did/dsk/d2` と `/dev/did/dsk/d13` を管理します。
- 3 それらのデバイスを含む **zpool** を作成します。


```
# zpool create mypool mirror /dev/did/dsk/d2 /dev/did/dsk/d13
```
- 4 リソースグループを作成し、ノードリストに唯一のグローバルゾーンのある複製したデバイス (デバイスグループ内) の移行を管理します。


```
# clrg create -n pnode1,pnode2 migrate_truecopydg-rg
```
- 5 手順4で作成したリソースグループに **hasp-rs** リソースを作成し、**globaldevicepaths** プロパティを raw ディスクのデバイスグループに設定します。このデバイスグループは手順2で作成しました。


```
# clrs create -t HASStoragePlus -x globaldevicepaths=mypool -g \
migrate_truecopydg-rg hasp2migrate_mypool
```
- 6 アプリケーションリソースグループがローカルゾーンで実行される場合、適切なローカルゾーンが含まれるノードリストを使用して新しいリソースグループを作成します。ローカルゾーンに対応するグローバルゾーンは、手順4で作成したリソースグループのノードリストに含まれている必要があります。このリソースグループの **rg_affinities** プロパティの **+++** 値を、手順4で作成したリソースグループに設定します。


```
# clrg create -n pnode1:zone-1,pnode2:zone-2 -p \
RG_affinities=+++migrate_truecopydg-rg sybase-rg
```

- 7 手順4または手順6のいずれかで作成したリソースグループ内に、手順3で作成した **zpool** の **HAStoragePlus** リソース (**hasp-rs**) を作成します。 **resource_dependencies** プロパティを、手順5で作成した **hasp-rs** リソースに設定します。

```
# clrs create -g sybase-rg -t HAStoragePlus -p zpools=mypool \
-p resource_dependencies=hasp2migrate_mypool \
-p ZpoolsSearchDir=/dev/did/dsk hasp2import_mypool
```

- 8 デバイスグループ名が必要な場合には、この新しいリソースグループ名を使用します。

デバイスグループ名を保守する

デバイスグループに対して様々な管理タスクを実行することができます。

デバイスグループを削除して登録を解除する方法 (Solaris Volume Manager)

デバイスグループは Oracle Solaris Cluster に登録されている Solaris Volume Manager ディスクセットです。Solaris Volume Manager デバイスグループを削除するには、**metaclear** と **metaset** コマンドを使用します。これらのコマンドは、Oracle Solaris Cluster デバイスグループと同じ名前を持つデバイスグループを削除して、ディスクグループの登録を解除します。

ディスクセットを削除する方法については、Solaris Volume Manager のドキュメントを参照してください。

▼ すべてのデバイスグループからノードを削除する方法

すべてのデバイスグループの潜在的なプライマリノードからクラスタノードを削除する場合は、この手順を使用します。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 すべてのデバイスグループの潜在的なプライマリノードとして削除するノード上で、スーパーユーザーになるか、RBACの承認 `solaris.cluster.modify` を提供する役割になります。
- 2 削除するノードがメンバーになっているデバイスグループ(複数可)を確認します。
各デバイスグループの `Device group node list` からこのノード名を検索します。

```
# cldevicegroup list -v
```
- 3 手順2で特定したデバイスグループの中に、デバイスグループタイプが **SVM** のものがある場合、そのタイプの各デバイスグループに対して135ページの「[デバイスグループからノードを削除する方法 \(Solaris Volume Manager\)](#)」の手順を実行します。
- 4 削除するノードがメンバーになっている **raw** デバイスディスクグループを特定します。

```
# cldevicegroup list -v
```
- 5 手順4で表示されたデバイスグループの中に、デバイスグループタイプが **Disk** または **Local_Disk** のものがある場合、これらの各デバイスグループに対して、137ページの「[raw ディスクデバイスグループからノードを削除する方法](#)」の手順を実行します。
- 6 すべてのデバイスグループの潜在的なプライマリノードのリストからノードが削除されていることを確認します。
ノードがどのデバイスグループの潜在的なプライマリノードのリストにも存在しなければ、このコマンドは何も返しません。

```
# cldevicegroup list -v nodename
```

▼ デバイスグループからノードを削除する方法 (Solaris Volume Manager)

Solaris Volume Manager デバイスグループの潜在的なプライマリノードのリストからクラスタノードを削除するには、次の手順を使用します。ノードを削除したいグループデバイスごとに `metaset` コマンドを繰り返します。



注意-ほかのノードが有効なクラスタメンバーであり、それらのノードの少なくとも1つがディスクセットを持つ場合は、クラスタの外側でブートされるクラスタノードで `metaset -s setname -f -t` を実行しないでください。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 ノードがまだデバイスグループのメンバーであり、かつ、このデバイスグループが **Solaris Volume Manager** デバイスグループであることを確認します。

Solaris Volume Manager のデバイスグループは、デバイスグループタイプが SDS/SVM のものです。

```
phys-schost-1% cldevicegroup show devicegroup
```

- 2 どのノードがデバイスグループの現在のプライマリノードであるかを特定します。

```
# cldevicegroup status devicegroup
```

- 3 変更したいデバイスグループを所有しているノードでスーパーユーザーになります。

- 4 デバイスグループからこのノードのホスト名を削除します。

```
# metaset -s setname -d -h nodelist
```

-s *setname* デバイスグループの名前を指定します。

-d -h で指定されたノードをデバイスグループから削除します。

-h *nodelist* 削除するノード (複数可) のノード名を指定します。

注- 更新が完了するまでに数分間かかることがあります。

コマンドが正常に動作しない場合は、コマンドに -f (force) オプションを追加します。

```
# metaset -s setname -d -f -h nodelist
```

- 5 潜在的なプライマリノードとしてノードを削除するデバイスグループごとに[手順 4](#)を繰り返します。

- 6 デバイスグループからノードが削除されたことを確認します。
デバイスグループ名は metaset に指定したディスクセット名と一致します。

```
phys-schost-1% cldevicegroup list -v devicegroup
```

例 5-23 デバイスグループからノードを削除する (Solaris Volume Manager)

次に、デバイスグループ構成からホスト名 phys-schost-2 を削除する例を示します。この例では、指定したデバイスグループから phys-schost-2 を潜在的なプライマ

リノードとして削除します。cldevicegroup show コマンドを実行することにより、ノードが削除されていることを確認します。削除したノードが画面に表示されていないことを確認します。

```
[Determine the Solaris Volume Manager
device group for the node:]
# cldevicegroup show dg-schost-1
=== Device Groups ===

Device Group Name:          dg-schost-1
Type:                      SVM
failback:                   no
Node List:                  phys-schost-1, phys-schost-2
preferenced:                yes
numsecondaries:             1
diskset name:               dg-schost-1
[Determine which node is the current primary for the device group:]
# cldevicegroup status dg-schost-1
=== Cluster Device Groups ===

--- Device Group Status ---

Device Group Name  Primary          Secondary        Status
-----
dg-schost-1       phys-schost-1    phys-schost-2    Online
[Become superuser on the node that currently owns the device group.]
[Remove the host name from the device group:]
# metaset -s dg-schost-1 -d -h phys-schost-2
[Verify removal of the node:]
phys-schost-1% cldevicegroup list -v dg-schost-1
=== Cluster Device Groups ===

--- Device Group Status ---

Device Group Name  Primary          Secondary        Status
-----
dg-schost-1       phys-schost-1    -                Online
```

▼ raw ディスクデバイスグループからノードを削除する方法

raw ディスクデバイスグループの潜在的プライマリノードリストからクラスタノードを削除する場合は、この手順を使用します。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタ内のノード、ただし削除するノード以外のノード上で、スーパーユーザーになるか、RBACの承認 `solaris.cluster.read` および `solaris.cluster.modify` を提供する役割になります。
- 2 削除されるノードに接続されたデバイスグループを特定し、どれが **raw** ディスクデバイスグループであるかを判別します。

```
# cldevicegroup show -n nodename -t rawdisk +
```
- 3 すべての **Local_Diskraw** ディスクデバイスグループの **localonly** プロパティを無効にします。

```
# cldevicegroup set -p localonly=false devicegroup
```

`localonly` プロパティについての詳細は、[cldevicegroup\(1CL\)](#) のマニュアルページを参照してください。
- 4 削除するノードに接続されているすべての **raw** ディスクデバイスグループの **localonly** プロパティが無効になっていることを確認します。

デバイスグループタイプ `Disk` は、この **raw** ディスクデバイスグループの `localonly` プロパティが無効になっていることを表します。

```
# cldevicegroup show -n nodename -t rawdisk -v +
```
- 5 **手順2** で特定されたすべての **raw** ディスクデバイスグループからノードを削除します。

この手順は、削除するノードに接続されている **raw** ディスクデバイスグループごとに行う必要があります。

```
# cldevicegroup remove-node -n nodename devicegroup
```

例 5-24 raw デバイスグループからノードを削除する

この例では、**raw** ディスクデバイスグループからノード (`phys-schost-2`) を削除します。すべてのコマンドは、クラスタの別のノード (`phys-schost-1`) から実行します。

```
[Identify the device groups connected to the node being removed, and determine which are raw-disk
 device groups:]
phys-schost-1# cldevicegroup show -n phys-schost-2 -t rawdisk -v +
Device Group Name:      dsk/d4
Type:                   Disk
failback:               false
Node List:              phys-schost-2
preferenced:            false
localonly:              false
autogen                 true
numsecondaries:         1
device names:           phys-schost-2

Device Group Name:      dsk/d2
Type:                   Disk
```

```

failback:                                true
Node List:                              pbrave2
preferenced:                             false
localonly:                               false
autogen                                  true
numsecondaries:                           1
diskgroup name:                           vxdg1

Device Group Name:                        dsk/d1
Type:                                     SVM
failback:                                 false
Node List:                               pbrave1, pbrave2
preferenced:                             true
localonly:                               false
autogen                                  true
numsecondaries:                           1
diskset name:                             ms1
(dsk/d4) Device group node list:  phys-schost-2
      (dsk/d2) Device group node list:  phys-schost-1, phys-schost-2
      (dsk/d1) Device group node list:  phys-schost-1, phys-schost-2
[Disable the localonly flag for each local disk on the node:]
phys-schost-1# cldevicegroup set -p localonly=false dsk/d4
[Verify that the localonly flag is disabled:]
phys-schost-1# cldevicegroup show -n phys-schost-2 -t rawdisk +
      (dsk/d4) Device group type:         Disk
      (dsk/d8) Device group type:         Local_Disk
[Remove the node from all raw-disk device groups:]

phys-schost-1# cldevicegroup remove-node -n phys-schost-2 dsk/d4
phys-schost-1# cldevicegroup remove-node -n phys-schost-2 dsk/d2
phys-schost-1# cldevicegroup remove-node -n phys-schost-2 dsk/d1

```

▼ デバイスグループのプロパティを変更する方法

デバイスグループの主所有権を確立する方法は、`preferenced` という所有権設定属性の設定に基づきます。この属性を設定していない場合は、ほかで所有されていないデバイスグループのプライマリ所有者が、そのグループ内のディスクへのアクセスを試みる最初のノードになります。一方、この属性を設定してある場合は、ノードが所有権の確立を試みる優先順位を指定する必要があります。

`preferenced` 属性を無効にすると、`failback` 属性も自動的に無効に設定されます。ただし、`preferenced` 属性を有効または再有効にする場合は、`failback` 属性を有効にするか無効にするかを選択できます。

`preferenced` 属性を有効または再有効にした場合は、プライマリ所有権の設定一覧でノードの順序を確立し直す必要があります。

この手順では、`clsetup` を使用して、Solaris Volume Manager デバイスグループの `preferenced` 属性と `failback` 属性を設定または設定解除します。

始める前に この手順を実行するには、属性値を変更するデバイスグループの名前が必要です。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 **solaris.cluster.read** および **solaris.cluster.modify** を提供する役割になります。
- 2 **clsetup** ユーティリティを起動します。
`# clsetup`
 メインメニューが表示されます。
- 3 デバイスグループの作業を行うには、デバイスグループやボリュームのオプションに対応する番号を入力します。
 「デバイスグループメニュー」が表示されます。
- 4 デバイスグループの重要なプロパティを変更するには、**Solaris Volume Manager** デバイスグループの重要なプロパティを変更するためのオプションに対応する番号を入力します。
 「デバイスグループのプロパティ変更メニュー」が表示されます。
- 5 デバイスグループのプロパティを変更するには、**preference**、**failback** などのプロパティを変更するオプションに対応する番号を入力します。
 指示に従って、デバイスグループの **preferenced** および **failback** オプションを設定します。
- 6 デバイスグループの属性が変更されたことを確認します。
 次のコマンドを実行し、表示されるデバイスグループ情報を確認します。

```
# cldevicegroup show -v devicegroup
```

例 5-25 デバイスグループのプロパティの変更

次に、**clsetup** でデバイスグループ (**dg-schost-1**) の属性値を設定したときに生成される **cldevicegroup** コマンドの例を示します。

```
# cldevicegroup set -p preferenced=true -p failback=true -p numsecondaries=1 \
-p nodelist=phys-schost-1,phys-schost-2 dg-schost-1
# cldevicegroup show dg-schost-1
```

```
=== Device Groups ===
```

```
Device Group Name:      dg-schost-1
Type:                  SVM
failback:              yes
Node List:              phys-schost-1, phys-schost-2
preferenced:           yes
numsecondaries:         1
diskset names:         dg-schost-1
```

▼ デバイスグループのセカンダリノードの希望数を設定する方法

`numsecondaries` プロパティは、プライマリノードに障害が発生した場合にグループをマスターできる、デバイスグループ内のノード数を指定します。デバイスサービスのセカンダリノードのデフォルト数は1です。この値には、1からデバイスグループ内で動作しているプライマリノード以外のプロバイダノード数までの任意の整数を設定できます。

この設定は、クラスタの性能と可用性のバランスをとるための重要な要因になります。たとえば、セカンダリノードの希望数を増やすと、クラスタ内で同時に複数の障害が発生した場合でも、デバイスグループが生き残る可能性が増えます。しかし、セカンダリノード数を増やすと、通常の動作中の性能が一様に下がります。通常、セカンダリノード数を減らすと、性能が上がりますが、可用性が下がります。しかし、セカンダリノード数を増やしても、必ずしも、当該のファイルシステムまたはデバイスグループの可用性が上がるわけではありません。詳細は、『[Oracle Solaris Cluster Concepts Guide](#)』の第3章「[Key Concepts for System Administrators and Application Developers](#)」を参照してください。

`numsecondaries` プロパティを変更すると、セカンダリノードの実際数と希望数の間に整合性がない場合、セカンダリノードはデバイスグループに追加されるか、またはデバイスグループから削除されます。

この手順では、`clsetup` ユーティリティを使用して、すべてのタイプのデバイスグループの `numsecondaries` プロパティを設定します。デバイスグループを構成する際のデバイスグループのオプションの詳細については、`cldevicegroup(1CL)` を参照してください。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBACの承認 `solaris.cluster.read` および `solaris.cluster.modify` を提供する役割になります。
- 2 `clsetup` ユーティリティを起動します。

```
# clsetup
```

 メインメニューが表示されます。
- 3 デバイスグループを使用して作業するには、「デバイスグループとボリューム (Device Groups and Volumes)」メニュー項目を選択します。
 「デバイスグループメニュー」が表示されます。
- 4 デバイスグループの重要なプロパティを変更するには、「デバイスグループのキープロパティを変更 (Change Key Properties of a Device Group)」メニュー項目を選択します。
 「デバイスグループのプロパティ変更メニュー」が表示されます。
- 5 セカンダリの希望数を変更するには、`numsecondaries` プロパティを変更するためのオプションに対応する番号を入力します。
 指示に従って、デバイスグループに構成したいセカンダリノードの希望数を入力します。対応する `cldevicegroup` コマンドが実行され、ログが出力され、ユーティリティは前のメニューに戻ります。
- 6 デバイスグループの構成を検証します。

```
# cldevicegroup show dg-schost-1
=== Device Groups ===
```

```
Device Group Name:      dg-schost-1
Type:                   Local_Disk      This might also be SDS.
failback:               yes
Node List:              phys-schost-1, phys-schost-2 phys-schost-3
preferenced:            yes
numsecondaries:         1
diskgroup names:       dg-schost-1
```

注-クラスタに登録されているディスクグループまたはボリュームの構成情報を変更する場合は、`clsetup` を使用してデバイスグループを再登録する必要があります。このような構成変更には、ボリュームの追加や削除、既存ボリュームのグループ、所有者、アクセス権の変更などがあります。構成変更後に登録を行うと、グローバルな名前空間が正しい状態になります。124 ページの「[グローバルデバイス名前空間を更新する方法](#)」を参照してください。

- 7 デバイスグループの属性が変更されたことを確認します。
 次のコマンドを実行して、表示されるデバイスグループ情報を確認します。
- ```
cldevicegroup show -v devicegroup
```

#### 例 5-26 セカンダリの希望数の変更 (Solaris Volume Manager)

次に、デバイスグループ (dg-schost-1) のセカンダリノードの希望数を構成するとき、`clsetup`によって生成される `cldevicegroup` コマンドの例を示します。この例では、ディスクグループとボリュームは以前に作成されているものと想定しています。

```
cldevicegroup set -p numsecondaries=1 dg-schost-1
cldevicegroup show -v dg-schost-1

=== Device Groups ===

Device Group Name: dg-schost-1
Type: SVM
failback: yes
Node List: phys-schost-1, phys-schost-2
preferenced: yes
numsecondaries: 1
diskset names: dg-schost-1
```

#### 例 5-27 セカンダリノードの希望数のデフォルト値への設定

次に、ヌル文字列値を使用して、セカンダリノードのデフォルト数を構成する例を示します。デバイスグループは、デフォルト値が変更されても、デフォルト値を使用するように構成されます。

```
cldevicegroup set -p numsecondaries= dg-schost-1
cldevicegroup show -v dg-schost-1

=== Device Groups ===

Device Group Name: dg-schost-1
Type: SVM
failback: yes
Node List: phys-schost-1, phys-schost-2 phys-schost-3
preferenced: yes
numsecondaries: 1
diskset names: dg-schost-1
```

## ▼ デバイスグループ構成の一覧を表示する方法

構成の一覧を表示するために、スーパーユーザーになる必要はありません。ただし、`solaris.cluster.read` の権限は必要です。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- Oracle Solaris Cluster Manager GUI 詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

`cldevicegroup show devicegroup` `cldevicegroup show devicegroup` を使用して、1つのデバイスグループの構成を一覧表示します。

`cldevicegroup status +` `cldevicegroup status +` を使用して、クラスタ内のすべてのデバイスグループのステータスを判別します。

**例 5-28** すべてのデバイスグループのステータスの一覧表示

```

--- Device Group Status ---

```

### 例 5-29 特定のデバイスグループの構成の一覧表示

Oracle Solaris Cluster システム管理 • 2013 年 3 月、E39396-02



```

failback: yes
Node List: phys-schost-2, phys-schost-3
preferenced: yes
numsecondaries: 1
diskset names: dg-schost-1

```

## ▼ デバイスグループのプライマリノードを切り替える

次の手順は、アクティブでないデバイスグループを起動する(オンラインにする)ときにも使用できます。

Oracle Solaris Cluster Manager GUI を使用して、アクティブでないデバイスグループをオンラインにしたり、デバイスグループのプライマリを切り替えたりすることもできます。詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.modify` を提供するプロファイルを使用します。
- 2 `cldevicegroup switch` を使用して、デバイスグループのプライマリノードを切り替えます。

```
cldevicegroup switch -n nodename devicegroup
```

`-n nodename` 切り替え先のノードの名前を指定します。このノードが新しいプライマリノードになります。

`devicegroup` 切り替えるデバイスグループを指定します。

- 3 デバイスグループが新しいプライマリノードに切り替わったことを確認します。  
デバイスグループが適切に登録されている場合、次のコマンドを使用すると、新しいデバイスグループの情報が表示されます。

```
cldevice status devicegroup
```

例 5-30 デバイスグループのプライマリノードの切り替え

次に、デバイスグループのプライマリノードを切り替えて変更結果を確認する例を示します。

```
cldevicegroup switch -n phys-schost-1 dg-schost-1

cldevicegroup status dg-schost-1

=== Cluster Device Groups ===

--- Device Group Status ---

Device Group Name Primary Secondary Status

dg-schost-1 phys-schost-1 phys-schost-2 Online
```

▼ デバイスグループを保守状態にする方法

デバイスグループを保守状態にすることによって、デバイスのいずれかにアクセスされたときに、デバイスグループが自動的にオンラインになることを防ぎます。デバイスグループを保守状態にするべきなのは、修理手順において、修理が終わるまで、すべての入出力活動を停止する必要がある場合などです。また、デバイスグループを保守状態にすることによって、別のノード上のディスクセットまたはディスクグループを修復していても、当該ノード上のデバイスグループはオンラインにならないため、データの損失を防ぎます。

破損したディスクセットを復元する方法については、[277 ページの「破損したディスクセットの復元」](#)を参照してください。

注-デバイスグループを保守状態にする前に、そのデバイスへのすべてのアクセスを停止し、依存するすべてのファイルシステムをマウント解除する必要があります。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 デバイスグループを保守状態にします。
  - a. デバイスグループが有効である場合は、デバイスグループを無効にします。

```
cldevicegroup disable devicegroup
```

b. デバイスグループをオフラインにします。

```
cldevicegroup offline devicegroup
```

- 2 修理手順を実行するときに、ディスクセットまたはディスクグループの所有権が必要な場合は、ディスクセットまたはディスクグループを手動でインポートします。

Solaris Volume Manager の場合:

```
metaset -C take -f -s diskset
```



注意 - Solaris Volume Manager ディスクセットの所有権を取得する場合、デバイスグループが保守状態にあるときは、`metaset -C take` コマンドを使用する必要があります。`metaset -t` を使用すると、所有権の取得作業の一部として、デバイスグループがオンラインになります。

- 3 必要な修理手順を実行します。
- 4 ディスクセットまたはディスクグループの所有権を解放します。



注意 - デバイスグループを保守状態から戻す前に、ディスクセットまたはディスクグループの所有権を解放する必要があります。所有権を解放しないと、データが失われる可能性があります。

Solaris Volume Manager の場合:

```
metaset -C release -s diskset
```

- 5 デバイスグループをオンラインにします。

```
cldevicegroup online devicegroup
```

```
cldevicegroup enable devicegroup
```

### 例 5-31 デバイスグループを保守状態にする

次に、デバイスグループ `dg-schost-1` を保守状態にし、保守状態からデバイスグループを削除する方法の例を示します。

```
[Place the device group in maintenance state.]
cldevicegroup disable dg-schost-1
cldevicegroup offline dg-schost-1
[If needed, manually import the disk set or disk group.]
For Solaris Volume Manager:
metaset -C take -f -s dg-schost-1
[Complete all necessary repair procedures.]

[Release ownership.]
For Solaris Volume Manager:
metaset -C release -s dg-schost-1
```

```
[Bring the device group online.]
cldevicegroup online dg-schost-1
cldevicegroup enable dg-schost-1
```

## ストレージデバイス用の **SCSI** プロトコル設定の管理

Oracle Solaris Cluster ソフトウェアをインストールすると、自動的に、すべてのストレージデバイスに SCSI リザーベーションが割り当てられます。次の手順に従って、複数のデバイスの設定を確認し、必要に応じてデバイスの設定をオーバーライドします。

- 148 ページの「すべてのストレージデバイスのデフォルトのグローバルな SCSI プロトコル設定を表示する方法」
- 149 ページの「単一ストレージデバイスの SCSI プロトコルを表示する方法」
- 150 ページの「すべてのストレージデバイスのデフォルトのグローバルなフェンシングプロトコル設定を変更する方法」
- 152 ページの「単一ストレージデバイスのフェンシングプロトコルを変更する方法」

### ▼ すべてのストレージデバイスのデフォルトのグローバルな **SCSI** プロトコル設定を表示する方法

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 スーパーユーザーになるか、RBAC の承認 **solaris.cluster.read** を提供する役割になります。
- 2 任意のノードから、現在のグローバルなデフォルト **SCSI** プロトコル設定を表示します。

```
cluster show -t global
```

詳細は、[cluster\(1CL\)](#) のマニュアルページを参照してください。

### 例 5-32 すべてのストレージデバイスのデフォルトのグローバルな SCSI プロトコル設定の表示

次の例に、クラスタ上のすべてのストレージデバイスの SCSI プロトコル設定を示します。

```
cluster show -t global

=== Cluster ===

Cluster Name: racerxx
installmode: disabled
heartbeat_timeout: 10000
heartbeat_quantum: 1000
private_netaddr: 172.16.0.0
private_netmask: 255.255.248.0
max_nodes: 64
max_privatenets: 10
global_fencing: prefer3
Node List: phys-racerxx-1, phys-racerxx-2
```

## ▼ 単一ストレージデバイスの SCSI プロトコルを表示する方法

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 スーパーユーザーになるか、RBAC の承認 `solaris.cluster.read` を提供する役割になります。
- 2 任意のノードから、ストレージデバイスの SCSI プロトコル設定を表示します。

```
cldevice show device
```

*device* デバイスパスの名前またはデバイス名。

詳細は、`cldevice(1CL)` のマニュアルページを参照してください。

### 例 5-33 単一デバイスの SCSI プロトコルの表示

次の例に、デバイス `/dev/rdisk/c4t8d0` の SCSI プロトコルを示します。

```
cldevice show /dev/rdisk/c4t8d0
```

```
=== DID Device Instances ===
```

```
DID Device Name: /dev/did/rdsk/d3
Full Device Path: phappy1:/dev/rdsk/c4t8d0
Full Device Path: phappy2:/dev/rdsk/c4t8d0
Replication: none
default_fencing: global
```

## ▼ すべてのストレージデバイスのデフォルトのグローバルなフェンシングプロトコル設定を変更する方法

フェンシングは、クラスタに接続されているすべてのストレージデバイスに対して、グローバルにオンまたはオフに設定できます。1つのストレージデバイスのデフォルトのフェンシングが `pathcount`、`prefer3`、または `nofencing` に設定されている場合、そのデバイスのデフォルトのフェンシング設定は、グローバル設定をオーバーライドします。ストレージデバイスのデフォルトのフェンシング設定が `global` に設定されている場合、ストレージデバイスはグローバル設定を使用します。たとえば、ストレージデバイスのデフォルト設定が `pathcount` である場合、ここでの手順を使用してグローバルな SCSI プロトコル設定を `prefer3` に変更しても、設定は変更されません。単一デバイスのデフォルト設定を変更するには、[152 ページ](#) の「[単一ストレージデバイスのフェンシングプロトコルを変更する方法](#)」の手順を使用します。



注意 - フェンシングを誤ってオフに設定すると、アプリケーションのフェイルオーバー時にデータの破損する可能性が生じやすくなります。フェンシングをオフに設定する場合は、そのような状況でもデータが破損しないかどうか十分に検査してください。共有ストレージデバイスが SCSI プロトコルをサポートしていない場合や、クラスタのストレージへのアクセスをクラスタ外のホストに対して許可する場合は、フェンシングをオフに設定できます。

定足数デバイスのデフォルトのフェンシング設定を変更するには、デバイスの構成を解除し、フェンシング設定を変更して、定足数デバイスを再構成します。フェンシングをオフに構成したあとで、定足数デバイスを含むデバイスについては定期的にオンに戻す場合は、定足数サーバーサービスを利用して定足数を構成することを確認してください(そうすることで、定足数の動作を中断せずに済みます)。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 スーパーユーザーになるか、RBAC 承認 **solaris.cluster.modify** を提供する役割になります。
- 2 定足数デバイスではないすべてのストレージデバイスのフェンシングプロトコルを設定します。

```
cluster set -p global_fencing={pathcount | prefer3 | nofencing | nofencing-noscrub}
```

|                                |                                                                                                                                                 |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-p global_fencing</code> | すべての共有デバイスの現在のグローバルなデフォルトフェンシングアルゴリズムを設定します。                                                                                                    |
| <code>prefer3</code>           | パスが2より多いデバイスに対して SCSI-3 プロトコルを使用します。                                                                                                            |
| <code>pathcount</code>         | 共有デバイスに接続されている DID パスの数でフェンシングプロトコルを決定します。pathcount 設定は、定足数デバイスで使用されます。                                                                         |
| <code>nofencing</code>         | フェンシングをオフに設定します (すべてのストレージデバイスについてフェンシングステータスを設定します)。                                                                                           |
| <code>nofencing-noscrub</code> | ディスク消し込みにより、持続的なすべての SCSI 予約情報からデバイスが消去され、クラスタの外側にあるシステムからストレージへのアクセスが可能になります。nofencing-noscrub オプションは、SCSI 予約に重大な問題があるストレージデバイスに対してのみ使用してください。 |

#### 例 5-34 すべてのストレージデバイスのデフォルトのグローバルなフェンシングプロトコル設定の設定

次の例では、クラスタ上のすべてのストレージデバイスのフェンシングプロトコルを、SCSI-3 プロトコルに設定します。

```
cluster set -p global_fencing=prefer3
```

## ▼ 単一ストレージデバイスのフェンシングプロトコルを変更する方法

フェンシングプロトコルは、1つのストレージデバイスに対して設定することもできます。

---

注- 定足数デバイスのデフォルトのフェンシング設定を変更するには、デバイスの構成を解除し、フェンシング設定を変更して、定足数デバイスを再構成します。フェンシングをオフに構成したあとで、定足数デバイスを含むデバイスについては定期的にオンに戻す場合は、定足数サーバーサービスを利用して定足数を構成することを検討してください (そうすることで、定足数の動作を中断せずに済みます)。

---

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。




---

注意- フェンシングを誤ってオフに設定すると、アプリケーションのフェイルオーバー時にデータの破損する可能性が生じやすくなります。フェンシングをオフに設定する場合は、そのような状況でもデータが破損しないかどうか十分に検査してください。共有ストレージデバイスが SCSI プロトコルをサポートしていない場合や、クラスタのストレージへのアクセスをクラスタ外のホストに対して許可する場合は、フェンシングをオフに設定できます。

---

- 1 スーパーユーザーになるか、RBAC 承認 `solaris.cluster.modify` を提供する役割になります。
- 2 ストレージデバイスのフェンシングプロトコルを設定します。

```
cldevice set -p default_fencing ={pathcount | \
scsi3 | global | nofencing | nofencing-noscrub} device
```

-p default\_fencing      デバイスの default\_fencing プロパティを変更します。

pathcount                共有デバイスに接続されている DID パスの数でフェンシングプロトコルを決定します。

scsi3                     SCSI-3 プロトコルを使用します。



|                   |                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 大域 (global)       | グローバルなデフォルトのフェンシング設定を使用します。global 設定は、定足数デバイス以外のデバイスで使用されます。<br><br>指定された DID インスタンスのフェンシングステータスを設定することで、フェンシングをオフに設定します。                           |
| nofencing-noscrub | ディスク消し込みにより、持続的なすべての SCSI 予約情報からデバイスが消去され、クラスタの外側にあるシステムからストレージデバイスへのアクセスが可能になります。nofencing-noscrub オプションは、SCSI 予約に重大な問題があるストレージデバイスに対してのみ使用してください。 |
| device            | デバイスパスの名前またはデバイス名を指定します。<br><br>詳細は、 <a href="#">cluster(1CL)</a> のマニュアルページを参照してください。                                                               |

例 5-35 単一デバイスのフェンシングプロトコルの設定

次の例では、(デバイス番号で指定される) デバイス d5 を SCSI-3 プロトコルに設定します。

```
cldevice set -p default_fencing=prefer3 d5
```

次の例では、d11 デバイスのデフォルトフェンシングをオフに設定します。

```
#cldevice set -p default_fencing=nofencing d11
```

# クラスタファイルシステムの管理

クラスタファイルシステムは、クラスタのどのノードからでも読み取りやアクセスが可能なグローバルなファイルシステムです。

表 5-5 タスクリスト:クラスタファイルシステムの管理

| タスク                                                   | 手順                                            |
|-------------------------------------------------------|-----------------------------------------------|
| クラスタファイルシステムを Oracle Solaris Cluster の最初のインストール後に追加する | <a href="#">154 ページの「クラスタファイルシステムを追加する方法」</a> |
| クラスタファイルシステムを削除する                                     | <a href="#">157 ページの「クラスタファイルシステムを削除する方法」</a> |

表 5-5 タスクリスト:クラスタファイルシステムの管理 (続き)

| タスク                                                 | 手順                                                 |
|-----------------------------------------------------|----------------------------------------------------|
| クラスタ内のグローバルマウントポイントをチェックして、ノード間の一貫性が保たれているかどうかを確認する | 159 ページの「 <a href="#">クラスタ内のグローバルマウントを確認する方法</a> 」 |

## ▼ クラスタファイルシステムを追加する方法

次のタスクは、Oracle Solaris Cluster の初期インストール後に作成するクラスタファイルシステムごとに実行します。



注意-必ず、正しいディスクデバイス名を指定してください。クラスタファイルシステムを作成すると、ディスク上のデータはすべて消去されます。デバイス名を誤って指定すると、本来消去する必要のないデータを失うことになります。

クラスタファイルシステムを追加する前に、次の必要条件が満たされていることを確認します。

- クラスタ内のノード上でスーパーユーザー特権が確立されていること。
- ボリュームマネージャーソフトウェアがクラスタ上にインストールおよび構成されています。
- クラスタファイルシステムの作成先のデバイスグループ (Solaris Volume Manager デバイスグループ) またはブロックディスクスライスが存在すること。

Oracle Solaris Cluster Manager を使用してデータサービスをインストールした場合は、クラスタファイルシステムがすでに自動的に作成されています (十分な共有ディスクが存在する場合)。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタ内にある任意のノード上でスーパーユーザーになります。  
クラスタ内に非大域ゾーンが構成されている場合は、大域ゾーンからこの手順を実行します。

ヒント-ファイルシステムを迅速に作成するには、ファイルシステムを作成するグローバルデバイスの現在のプライマリでスーパーユーザーになります。

## 2 ファイルシステムを作成する。



**Caution** - ファイルシステムを作成するとき、ディスク上のデータは破壊されます。必ず、正しいディスクデバイス名を指定してください。間違ったデバイス名を指定した場合、削除するつもりのないデータが削除されてしまいます。

- UFS ファイルシステムの場合は、**newfs(1M)** コマンドを使用します。

```
phys-schost# newfs raw-disk-device
```

次の表に、引数 *raw-disk-device* の名前の例を挙げます。命名規約はボリューム管理ソフトウェアごとに異なるので注意してください。

| ボリューム管理ソフトウェア          | ディスクデバイス名の例            | 説明                            |
|------------------------|------------------------|-------------------------------|
| Solaris Volume Manager | /dev/md/nfs/rdisk/d1   | nfs ディスクセット内の raw ディスクデバイス d1 |
| なし                     | /dev/global/rdisk/d1s3 | raw ディスクデバイス d1s3             |

## 3 クラスタ内の各ノードで、クラスタファイルシステムのマウントポイントのディレクトリを作成します。

そのノードからはクラスタファイルシステムにアクセスしない場合でも、マウントポイントはノードごとに必要です。

ヒント-管理を容易にするには、マウントポイントを `/global/device-group/` ディレクトリに作成します。この場所を使用すると、グローバルに利用できるクラスタファイルシステムとローカルファイルシステムを区別しやすくなります。

```
phys-schost# mkdir -p /global/device-group/mountpoint/
```

*device-group*      デバイスが含まれるデバイスグループ名に対応するディレクトリ名を指定します。

*mountpoint*      クラスタファイルシステムのマウント先のディレクトリ名を指定します。

## 4 クラスタ内の各ノードで、マウントポイント用の `/etc/vfstab` ファイルにエントリを追加します。

詳細は、[vfstab\(4\)](#) のマニュアルページを参照してください。

---

注- クラスタ内に非大域ゾーンが構成されている場合は、大域ゾーンのルートディレクトリのパスに大域ゾーンのクラスタファイルシステムをマウントするようにします。

---

- a. 各エントリで、使用する種類のファイルシステムに必要なマウントオプションを指定します。
- b. クラスタファイルシステムを自動的にマウントするには、**mount at boot** フィールドを **yes** に設定します。
- c. 各クラスタファイルシステムで、**/etc/vfstab** エントリの情報が各ノードで同じになるようにします。
- d. 各ノードの **/etc/vfstab** ファイルのエントリに、デバイスが同じ順序で表示されることを確認します。
- e. ファイルシステムのブート順の依存関係を検査します。

たとえば、phys-schost-1 がディスクデバイス d0 を **/global/oracle/** にマウントし、phys-schost-2 がディスクデバイス d1 を **/global/oracle/logs/** にマウントするとします。この構成では、phys-schost-1 がブートされ、**/global/oracle/** がマウントされたあとにのみ、phys-schost-2 をブートし、**/global/oracle/logs/** をマウントできます。

- 5 クラスタの任意のノード上で、構成確認ユーティリティを実行します。

```
phys-schost# cluster check -k vfstab
```

構成確認ユーティリティは、マウントポイントが存在することを確認します。また、**/etc/vfstab** ファイルのエントリが、クラスタのすべてのノードで正しいことを確認します。エラーが発生していない場合は、何も戻されません。

詳細は、[cluster\(1CL\)](#) のマニュアルページを参照してください。

- 6 クラスタファイルシステムをマウントします。

UFS および QFS の場合、クラスタ内の任意のノードからクラスタファイルシステムをマウントします。

```
phys-schost# mount /global/device-group/mountpoint/
```

- 7 クラスタ内にある各ノード上で、クラスタファイルシステムがマウントされていることを確認します。

**df** コマンドまたは **mount** コマンドのいずれかを使用し、マウントされたファイルシステムの一覧を表示します。詳細は、[df\(1M\)](#) のマニュアルページまたは [mount\(1M\)](#) のマニュアルページを参照してください。

クラスタファイルシステムには大域ゾーンと非大域ゾーンの両方からアクセスできます。

### 例 5-36 UFS クラスタファイルシステムの作成

次に、Solaris Volume Manager ボリューム /dev/md/oracle/rdsk/d1 上に、UFS クラスタファイルシステムを作成する例を示します。各ノードの `vfstab` ファイルにクラスタファイルシステムのエントリが追加されます。次に、1つのノードから `cluster check` コマンドを実行します。構成確認プロセスが正しく終了すると、1つのノードからクラスタファイルシステムがマウントされ、全ノードで確認されます。

```
phys-schost# newfs /dev/md/oracle/rdsk/d1
...
phys-schost# mkdir -p /global/oracle/d1
phys-schost# vi /etc/vfstab
#device device mount FS fsck mount mount
#to mount to fsck point type pass at boot options
#
/dev/md/oracle/dsk/d1 /dev/md/oracle/rdsk/d1 /global/oracle/d1 ufs 2 yes global,logging
...
phys-schost# cluster check -k vfstab
phys-schost# mount /global/oracle/d1
phys-schost# mount
...
/global/oracle/d1 on /dev/md/oracle/dsk/d1 read/write/setuid/global/logging/largefiles
on Sun Oct 3 08:56:16 2005
```

## ▼ クラスタファイルシステムを削除する方法

クラスタファイルシステムを削除するには、単に、そのクラスタファイルシステムのマウントを解除します。データも削除する場合は、配下のディスクデバイス (またはメタデバイスかボリューム) をシステムから削除します。

---

注 - クラスタファイルシステムは、`cluster shutdown` を実行してクラスタ全体を停止したときに、システム停止処理の一環として自動的にマウント解除されます。`shutdown` を実行して単独でノードを停止したときはマウント解除されません。なお、停止するノードが、ディスクに接続されている唯一のノードの場合は、そのディスク上のクラスタファイルシステムにアクセスしようとするエラーが発生します。

---

クラスタファイルシステムをマウント解除する前に、次の必要条件が満たされていることを確認します。

- クラスタ内のノード上でスーパーユーザー特権が確立されていること。

- ファイルシステムが使用中でないこと。ファイルシステムが使用中と見なされるのは、ユーザーがファイルシステム内のディレクトリにアクセスしている場合や、プログラムがファイルシステム内のファイルを開いている場合です。ユーザーやプログラムは、クラスタ内のどのノードでもアクセスできます。

1 クラスタ内にある任意のノード上でスーパーユーザーになります。

2 マウントされているクラスタファイルシステムを確認します。

```
mount -v
```

3 各ノードで、クラスタファイルシステムを使用中の全プロセスの一覧を表示し、停止するプロセスを判断します。

```
fuser -c [-u] mountpoint
```

-c                      ファイルシステムのマウントポイントとなっているファイルと、マウントされているファイルシステム内のファイルがすべて表示されます。

-u                      (任意) 各プロセス ID のユーザーログイン名を表示します。

mountpoint            プロセスを停止するクラスタファイルシステムの名前を指定します。

4 各ノードで、クラスタファイルシステムのプロセスをすべて停止します。  
プロセスは任意の方法で停止できます。必要であれば、次のコマンドを使用して、クラスタファイルシステムに関するプロセスを強制終了してください。

```
fuser -c -k mountpoint
```

クラスタファイルシステムを使用している各ノードに SIGKILL が送信されます。

5 各ノードで、ファイルシステムを使用しているプロセスがないことを確認します。

```
fuser -c mountpoint
```

6 1つのノードからファイルシステムをマウント解除します。

```
umount mountpoint
```

mountpoint            マウント解除するクラスタファイルシステムの名前を指定します。クラスタファイルシステムがマウントされているディレクトリの名前や、ファイルシステムのデバイス名パスを指定できます。

7 (任意)/etc/vfstab ファイルを編集して、削除するクラスタファイルシステムのエントリを削除します。

この手順は、/etc/vfstab ファイルにこのクラスタファイルシステムのエントリがある各クラスタノードで実行してください。

- 8 (任意) ディスクデバイス **group/metadevice/volume/plex** を削除します。  
詳細については、ボリューム管理ソフトウェアのドキュメントを参照してください。

### 例 5-37 クラスタファイルシステムの削除

次に、Solaris Volume Manager メタデバイスまたはボリューム `/dev/md/oracle/rdsk/d1` にマウントされた UFS クラスタファイルシステムを削除する例を示します。

```
mount -v
...
/global/oracle/d1 on /dev/md/oracle/dsk/d1 read/write/setuid/global/logging/largefiles
fuser -c /global/oracle/d1
/global/oracle/d1: 4006c
fuser -c -k /global/oracle/d1
/global/oracle/d1: 4006c
fuser -c /global/oracle/d1
/global/oracle/d1:
umount /global/oracle/d1

(On each node, remove the highlighted entry:)
vi /etc/vfstab
#device device mount FS fsck mount mount
#to mount to fsck point type pass at boot options
#
/global/oracle/dsk/d1 /dev/md/oracle/rdsk/d1 /global/oracle/d1 ufs 2 yes global,logging

[Save and exit.]
```

クラスタファイルシステム上のデータを削除するには、配下のデバイスを削除します。詳細については、ボリューム管理ソフトウェアのドキュメントを参照してください。

## ▼ クラスタ内のグローバルマウントを確認する方法

**cluster(1CL)** ユーティリティーは `/etc/vfstab` ファイル内の、クラスタファイルシステムに対するエントリの構文を検証します。エラーが発生していない場合は、何も戻されません。

注-クラスタファイルシステムの削除など、デバイスやボリューム管理コンポーネントに影響を及ぼすような変更をクラスタ構成に加えたあとに `cluster check` コマンドを実行します。

- 1 クラスタ内にある任意のノード上でスーパーユーザーになります。

2 クラスタのグローバルマウントを確認します。

```
cluster check -k vfstab
```

# ディスクパス監視の管理

ディスクパス監視 (DPM) の管理コマンドを使用すれば、セカンダリディスクパス障害の通知を受け取ることができます。このセクションでは、ディスクパスの監視に必要な管理タスクを行うための手順を説明します。ディスクパス監視デーモンに関する概念については、『[Oracle Solaris Cluster Concepts Guide](#)』の第3章「[Key Concepts for System Administrators and Application Developers](#)」を参照してください。コマンドオプションと関連するコマンドについては、[cldevice\(1CL\)](#)のマニュアルページを参照してください。scdpmd デーモンの調整に関する詳細は、[scdpmd.conf\(4\)](#)のマニュアルページを参照してください。デーモンが報告するログエラーに関しては、[syslogd\(1M\)](#)のマニュアルページも参照してください。

注-cldevice コマンドを使ってノードに入出力デバイスを追加すると、監視を行っていた監視リストにディスクパスが自動的に追加されます。Oracle Solaris Cluster コマンドを使ってノードからデバイスを削除すると、ディスクパスは自動的に監視から除外されます。

表 5-6 タスクマップ:ディスクパス監視の管理

| タスク                                                | 手順                                                                                                                                               |
|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| ディスクパスを監視します。                                      | <a href="#">161 ページの「ディスクパスを監視する方法」</a>                                                                                                          |
| ディスクパスの監視を解除します。                                   | <a href="#">162 ページの「ディスクパスの監視を解除する方法」</a>                                                                                                       |
| あるノードに対する障害のあるディスクパスのステータスを表示します。                  | <a href="#">163 ページの「障害のあるディスクパスを表示する方法」</a>                                                                                                     |
| ファイルからディスクパスを監視します。                                | <a href="#">165 ページの「ファイルからディスクパスを監視する方法」</a>                                                                                                    |
| 監視しているすべての共有ディスクパスが失敗したときのノードの自動リブートを有効化または無効化します。 | <a href="#">167 ページの「監視しているすべての共有ディスクパスが失敗したときのノードの自動リブートを有効にする方法」</a><br><br><a href="#">168 ページの「すべての監視共有ディスクパスが失敗した場合にノードの自動リブートを無効にする方法」</a> |



表 5-6 タスクマップ:ディスクパス監視の管理 (続き)

| タスク                                                                                                           | 手順                               |
|---------------------------------------------------------------------------------------------------------------|----------------------------------|
| 不正なディスクパスステータスを解決します。ブート時に監視対象の DID デバイスを利用できず、DID インスタンスが DID ドライバにアップロードされない場合、不正なディスクパスステータスが報告されることがあります。 | 164 ページの「ディスクパスのステータスエラーを解決する方法」 |

cldevice コマンドを実行する以下のセクションの手順にはディスクパス引数が含まれます。ディスクパス引数はノード名とディスク名からなります。ただし、ノード名は必須ではありません。指定しないと、all が使用されます。

## ▼ ディスクパスを監視する方法

このタスクは、クラスタのディスクパスを監視するときに行います。



注意 - DPM は、Sun Cluster 3.1 10/03 ソフトウェアより前にリリースされたバージョンを実行するノードではサポートされません。ローリングアップグレードの進行中には DPM コマンドを使用しないでください。すべてのノードがアップグレードされたあとで、DPM コマンドを使用するためにノードをオンラインにする必要があります。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 **solaris.cluster.modify** を提供する役割になります。
- 2 ディスクパスを監視します。  
`# cldevice monitor -n node disk`
- 3 ディスクパスが監視されているか確認します。  
`# cldevice status device`

**例 5-38 単一ノードのディスクパスを監視**

次の例では、単一ノードから `schost-1:/dev/did/rdisk/d1` ディスクパスを監視します。ディスク `/dev/did/dsk/d1` へのパスを監視するのは、ノード `schost-1` 上の DPM デーモンだけです。

```
cldevice monitor -n schost-1 /dev/did/dsk/d1
cldevice status d1
```

| Device Instance   | Node          | Status |
|-------------------|---------------|--------|
| -----             |               |        |
| /dev/did/rdisk/d1 | phys-schost-1 | Ok     |

**例 5-39 すべてのノードのディスクパスを監視**

次の例では、すべてのノードから `schost-1:/dev/did/dsk/d1` ディスクパスを監視します。DPM は、`/dev/did/dsk/d1` が有効なパスであるすべてのノードで起動されます。

```
cldevice monitor /dev/did/dsk/d1
cldevice status /dev/did/dsk/d1
```

| Device Instance   | Node          | Status |
|-------------------|---------------|--------|
| -----             |               |        |
| /dev/did/rdisk/d1 | phys-schost-1 | Ok     |

**例 5-40 CCR からディスク構成を読み直す**

次の例では、デーモンが CCR からディスク構成を読み直し、監視されているディスクパスをそのステータスとともに出力します。

```
cldevice monitor +
cldevice status
```

| Device Instance   | Node     | Status |
|-------------------|----------|--------|
| -----             |          |        |
| /dev/did/rdisk/d1 | schost-1 | Ok     |
| /dev/did/rdisk/d2 | schost-1 | Ok     |
| /dev/did/rdisk/d3 | schost-1 | Ok     |
|                   | schost-2 | Ok     |
| /dev/did/rdisk/d4 | schost-1 | Ok     |
|                   | schost-2 | Ok     |
| /dev/did/rdisk/d5 | schost-1 | Ok     |
|                   | schost-2 | Ok     |
| /dev/did/rdisk/d6 | schost-1 | Ok     |
|                   | schost-2 | Ok     |
| /dev/did/rdisk/d7 | schost-2 | Ok     |
| /dev/did/rdisk/d8 | schost-2 | Ok     |

**▼ ディスクパスの監視を解除する方法**

ディスクパスの監視を解除する場合は、この手順を使用します。



注意 - DPM は、Sun Cluster 3.1 10/03 ソフトウェアより前にリリースされたバージョンを実行するノードではサポートされません。ローリングアップグレードの進行中には DPM コマンドを使用しないでください。すべてのノードがアップグレードされたあとで、DPM コマンドを使用するためにノードをオンラインにする必要があります。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.modify` を提供する役割になります。
- 2 監視を解除するディスクパスの状態を調べます。  
`# cldevice status device`
- 3 各ノードで、適切なディスクパスの監視を解除します。  
`# cldevice unmonitor -n node disk`

#### 例 5-41 ディスクパスの監視解除

次の例では、`schost-2:/dev/did/rdisk/d1` ディスクパスの監視を解除し、クラスタ全体のディスクパスの一覧とそのステータスを出力します。

```
cldevice unmonitor -n schost2 /dev/did/rdisk/d1
cldevice status -n schost2 /dev/did/rdisk/d1
```

| Device Instance   | Node     | Status      |
|-------------------|----------|-------------|
| -----             | ----     | -----       |
| /dev/did/rdisk/d1 | schost-2 | Unmonitored |

## ▼ 障害のあるディスクパスを表示する方法

クラスタに障害のあるディスクパスを表示する場合は、次の手順を使用します。



注意 - DPM は、Sun Cluster 3.1 10/03 ソフトウェアより前にリリースされたバージョンを実行するノードではサポートされません。ローリングアップグレードの進行中には DPM コマンドを使用しないでください。すべてのノードがアップグレードされたあとで、DPM コマンドを使用するためにノードをオンラインにする必要があります。

- 1 クラスタ内にある任意のノード上でスーパーユーザーになります。
- 2 全クラスタ内の障害のあるディスクパスを表示します。

```
cldevice status -s fail
```

#### 例 5-42 障害のあるディスクパスを表示する

次の例では、全クラスタ内の障害のあるディスクパスを表示します。

```
cldevice status -s fail
```

| Device Instance | Node          | Status |
|-----------------|---------------|--------|
| -----           | ----          | -----  |
| dev/did/dsk/d4  | phys-schost-1 | fail   |

## ▼ ディスクパスのステータスエラーを解決する方法

次のイベントが発生すると、DPM が障害の発生したパスがオンラインになっても、そのパスのステータスを更新しない可能性があります。

- 監視対象パスの障害によって、ノードがリブートする。
- リブートしたノードがオンラインに戻るまで、監視対象の DID パスの下のデバイスがオンラインに戻らない。

ブート時に監視対象の DID デバイスを利用できず、このため DID インスタンスが DID ドライバにアップロードされないため、不正なディスクパスステータスが報告されます。このような状態が発生する場合は、手動で DID 情報を更新します。

- 1 1つのノードからグローバルデバイス名前空間を更新します。
- 2 次の手順に進む前に、各ノードでコマンド処理が完了していることを確認します。  
このコマンドは、1つのノードからのみ実行されても、リモートからすべてのノードで実行されます。コマンドが処理を終了したかどうかを確認するには、クラスタの各ノードで次のコマンドを実行します。

```
ps -ef | grep cldevice populate
```

- 3 DPM ポーリングタイムフレーム内で障害の発生したディスクパスのステータスが **OK** になっていることを確認します。

```
cldevice status disk-device
```

| Device Instance | Node          | Status |
|-----------------|---------------|--------|
| -----           | ----          | -----  |
| dev/did/dsk/dN  | phys-schost-1 | Ok     |

## ▼ ファイルからディスクパスを監視する方法

ファイルを使ってディスクパスを監視したり、その監視を解除する場合は、次の手順を使用します。

ファイルを使用してクラスタ構成を変更するには、まず現在の構成をエクスポートします。このエクスポート操作により XML ファイルが作成されます。このファイルは、変更する構成項目を設定するために修正できます。この手順では、このプロセス全体を説明します。



注意 - DPM は、Sun Cluster 3.1 10/03 ソフトウェアより前にリリースされたバージョンを実行するノードではサポートされません。ローリングアップグレードの進行中には DPM コマンドを使用しないでください。すべてのノードがアップグレードされたあとで、DPM コマンドを使用するためにノードをオンラインにする必要があります。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 **solaris.cluster.modify** を提供する役割になります。
- 2 デバイス構成を XML ファイルにエクスポートします。  

```
cldevice export -o configurationfile
```

 -o configurationfile XML ファイルのファイル名を指定します。
- 3 デバイスパスが監視されるよう、構成ファイルを変更します。  
 監視するデバイスパスを検索し、monitored 属性を true に設定します。
- 4 デバイスパスを監視します。  

```
cldevice monitor -i configurationfile
```

`-i configurationfile` 変更された XML ファイルのファイル名を指定します。

- 5 この時点でデバイスパスが監視されていることを確認します。

`# cldevice status`

#### 例 5-43 ファイルからディスクパスを監視する

次の例では、ノード `phys-schost-2` とデバイス `d3` の間のデバイスパスが、XML ファイルを使用することによって監視されています。

最初に、現在のクラスタ構成をエクスポートします。

`# cldevice export -o deviceconfig`

`deviceconfig` XML ファイルは、`phys-schost-2` と `d3` の間のパスが現在は監視されていないことを示しています。

```
<?xml version="1.0"?>
<!DOCTYPE cluster SYSTEM "/usr/cluster/lib/xml/cluster.dtd">
<cluster name="brave_clus">
.
.
.
 <deviceList readonly="true">
 <device name="d3" ctd="c1t8d0">
 <devicePath nodeRef="phys-schost-1" monitored="true"/>
 <devicePath nodeRef="phys-schost-2" monitored="false"/>
 </device>
 </deviceList>
</cluster>
```

そのパスを監視するには、次のように、監視される attribute を `true` に設定します。

```
<?xml version="1.0"?>
<!DOCTYPE cluster SYSTEM "/usr/cluster/lib/xml/cluster.dtd">
<cluster name="brave_clus">
.
.
.
 <deviceList readonly="true">
 <device name="d3" ctd="c1t8d0">
 <devicePath nodeRef="phys-schost-1" monitored="true"/>
 <devicePath nodeRef="phys-schost-2" monitored="true"/>
 </device>
 </deviceList>
</cluster>
```

`cldevice` コマンドを使用して、ファイルを読み込み、監視を有効にします。

`# cldevice monitor -i deviceconfig`

`cldevice` コマンドを使用して、この時点でデバイスが監視されていることを確認します。

```
cldevice status
```

参照 クラスタ構成のエクスポート、および結果のXMLファイルを使用したクラスタ構成の設定の詳細は、[cluster\(1CL\)](#) および [clconfiguration\(5CL\)](#) のマニュアルページを参照してください。

## ▼ 監視しているすべての共有ディスクパスが失敗したときのノードの自動リブートを有効にする方法

この機能を有効にすると、次の条件が満たされる場合、ノードは自動的にリブートします。

- ノード上ですべての監視対象の共有ディスクパスが失敗した。
- 監視対象のディスクのうち、少なくとも1つに、クラスタ内の異なるノードからアクセスできる。`scdpm` デーモンは、プライベートインターコネクトを使用して、ディスクがクラスタ内の別のノードからアクセス可能かどうかをチェックします。プライベートインターコネクトが無効な場合、`scdpm` デーモンは別のノードからディスクのステータスを取得できません。

ノードが再起動すると、そのノード上でマスターされているすべてのリソースグループとデバイスグループが別のノード上で再起動します。

ノードが自動リブートしたあと、ノード上のすべての監視対象共有ディスクパスがアクセス不能のままである場合、そのノードは再び自動リブートしません。しかし、ノードがリブートしたが失敗したあとに、利用可能になったディスクパスがある場合、そのノードは再び自動リブートします。

`reboot_on_path_failure` プロパティを有効にすると、ローカルディスクパスの状態は、ノードのリブートが必要かどうか決定するときには考慮されません。監視された共有ディスクのみが影響を受けます。

- 1 クラスタ内の任意のノードで、スーパーユーザーになるか、`solaris.cluster.modify RBAC` の承認を提供する役割になります。
- 2 クラスタのすべてのノードに対して、監視共有ディスクパスがすべて失敗したときの、ノードの自動リブートを有効にします。

```
clnode set -p reboot_on_path_failure=enabled +
```

## ▼ すべての監視共有ディスクパスが失敗した場合に ノードの自動リブートを無効にする方法

この機能を無効にすると、あるノード上のすべての監視共有ディスクパスに障害が発生しても、ノードは自動的にリブートしません。

- 1 クラスタ内の任意のノードで、スーパーユーザーになるか、**solaris.cluster.modify RBAC**の承認を提供する役割になります。
- 2 クラスタ内のすべてのノードに対して、ノードへのすべての監視共有ディスクパスに障害が発生した場合の、ノードの自動リブートを無効にします。

```
clnode set -p reboot_on_path_failure=disabled +
```



## 定足数の管理

---

この章では、Oracle Solaris Cluster および Oracle Solaris Cluster 定足数サーバー内の定足数デバイスの管理手順について説明します。定足数の概念については、『[Oracle Solaris Cluster Concepts Guide](#)』の「[Quorum and Quorum Devices](#)」を参照してください。

- 169 ページの「定足数デバイスの管理」
- 193 ページの「Oracle Solaris Cluster 定足数サーバーの管理」

### 定足数デバイスの管理

定足数デバイスとは、複数のノードによって共有される共有ストレージデバイスまたは定足数サーバーで、定足数を確立するために使用される票を構成します。このセクションでは、定足数デバイスを管理するための手順について説明します。

`clquorum(1CL)` コマンドを使用すると、定足数デバイスの管理手順をすべて実行できます。また、`clsetup(1CL)` 対話型ユーティリティまたは Oracle Solaris Cluster Manager GUI を使用して、いくつかの手順を行えます。このセクションの管理手順は、可能なかぎり `clsetup` ユーティリティを使用して説明してあります。Oracle Solaris Cluster Manager オンラインヘルプでは、GUI を使用して定足数の手順を実行する方法を説明しています。定足数デバイスを使用して作業する際は、次のガイドラインに注意してください。

- 定足数コマンドはすべて、グローバルクラスタ投票ノードで実行する必要があります。
- `clquorum` コマンドが中断または失敗すると、定足数の構成情報は、クラスタ構成データベースで矛盾することになります。このような矛盾が発生した場合は、このコマンドを再度実行するか、`clquorum reset` コマンドを実行して定足数構成をリセットします。

- クラスタの可用性を最高にするには、定足数デバイスによる合計の投票数が、ノードによる合計の投票数よりも少なくなるようにします。少なくなければ、すべてのノードが機能していても、すべての定足数デバイスを使用できない場合、そのノードはクラスタを形成できません。
- 現在定足数デバイスとして構成されているディスクは、Solaris ZFS ストレージプールには追加しないでください。構成済みの定足数デバイスを Solaris ZFS ストレージプールに追加すると、ディスクは EFI ディスクとしてラベルが変更され、また定足数構成情報が失われ、ディスクはクラスタへの定足数投票を提供しなくなります。ディスクがストレージプール内に入ると、そのディスクは定足数デバイスとして構成できます。または、ディスクの定足数デバイス構成を解除し、ディスクをストレージプールに追加した後に、そのディスクを定足数デバイスとして再構成することができます。

注-clsetup コマンドは、ほかの Oracle Solaris Cluster コマンドに対する対話型インタフェースです。clsetup の実行時、このコマンドは適切な固有のコマンドを生成します。今回の場合は、clquorum コマンドです。これらのコマンドは、各説明の後にある例の中で示しています。

定足数構成を表示するには、clquorum show を使用します。clquorum list コマンドは、クラスタ内の定足数デバイスの名前を表示します。clquorum status コマンドは、ステータスと投票数の情報を提供します。

このセクションで示す例は、主に 3 ノードクラスタです。

表 6-1 タスクリスト:定足数の管理

タスク	説明
clsetup ユーティリティを使用し て、クラスタに定足数デバイスを追加 する	<a href="#">172 ページの「定足数デバイスの追加」</a>
clsetup を使用して (clquorum を生 成)、クラスタから定足数デバイスを 削除する	<a href="#">181 ページの「定足数デバイスを削除する方法」</a>
clsetup を使用して (clquorum を生 成)、クラスタから最後の定足数デバ イスを削除する	<a href="#">183 ページの「クラスタから最後の定足数デバイスを削除 する方法」</a>
追加と削除の手順を使用すること で、クラスタ内の定足数デバイスを交 換する	<a href="#">184 ページの「定足数デバイスを交換する方法」</a>
追加と削除の手順を使用すること で、定足数デバイスのリストを変更す る	<a href="#">185 ページの「定足数デバイスのノードリストを変更する 方法」</a>

表 6-1 タスクリスト:定足数の管理 (続き)

タスク	説明
clsetup を使用して (clquorum を生成)、定足数デバイスを保守状態にする  (保守状態にある場合、定足数デバイスは定足数確立の投票に参加しません。)	<a href="#">187 ページの「定足数デバイスを保守状態にする方法」</a>
clsetup ユーティリティを使用して (clquorum を生成)、定足数構成をデフォルト状態にリセットする	<a href="#">189 ページの「定足数デバイスを保守状態から戻す」</a>
clquorum コマンドを使用して、定足数デバイスと投票数を一覧表示する	<a href="#">190 ページの「クラスタ構成を一覧表示する方法」</a>

## 定足数デバイスへの動的再構成

クラスタ内の定足数デバイス上で動的再構成 (DR) を実行するときには、いくつかの問題を考慮する必要があります。

- Oracle Solaris の DR 機能の説明で述べられているすべての必要条件、手順、制限は、Oracle Solaris Cluster の DR サポートにも適用されます (オペレーティングシステムでの休止操作は除く)。したがって、Oracle Solaris Cluster ソフトウェアで DR 機能を使用する前に、Oracle Solaris の DR 機能についての説明を参照してください。特に、DR の切り離し操作中に、ネットワークに接続されていない入出力デバイスに影響する問題について確認してください。
- Oracle Solaris Cluster は、定足数デバイス用に構成されたインタフェースが存在する場合 DR 削除操作を実行できません。
- DR 操作がアクティブなデバイスに影響する場合、Oracle Solaris Cluster はその操作を拒否して、その操作から影響を受けるデバイスを識別します。

定足数デバイスを削除するには、次の手順をその順番どおりに行います。

表 6-2 タスクマップ:定足数デバイスへの動的再構成

タスク	説明
1. 削除する定足数デバイスと交換する、新しい定足数デバイスを有効に設定	<a href="#">172 ページの「定足数デバイスの追加」</a>
2. 削除する定足数デバイスを無効に設定	<a href="#">181 ページの「定足数デバイスを削除する方法」</a>

表 6-2 タスクマップ: 定足数デバイスへの動的再構成 (続き)

タスク	説明
3. 削除する定足数デバイス上で DR 削除操作を実行	

## 定足数デバイスの追加

このセクションでは、定足数デバイスを追加する手順について説明します。クラスタのすべてのノードがオンラインか確認してから、新しい定足数デバイスを追加します。クラスタに必要な定足数投票数の決定、推奨される定足数構成、および障害フェンシングについては、『[Oracle Solaris Cluster Concepts Guide](#)』の「[Quorum and Quorum Devices](#)」を参照してください。



注意 - 現在定足数デバイスとして構成されているディスクは、Solaris ZFS ストレージプールには追加しないでください。構成済みの定足数デバイスを Solaris ZFS ストレージプールに追加すると、ディスクは EFI ディスクとしてラベルが変更され、また定足数構成情報が失われ、ディスクはクラスタへの定足数投票を提供しなくなります。ディスクがストレージプール内に入ると、そのディスクは定足数デバイスとして構成できます。ディスクの定足数デバイス構成を解除し、ディスクをストレージプールに追加したあとに、そのディスクを定足数デバイスとして再構成することもできます。

Oracle Solaris Cluster ソフトウェアは、次の種類の定足数デバイスをサポートしています。

- 以下の共有 LUN
  - 共有 SCSI ディスク
  - Serial Attached Technology Attachment (SATA) ストレージ
  - Sun NAS
  - Oracle 社製 Sun ZFS Storage Appliance
- Oracle Solaris Cluster Quorum Server

これらのデバイスを追加する方法については、次のセクションで説明しています。

[173 ページの「共有ディスク定足数デバイスを追加する方法」](#)

- [177 ページの「定足数サーバー定足数をデバイスとして追加する方法」](#)

注 - 複製されたディスクを定足数デバイスとして構成することはできません。複製されたディスクを定足数デバイスとして追加しようとする、次のエラーメッセージが表示され、コマンドはエラーコードとともに終了します。

---

*Disk-name* is a replicated device. Replicated devices cannot be configured as quorum devices.

---

共有ディスク定足数デバイスは、Oracle Solaris Cluster ソフトウェアがサポートする任意の接続済みストレージデバイスです。共有ディスクは、クラスタの複数のノードに接続されます。フェンシングをオンに構成すると、デュアルポートのディスクを定足数デバイスとして構成して、SCSI-2 または SCSI-3 (デフォルトは SCSI-2) を使用できます。フェンシングがオンに構成され、共有デバイスが3つ以上のノードに接続されている場合は、SCSI-3 プロトコル (2 ノードを超える場合のデフォルトのプロトコル) を使用する定足数デバイスとして共有ディスクを構成できます。SCSI オーバーライドフラグを使用すると、デュアルポートの共有ディスクで SCSI-3 プロトコルを使用するように Oracle Solaris Cluster ソフトウェアに対して指示できます。

共有ディスクのフェンシングをオフに構成した場合は、ソフトウェア定足数プロトコルを使用する定足数デバイスとしてディスクを構成できます。これは、そのディスクが SCSI-2 と SCSI-3 のどちらのプロトコルをサポートしている場合でも有効です。ソフトウェアの定足数は、SCSI Persistent Group Reservations (PGR) のフォームをエミュレートする、Oracle のプロトコルです。



---

注意 - 使用するディスクが SCSI (SATA など) をサポートしていない場合は、SCSI フェンシングをオフにするようにしてください。

---

定足数デバイスには、ユーザーデータを格納しているディスクまたはデバイスグループのメンバーになっているディスクを使用できます。共有ディスクがある定足数サブシステムで使用されているプロトコルは、`cluster show` コマンドの出力の、共有ディスクの `access-mode` 値で確認します。

これらの手順は、Oracle Solaris Cluster Manager GUI を使用しても実行できます。詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

次の手順で使用されるコマンドについては、`clsetup(1CL)` および `clquorum(1CL)` のマニュアルページを参照してください。

## ▼ 共有ディスク定足数デバイスを追加する方法

Oracle Solaris Cluster ソフトウェアでは、共有ディスク (SCSI と SATA の両方) デバイスを定足数デバイスとして使用できます。SATA デバイスは SCSI 予約をサポートしていないため、その種類のディスクを定足数デバイスとして構成するには、SCSI 予約フェンシングフラグをオフに構成し、ソフトウェア定足数プロトコルを使用します。

この手順を実行するには、ノードが共有するデバイス ID (Device Identifier、DID) によりディスクドライブを確認します。`cldevice show` コマンドを使用して、DID 名のー

覧を参照します。詳細は、[cldevice\(1CL\)](#)のマニュアルページを参照してください。クラスタのすべてのノードがオンラインか確認してから、新しい定足数デバイスを追加します。

次の手順を実行して、SCSI または SATA デバイスを構成します。

**phys-schost#** プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 **solaris.cluster.modify** を提供する役割になります。
- 2 **clsetup** ユーティリティを起動します。  
**# clsetup**  
**clsetup** のメインメニューが表示されます。
- 3 定足数のオプションに対応する番号を入力します。  
「定足数メニュー」が表示されます。
- 4 定足数デバイスを追加するためのオプションに対応する番号を入力し、**clsetup** ユーティリティから追加する定足数デバイスの確認を求められたら、**yes** と入力します。  
追加する定足数デバイスの種類を確認するメッセージが表示されます。
- 5 共有ディスク定足数デバイスのオプションに対応する番号を入力します。  
どのグローバルデバイスを使用するかを確認するメッセージが表示されます。
- 6 使用しているグローバルデバイスを入力します。  
指定したグローバルデバイスに新しい定足数デバイスを追加するか確認を求めるメッセージが表示されます。
- 7 「**yes**」と入力し、新しい定足数デバイスの追加を続行します。  
新しい定足数デバイスが正常に追加されると、**clsetup** ユーティリティではその旨のメッセージが表示されます。
- 8 定足数デバイスが追加されていることを確認します。

**# clquorum list -v**

## 例6-1 共有ディスク定足数デバイスの追加

次の例は、共有ディスク定足数デバイスを追加する際に `clsetup` によって生成される `clquorum` コマンドと、検証ステップを示しています。

```
Become superuser or assume a role that provides solaris.cluster.modify RBAC authorization on any
cluster node.
```

```
[Start the clsetup utility:]
clsetup
[Select Quorum>Add a quorum device]
[Answer the questions when prompted.]
[You will need the following information.]
 [Information: Example:]
 [Directly attached shared disk shared_disk]
 [Global device d20]
```

```
[Verify that the clquorum command was completed successfully:]
clquorum add d20
```

```
 Command completed successfully.
[Quit the clsetup Quorum Menu and Main Menu.]
[Verify that the quorum device is added:]
clquorum list -v
```

Quorum	Type
-----	----
d20	shared_disk
scphyshost-1	node
scphyshost-2	node

### ▼ Sun NAS または Sun ZFS Storage Appliance NAS の定足数デバイスを追加する方法

クラスタのすべてのノードがオンラインか確認してから、新しい定足数デバイスを追加します。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 **Sun NAS の GUI を使用して、Sun NAS ファイラで iSCSI デバイスを設定します。iSCSI デバイスのセットアップ手順については、Sun ZFS Storage Appliance に付属のインストールドキュメントまたはアプライアンスのオンラインヘルプを参照してください。**

Sun NAS デバイスがある場合は、次のコマンドを使用します。

- a. サイズが約 **50M** バイトのファイルボリュームを作成します。



- b. 各ノードの iSCSI アクセスリストを作成します。
  - i. クラスタの名前を、iSCSI アクセスリスト名として使用します。
  - ii. アクセスリストに各クラスタノードのイニシエータノード名を追加します。CHAP および IQN は不要です。
- c. iSCSI LUN を構成します  
 バッキングファイルボリュームの名前を LUN の名前として使用できます。各ノードのアクセスリストを LUN に追加します。
- 2 各クラスタノードで、iSCSI LUN を検出して、iSCSI アクセスリストを静的構成に設定します。
 

```
iscsiadm modify discovery -s enable

iscsiadm list discovery
Discovery:
 Static: enabled
 Send Targets: disabled
 iSNS: disabled

iscsiadm add static-config iqn.LUNName,IPAddress_of_NASDevice
devfsadm -i iscsi
cldevice refresh
```
- 3 1つのクラスタノードから DID を iSCSI LUN 用に構成します。
 

```
/usr/cluster/bin/cldevice populate
```
- 4 iSCSI を使用するクラスタに構成した NAS デバイス LUN を表す DID デバイスを特定します。cldevice show コマンドを使用して、DID 名の一覧を参照します。詳細は、cldevice(1CL)のマニュアルページを参照してください。
- 5 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 solaris.cluster.modify を提供する役割になります。
- 6 clquorum コマンドを使用して、手順 4 で特定した DID デバイスを使用する定足数デバイスとして NAS デバイスを追加します。
 

```
clquorum add d20
```

クラスタには、scsi-2、scsi-3、またはソフトウェア定足数プロトコルのどれを使用するかを判断するためのデフォルトのルールがあります。詳細については、clquorum(1CL)を参照してください。

## 例 6-2 Sun NAS または Sun ZFS Storage Appliance NAS の定足数デバイスの追加

次の例は、Sun NAS 定足数デバイスを追加する際に clsetup によって生成される clquorum コマンドと、検証段階を示しています。iSCSI デバイスのセットアップ手順



については、Sun ZFS Storage Appliance に付属のインストールドキュメントまたはアプライアンスのオンラインヘルプを参照してください。

```
Add an iSCSI device on the Sun NAS filer.
Use the Sun NAS GUI to create a file volume that is approximately 50mb in size.
File Volume Operations -> Create File Volume
For each node, create an iSCSI access list.
iSCSI Configuration -> Configure Access List
Add the initiator node name of each cluster node to the access list.
*** Need GUI or command syntax for this step. ***
Configure the iSCSI LUN
iSCSI Configuration -> Configure iSCSI LUN
On each of the cluster nodes, discover the iSCSI LUN and set the iSCSI access list to static configuration.
iscsiadm modify discovery -s enable
iscsiadm list discovery
Discovery:
 Static: enabled
 Send Targets: enabled
 iSNS: disabled
iscsiadm add static-config
iqn.1986-03.com.sun0-1:000e0c66efe8.4604DE16.thinqorum,10.11.160.20
devsadm -i iscsi
From one cluster node, configure the DID devices for the iSCSI LUN.
/usr/cluster/bin/scldevice populate
/usr/cluster/bin/scldevice populate
Become superuser or assume a role that provides solaris.cluster.modify RBAC authorization on any
cluster node.

[Add the NAS device as a quorum device
using the DID device:]
clquorum add d20

Command completed successfully.
```

## ▼ 定足数サーバー定足数をデバイスとして追加する方法

始める前に Oracle Solaris Cluster 定足数サーバーを定足数デバイスとして追加するには、Oracle Solaris Cluster 定足数サーバーソフトウェアがホストマシンにインストールされていて、定足数サーバーが起動および実行されている必要があります。定足数サーバーのインストールについては、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』の「[Quorum Server ソフトウェアをインストールおよび構成する方法](#)」を参照してください。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 **solaris.cluster.modify** を提供する役割になります。

- 2 すべての **Oracle Solaris Cluster** ノードがオンライン状態であり、**Oracle Solaris Cluster Quorum Server** と通信が行えることを確認します。
  - a. クラスタノードに直接接続されているネットワークスイッチが次の基準のいずれかを満たすことを確認します。
    - スイッチは RSTP (Rapid Spanning Tree Protocol) をサポートしています。
    - スイッチ上で高速ポートモードが有効になっています。

クラスタノードと定足数サーバー間ですぐに通信できるようにするには、これらの機能の1つが必要です。この通信がスイッチによって大幅に遅延すると、クラスタはこの通信の中断を定足数デバイスが失われたものと解釈します。
  - b. パブリックネットワークで可変長サブネット化 (**CIDR (Classless Inter-Domain Routing)** と呼ばれる) を使用している場合は、各ノードで次のファイルを変更します。
 

クラスフルサブネットを使用する場合は、これらの手順を実行する必要はありません。

    - i. **/etc/inet/netmasks** ファイルにクラスタが使用する各パブリックサブネットのエントリを追加します。
 

パブリックネットワークの IP アドレスとネットマスクを含むエントリの例を次に示します。

```
10.11.30.0 255.255.255.0
```
    - ii. それぞれの **/etc/hostname.adapter** ファイルに **netmask + broadcast +** を追加します。
 

```
nodename netmask + broadcast +
```
  - c. クラスタ内の各ノード上で、定足数サーバーのホスト名を **/etc/inet/hosts** ファイルまたは **/etc/inet/ipnodes** ファイルに追加します。
 

次のように、ホスト名とアドレスのマッピングをファイルに追加します。

```
ipaddress qshost1
ipaddress 定足数サーバーが実行中であるコンピュータの IP アドレス。
qshost1 定足数サーバーが実行中であるコンピュータのホスト名。
```
  - d. ネームサービスを使用する場合、定足数サーバーホストの名前とアドレスの対応付けをネームサービスデータベースに追加します。
- 3 **clsetup** ユーティリティを起動します。
 

```
clsetup
```

**clsetup** のメインメニューが表示されます。

- 4 定足数のオプションに対応する番号を入力します。  
「定足数メニュー」が表示されます。
  - 5 定足数デバイスを追加するためのオプションに対応する番号を入力します。定足数デバイスを追加することを確認するには、「**yes**」と入力します。  
追加する定足数デバイスの種類を確認するメッセージが表示されます。
  - 6 定足数サーバー定足数デバイスのオプションに対応する番号を入力します。**quorum\_server** 定足数デバイスを追加することを確認するには、「**yes**」と入力します。  
`clsetup` ユーティリティにより、新しい定足数デバイスの名前の入力を求められます。
  - 7 追加する定足数デバイスの名前を入力します。  
定足数デバイスの名前は任意に選択できます。この名前は、今後の管理コマンドの処理だけに使用されるものです。  
定足数サーバーのホスト名を入力するように、`clsetup` ユーティリティのプロンプトが表示されます。
  - 8 定足数サーバーのホストの名前を入力します。  
この名前で、定足数サーバーが動作するマシンの IP アドレス、またはネットワーク上のマシンのホスト名を指定します。  
ホストの IPv4 または IPv6 構成に応じて、マシンの IP アドレスを `/etc/hosts` ファイル、`/etc/inet/ipnodes` ファイル、またはその両方で指定します。
- 
- 注-指定したマシンはすべてのクラスタノードから到達可能で、定足数サーバーをマシン上で実行してある必要があります。
- 
- `clsetup` ユーティリティは、定足数サーバーのポート番号を入力するようメッセージを表示します。
- 9 クラスタノードとやり取りする際に定足数サーバーが使用するポート番号を入力します。  
新しい定足数デバイスを追加するか確認を求めるメッセージが表示されます。
  - 10 「**yes**」と入力し、新しい定足数デバイスの追加を続行します。  
新しい定足数デバイスが正常に追加されると、`clsetup` ユーティリティではその旨のメッセージが表示されます。

## 11 定足数デバイスが追加されていることを確認します。

```
clquorum list -v
```

### 例6-3 定足数サーバー定足数デバイスの追加

次の例は、定足数サーバー定足数デバイスを追加する際に `clsetup` によって生成される `clquorum` コマンドを示しています。またこの例では検証ステップも示します。

Become superuser or assume a role that provides solaris.cluster.modify RBAC authorization on any cluster node.

[Start the clsetup utility:]

```
clsetup
```

[Select Quorum > Add a quorum device]

[Answer the questions when prompted.]

[You will need the following information.]

[Information:	Example:]
[Quorum Device	quorum_server quorum device]
[Name:	qdl]
[Host Machine Name:	10.11.124.84]
[Port Number:	9001]

[Verify that the clquorum command was completed successfully:]

```
clquorum add -t quorum_server -p qshost=10.11.124.84,-p port=9001 qdl
```

Command completed successfully.

[Quit the clsetup Quorum Menu and Main Menu.]

[Verify that the quorum device is added:]

```
clquorum list -v
```

Quorum	Type
-----	----
qdl	quorum_server
scphyshost-1	node
scphyshost-2	node

```
clquorum status
```

```
=== Cluster Quorum ===
```

```
-- Quorum Votes Summary --
```

Needed	Present	Possible
-----	-----	-----
3	5	5

```
-- Quorum Votes by Node --
```

Node Name	Present	Possible	Status
-----	-----	-----	-----
phys-schost-1	1	1	Online
phys-schost-2	1	1	Online

```
-- Quorum Votes by Device --
```

Device Name	Present	Possible	Status
-------------	---------	----------	--------

-----	-----	-----	-----
qd1	1	1	Online
d3s2	1	1	Online
d4s2	1	1	Online

# 定足数デバイスの削除または交換

このセクションでは、定足数デバイスを削除または交換するための次の手順を説明します。

- [181 ページの「定足数デバイスを削除する方法」](#)
- [183 ページの「クラスタから最後の定足数デバイスを削除する方法」](#)
- [184 ページの「定足数デバイスを交換する方法」](#)

## ▼ 定足数デバイスを削除する方法

この手順は、Oracle Solaris Cluster Manager GUI を使用しても実行できます。詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

定足数デバイスが削除されると、その定足数デバイスは定足数を確立するための投票に参加しなくなります。2 ノードクラスタでは、定足数デバイスが少なくとも1つは構成されている必要があります。それがクラスタの最後の定足数デバイスの場合、`clquorum(1CL)` で、構成からそのデバイスを削除することはできません。ノードを削除する場合は、そのノードに接続されている定足数デバイスをすべて削除してください。

---

注- 削除するデバイスがクラスタの最後の定足数デバイスの場合は、[183 ページの「クラスタから最後の定足数デバイスを削除する方法」](#) の手順を参照してください。

---

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.modify` を提供する役割になります。
- 2 削除する定足数デバイスを判別します。

`# clquorum list -v`

- 3 **clsetup(1CL)** ユーティリティーを実行します。

```
clsetup
```

メインメニューが表示されます。

- 4 定足数のオプションに対応する番号を入力します。
- 5 定足数デバイスを削除するためのオプションに対応する番号を入力します。  
削除プロセス中に表示される質問に答えます。
- 6 **clsetup** を終了します。
- 7 定足数デバイスが削除されたことを確認します。

```
clquorum list -v
```

#### 例 6-4 定足数デバイスの削除

次に、2 つ以上の定足数デバイスが構成されているクラスタから定足数デバイスを削除する例を示します。

Become superuser or assume a role that provides solaris.cluster.modify RBAC authorization on any cluster node.

```
[Determine the quorum device to be removed:]
```

```
clquorum list -v
```

```
[Start the clsetup utility:]
```

```
clsetup
```

```
[Select Quorum>Remove a quorum device]
```

```
[Answer the questions when prompted.]
```

```
Quit the clsetup Quorum Menu and Main Menu.]
```

```
[Verify that the quorum device is removed:]
```

```
clquorum list -v
```

```
Quorum Type

scphyshost-1 node
scphyshost-2 node
scphyshost-3 node
```

**注意事項** 定足数サーバー定足数デバイスの削除中に、クラスタと定足数サーバーホストの間の通信が失われた場合、定足数サーバーホストに関する無効な構成情報をクリーンアップする必要があります。このクリーンアップの実行に関する説明は、[196 ページ](#)の「**期限切れの定足数サーバークラスタ情報のクリーンアップ**」を参照してください。

## ▼ クラスタから最後の定足数デバイスを削除する方法

この手順では、`clquorum force` オプション、`-F` を使用して、2 ノードクラスタから最後の定足数デバイスを削除します。通常、不具合が起きたデバイスをまず削除し、代替りの定足数デバイスを追加します。これが2 ノードクラスタの最後の定足数デバイスでない場合は、[181 ページ](#)の「定足数デバイスを削除する方法」の手順に従ってください。

定足数デバイスを追加する処理では、ノードが再構成されるため、障害のあった定足数デバイスに影響が及び、マシンでパニックが発生します。`F` (強制) オプションを使用すると、マシンでパニックを発生させることなく、障害があった定足数デバイスを削除できます。`clquorum` コマンドでは、構成からデバイスを削除できます。不具合が発生した定足数デバイスを削除したあと、`clquorum add` コマンドで新しいデバイスを追加することができます。[172 ページ](#)の「定足数デバイスの追加」と [clquorum\(1CL\)](#) のマニュアルページを参照してください。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.modify` を提供する役割になります。
- 2 `clquorum` コマンドを使用して定足数デバイスを削除します。定足数デバイスに障害が発生した場合は、`-F` (強制) オプションを使用して、障害が発生したデバイスを削除します。

```
clquorum remove -F qd1
```

---

注- また、削除するノードを保持状態とし、定足数デバイスを `clquorum removequorum` コマンドを使用して削除することができます。`clsetup(1CL)` クラスタ管理メニューオプションは、クラスタがインストールモードのときは使用できません。詳細については、[251 ページ](#)の「ノードを保守状態にする」を参照してください。

---

- 3 定足数デバイスが削除されたことを確認します。

```
clquorum list -v
```

### 例 6-5 最後の定足数デバイスの削除

この例では、クラスタを保持モードにし、クラスタ構成で最後の定足数デバイスを削除する方法を示しています。

```
[Become superuser or assume a role that provides solaris.cluster.modify RBAC authorization on any
cluster node.]
[Place the cluster in install mode:]
cluster set -p installmode=enabled
[Remove the quorum device:]
clquorum remove d3
[Verify that the quorum device has been removed:]
clquorum list -v
 Quorum Type

scphyshost-1 node
scphyshost-2 node
scphyshost-3 node
```

## ▼ 定足数デバイスを交換する方法

この作業は、既存の定足数デバイスをほかの定足数デバイスに交換する場合に行います。定足数デバイスは、類似したデバイスタイプに交換することも(例: NAS デバイスをほかの NAS デバイスに置き換える)、あるいは類似点がないデバイスに交換することも(例: NAS デバイスを共有ディスクに置き換える)こともできます。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

### 1 新しい定足数デバイスを構成します。

最初に、古いデバイスの代わりに、新しい定足数デバイスを構成に追加する必要があります。クラスタに新しい定足数デバイスを追加する方法は、[172 ページの「定足数デバイスの追加」](#)を参照してください。

### 2 定足数デバイスとして交換するデバイスを削除します。

構成から古い定足数デバイスを削除する方法は、[181 ページの「定足数デバイスを削除する方法」](#)を参照してください。

### 3 定足数デバイスが障害が発生したディスクである場合は、ディスクを取り替えます。

『[Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual](#)』で、ディスクエンクロージャーのハードウェア手順を参照してください。



## 定足数デバイスの保守

このセクションでは、定足数デバイスを保守するための次の手順を説明します。

- 185 ページの「定足数デバイスのノードリストを変更する方法」
- 187 ページの「定足数デバイスを保守状態にする方法」
- 189 ページの「定足数デバイスを保守状態から戻す」
- 190 ページの「クラスタ構成を一覧表示する方法」
- 191 ページの「定足数デバイスを修復する方法」
- 192 ページの「定足数のデフォルトのタイムアウトの変更」

### ▼ 定足数デバイスのノードリストを変更する方法

**clsetup(1CL)** ユーティリティーを使用すると、既存の定足数デバイスのノードリストにノードを追加したり、ノードリストからノードを削除したりできます。定足数デバイスのノードリストを変更するには、定足数デバイスを削除し、削除した定足数デバイスへのノードの物理的な接続を変更して、定足数デバイスをクラスタ構成に追加し直す必要があります。定足数デバイスが追加されると、**clquorum(1CL)** によって、そのディスクに接続されているすべてのノードに対して、ノードからディスクへのパスが自動的に構成されます。

**phys-schost#** プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 **solaris.cluster.modify** を提供する役割になります。
- 2 変更したい定足数デバイスの名前を判別します。  

```
clquorum list -v
```
- 3 **clsetup** ユーティリティーを起動します。  

```
clsetup
```

 メインメニューが表示されます。
- 4 定足数のオプションに対応する番号を入力します。  
 「定足数メニュー」が表示されます。
- 5 定足数デバイスを削除するためのオプションに対応する番号を入力します。  
 指示に従います。削除するディスクの名前を問い合わせられます。

- 6 定足数デバイスへのノード接続を追加または削除します。
- 7 定足数デバイスを追加するためのオプションに対応する番号を入力します。  
指示に従います。定足数デバイスとして使用するディスクの名前を問い合わせられます。
- 8 定足数デバイスが追加されていることを確認します。  
**# clquorum list -v**

## 例 6-6 定足数デバイスノードリストの変更

次の例に、**clsetup** ユーティリティを使用して、定足数デバイスのノードリストにノードを追加したり、ノードリストからノードを削除する方法を示します。この例では、定足数デバイスの名前は **d2** であり、この手順の最終目的は別のノードを定足数デバイスのノードリストに追加することです。

[Become superuser or assume a role that provides solaris.cluster.modify RBAC authorization on any node in the cluster.]

[Determine the quorum device name:]

```
clquorum list -v
Quorum Type

d2 shared_disk
sc-phys-schost-1 node
sc-phys-schost-2 node
sc-phys-schost-3 node
```

[Start the clsetup utility:]

```
clsetup
```

[Type the number that corresponds with the quorum option.]

```
.
```

[Type the number that corresponds with the option to remove a quorum device.]

```
.
```

[Answer the questions when prompted.]

[You will need the following information:]

Information:	Example:
Quorum Device Name:	d2

[Verify that the clquorum command completed successfully:]

```
clquorum remove d2
Command completed successfully.
```

[Verify that the quorum device was removed.]

```
clquorum list -v
Quorum Type

sc-phys-schost-1 node
sc-phys-schost-2 node
sc-phys-schost-3 node
```

```
[Type the number that corresponds with the Quorum option.]
.
[Type the number that corresponds with the option to add a quorum device.]
.
[Answer the questions when prompted.]
[You will need the following information:]

 Information Example:
 quorum device name d2

[Verify that the clquorum command was completed successfully:]
clquorum add d2
 Command completed successfully.

Quit the clsetup utility.

[Verify that the correct nodes have paths to the quorum device.
In this example, note that phys-schost-3 has been added to the
enabled hosts list.]
clquorum show d2 | grep Hosts
=== Quorum Devices ===

Quorum Device Name: d2
 Hosts (enabled): phys-schost-1, phys-schost-2, phys-schost-3

[Verify that the modified quorum device is online.]

clquorum status d2
=== Cluster Quorum ===

--- Quorum Votes by Device ---

Device Name Present Possible Status

d2 1 1 Online
```

## ▼ 定足数デバイスを保守状態にする方法

**clquorum(1CL)** コマンドを使用して定足数デバイスを保守状態にします。現在、**clsetup(1CL)** ユーティリティにこの機能はありません。この手順は、Oracle Solaris Cluster Manager GUI を使用しても実行できます。詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

サービスから定足数デバイスを長時間はずす場合は、その定足数デバイスを保守状態にします。定足数デバイスの定足数投票数 (quorum vote count) はゼロに設定されるため、そのデバイスが稼働中でも定足数確立の投票には参加しません。保守状態でも定足数デバイスの構成情報は保持されます。

---

注-2 ノードクラスタでは、定足数デバイスが少なくとも1つは構成されている必要があります。構成されているデバイスが2 ノードクラスタの最後の定足数デバイスの場合は、**clquorum** は失敗してデバイスは保守状態になりません。

---

クラスタノードを保守状態にする方法については、[251 ページ](#)の「ノードを保守状態にする」を参照してください。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.modify` を提供する役割になります。
- 2 定足数デバイスを保守状態にします。  
`# clquorum disable device`  
`device` 変更するディスクデバイスの DID 名 (d4 など) を指定します。
- 3 定足数デバイスが保守状態にあることを確認します。  
保守状態にしたデバイスの出力は、定足数デバイスの投票数 (以下の例の Quorum device votes) がゼロになっていなければなりません。  
`# clquorum status device`

例 6-7 定足数デバイスを保守状態にする

次に、定足数デバイスを保守状態にし、結果を検証する例を示します。

```
clquorum disable d20
clquorum status d20

=== Cluster Quorum ===

--- Quorum Votes by Device ---

Device Name Present Possible Status

d20 1 1 Offline
```

参照 定足数デバイスを有効にし直す方法については、[189 ページ](#)の「定足数デバイスを保守状態から戻す」を参照してください。

ノードを保守状態にする方法については、[251 ページ](#)の「ノードを保守状態にする」を参照してください。

## ▼ 定足数デバイスを保守状態から戻す

この作業は、定足数デバイスが保守状態にある場合にその状態から定足数デバイスを戻して定足数投票数をデフォルトにリセットするときに実行します。



注意 - `globaldev` または `node` オプションのどちらも指定しない場合、定足数投票数はクラスタ全体でリセットされます。

定足数デバイスを構成する場合、Oracle Solaris Cluster ソフトウェアは定足数デバイスに投票数として  $N-1$  を割り当てます ( $N$  は定足数デバイスに結合された投票の数)。たとえば、2 つのノードに接続された、投票数がゼロ以外の定足数デバイスの投票数は 1 (2 マイナス 1) になります。

- クラスタノードと、そのクラスタノードに関係付けられた定足数デバイスを保守状態から戻す方法については、[253 ページの「ノードを保守状態から戻す」](#)を参照してください。
- 定足数投票数の詳細は、『[Oracle Solaris Cluster Concepts Guide](#)』の「[About Quorum Vote Counts](#)」を参照してください。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.modify` を提供する役割になります。
- 2 定足数投票数をリセットします。  

```
clquorum enable device
```

`device` リセットする定足数デバイスの DID 名 (d4 など) を指定します。
- 3 ノードが保守状態にあったために定足数投票数をリセットする場合は、このノードをリブートします。
- 4 定足数投票数を確認します。  

```
clquorum show +
```

### 例 6-8 定足数投票数 (定足数デバイス) のリセット

次に、定足数デバイスの投票数をリセットしてデフォルト設定に戻し、結果を検証する例を示します。

```
clquorum enable d20
clquorum show +

=== Cluster Nodes ===

Node Name: phys-schost-2
Node ID: 1
Quorum Vote Count: 1
Reservation Key: 0x43BAC41300000001

Node Name: phys-schost-3
Node ID: 2
Quorum Vote Count: 1
Reservation Key: 0x43BAC41300000002

=== Quorum Devices ===

Quorum Device Name: d3
Enabled: yes
Votes: 1
Global Name: /dev/did/rdsd/d20s2
Type: shared_disk
Access Mode: scsi3
Hosts (enabled): phys-schost-2, phys-schost-3
```

## ▼ クラスタ構成を一覧表示する方法

この手順は、Oracle Solaris Cluster Manager GUI を使用しても実行できます。詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

定足数構成を一覧表示するには、スーパーユーザーになる必要はありません。RBAC の承認 `solaris.cluster.read` を提供する任意の役割になることができます。

---

注- 定足数デバイスに対するノード接続の数を増減させる場合、定足数が自動的に再計算されることはありません。すべての定足数デバイスをいったん削除し、その後それらを構成に追加し直すと、正しい定足数が再構成されます。2 ノードクラスタの場合、定足数デバイスを取り外して、もとの定足数デバイスに戻す前に一時的に新しい定足数デバイスを追加します。次に一時的に追加した定足数デバイスを取り外します。

---

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- **clquorum** コマンドを使用して、定足数の構成を一覧表示します。

```
% clquorum show +
```

### 例 6-9 定足数構成の一覧表示

```
% clquorum show +
```

```
=== Cluster Nodes ===
```

```
Node Name: phys-schost-2
Node ID: 1
Quorum Vote Count: 1
Reservation Key: 0x43BAC41300000001
```

```
Node Name: phys-schost-3
Node ID: 2
Quorum Vote Count: 1
Reservation Key: 0x43BAC41300000002
```

```
=== Quorum Devices ===
```

```
Quorum Device Name: d3
Enabled: yes
Votes: 1
Global Name: /dev/did/rdisk/d20s2
Type: shared_disk
Access Mode: scsi3
Hosts (enabled): phys-schost-2, phys-schost-3
```

## ▼ 定足数デバイスを修復する方法

この作業は、動作が不正な定足数デバイスを交換する場合に行なってください。

**phys-schost#** プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 定足数デバイスとして交換するディスクデバイスを削除します。

---

注- 削除するデバイスが最後の定足数デバイスである場合は、必要に応じて初めにほかのディスクを新しい定足数デバイスとして追加してください。この手順により、交換作業中に障害が発生した場合も定足数デバイスが有効になります。新しい定足数デバイスを追加する方法については、[172 ページの「定足数デバイスの追加」](#)を参照してください。

---

定足数デバイスとしてのディスクデバイスを削除する方法については、[181 ページの「定足数デバイスを削除する方法」](#)を参照してください。

- 2 ディスクデバイスを交換します。  
ディスクデバイスを交換する場合は、『[Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual](#)』で、ディスクエンクロージャーのハードウェア手順を参照してください。
- 3 交換したディスクを新しい定足数デバイスとして追加します。  
ディスクを新しい定足数デバイスとして追加する方法については、[172 ページの「定足数デバイスの追加」](#)を参照してください。

---

注-手順1で定足数デバイスを別途追加した場合は、デバイスを削除しても安全です。定足数デバイスを削除する方法については、[181 ページの「定足数デバイスを削除する方法」](#)を参照してください。

---

## 定足数のデフォルトのタイムアウトの変更

クラスタ再構成時の定足数の操作を完了するまでのタイムアウトは、デフォルトで25秒に構成されています。定足数タイムアウトの値は、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』の「[定足数デバイスを構成する方法](#)」の指示に従って増分することができます。タイムアウト値を増分する代わりに、別の定足数デバイスに切り替えるという方法もあります。

その他のトラブルシューティング情報については、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』の「[定足数デバイスを構成する方法](#)」を参照してください。

---

注- Oracle RAC (Oracle Real Application Clusters) では、デフォルトの定足数タイムアウトである25秒を変更しないでください。一部のスプリットブレインシナリオでは、タイムアウト時間を長くすると、VIPリソースのタイムアウトが原因でOracle RAC VIP フェイルオーバーが失敗する可能性があります。使用している定足数デバイスがデフォルトの25秒のタイムアウトに適合しない場合は、別の定足数デバイスを使用してください。

---



# Oracle Solaris Cluster 定足数サーバーの管理

Oracle Solaris Cluster Quorum Server は、共有ストレージデバイスではない、定足数デバイスを提供します。このセクションでは、Oracle Solaris Cluster 定足数サーバーを管理するための次のような手順について説明します。

- 193 ページの「[Quorum Server Software の起動および停止](#)」
- 193 ページの「[定足数サーバーを起動する方法](#)」
- 194 ページの「[定足数サーバーを停止する方法](#)」
- 195 ページの「[定足数サーバーに関する情報の表示](#)」
- 196 ページの「[期限切れの定足数サーバークラスタ情報のクリーンアップ](#)」

Oracle Solaris Cluster 定足数サーバーのインストールおよび構成については、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』の「[Quorum Server ソフトウェアをインストールおよび構成する方法](#)」を参照してください。

## Quorum Server Software の起動および停止

次の手順では、Oracle Solaris Cluster ソフトウェアを起動および停止する方法を説明します。

デフォルトでは、次の手順は、定足数サーバー構成ファイル `/etc/scqsd/scqsd.conf` の内容をカスタマイズしていない場合の、1つのデフォルト定足数サーバーを起動および停止します。デフォルトの定足数サーバーはポート 9000 上にバインドされ、定足数情報には `/var/scqsd` ディレクトリを使用します。

定足数サーバーソフトウェアのインストールについては、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』の「[Quorum Server ソフトウェアをインストールおよび構成する方法](#)」を参照してください。定足数タイムアウトの値を変更する方法については、192 ページの「[定足数のデフォルトのタイムアウトの変更](#)」を参照してください。

### ▼ 定足数サーバーを起動する方法

- 1 Oracle Solaris Cluster ソフトウェアを起動するホスト上でスーパーユーザーになります。
- 2 ソフトウェアを起動するには、**clquorumserver start** コマンドを使用します。

```
/usr/cluster/bin/clquorumserver start quorumserver
```

*quorumserver* 定足数サーバーを識別します。定足数サーバーが待機するポート番号を使用できます。構成ファイルでインスタンス名を指定した場合は、代わりにその名前を使用できます。

1 台の定足数サーバーを起動するには、インスタンス名とポート番号のいずれかを指定します。複数の定足数サーバーを構成している場合、すべての定足数サーバーを起動するには、+ オペランドを使用します。

#### 例 6-10 すべての構成済み定足数サーバーの起動

次の例では、構成されているすべての定足数サーバーを起動します。

```
/usr/cluster/bin/clquorumserver start +
```

#### 例 6-11 特定の定足数サーバーの起動

次の例では、ポート番号 2000 で待機している定足数サーバーを起動します。

```
/usr/cluster/bin/clquorumserver start 2000
```

## ▼ 定足数サーバーを停止する方法

- 1 Oracle Solaris Cluster ソフトウェアを起動するホスト上でスーパーユーザーになります。
- 2 ソフトウェアを停止するには、**clquorumserver stop** コマンドを使用します。

```
/usr/cluster/bin/clquorumserver stop [-d] quorumserver
```

*-d* マシンを次回起動したときに、定足数サーバーを起動するかどうかを制御します。*-d* オプションを指定すると、次のマシン起動時に定足数サーバーは起動しません。

*quorumserver* 定足数サーバーを識別します。定足数サーバーが待機するポート番号を使用できます。構成ファイルでインスタンス名を指定した場合は、代わりにその名前を使用できます。

1 台の定足数サーバーを停止するには、インスタンス名とポート番号のいずれかを指定します。複数の定足数サーバーを構成している場合、すべての定足数サーバーを停止するには、+ オペランドを使用します。

**例 6-12** すべての構成済み定足数サーバーの停止

次の例では、構成されているすべての定足数サーバーを停止します。

```
/usr/cluster/bin/clquorumserver stop +
```

**例 6-13** 特定の定足数サーバーの停止

次の例では、ポート番号 2000 で待機している定足数サーバーを停止します。

```
/usr/cluster/bin/clquorumserver stop 2000
```

## 定足数サーバーに関する情報の表示

定足数サーバーについての構成情報を表示することができます。このコマンドは、定足数サーバーを定足数デバイスとして構成しているすべてのクラスタごとに、対応するクラスタ名、クラスタ ID、予約鍵のリスト、および登録鍵のリストを表示します。

### ▼ 定足数サーバーに関する方法情報を表示する方法

- 1 定足数サーバーの情報を表示するホスト上でスーパーユーザーになります。スーパーユーザー以外のユーザーには、`solaris.cluster.read` RBAC (Role-Based Access Control) の承認が必要です。RBAC 権利プロファイルの詳細については、[rbac\(5\)](#) のマニュアルページを参照してください。
- 2 `clquorumserver` コマンドを使用することで、定足数サーバーの構成情報を表示します。

```
/usr/cluster/bin/clquorumserver show quorumserver
```

*quorumserver*      1 つまたは複数の定足数サーバーを識別します。インスタンス名またはポート番号で定足数サーバーを指定できます。すべての定足数サーバーの構成情報を表示するには、+ オペランドを使用します。

**例 6-14** 1 つの定足数サーバーの構成の表示

次の例では、ポート 9000 を使用する定足数サーバーの構成情報を表示します。次のコマンドは、定足数サーバーが定足数デバイスとして構成されているすべてのクラスタの情報を表示します。この情報にはクラスタの名前と ID、およびデバイスの予約鍵と登録鍵のリストが含まれます。

次の例では、クラスタ `bastille` の ID が 1、2、3、および 4 であるノードが、定足数サーバー上に鍵を登録しています。また、ノード 4 は定足数デバイスの予約を所有しているため、その鍵は予約リストに表示されます。

```
/usr/cluster/bin/clquorumserver show 9000

=== Quorum Server on port 9000 ===

--- Cluster bastille (id 0x439A2EFB) Reservation ---

Node ID: 4
Reservation key: 0x439a2efb00000004

--- Cluster bastille (id 0x439A2EFB) Registrations ---

Node ID: 1
Registration key: 0x439a2efb00000001

Node ID: 2
Registration key: 0x439a2efb00000002

Node ID: 3
Registration key: 0x439a2efb00000003

Node ID: 4
Registration key: 0x439a2efb00000004
```

#### 例 6-15 複数の定足数サーバーの構成の表示

次の例では、3 つの定足数サーバー `qs1`、`qs2`、および `qs3` の構成情報を表示します。

```
/usr/cluster/bin/clquorumserver show qs1 qs2 qs3
```

#### 例 6-16 動作しているすべての定足数サーバーの構成の表示

次の例では、動作しているすべての定足数サーバーの構成情報を表示します。

```
/usr/cluster/bin/clquorumserver show +
```

## 期限切れの定足数サーバークラスタ情報のク リーンアップ

`quorumserver` のタイプの定足数デバイスを削除するには、How to Remove a Quorum Device で説明されているように、[181 ページ](#)の「定足数デバイスを削除する方法」コマンドを使用します。通常の動作では、このコマンドは定足数サーバーホストに関

する定足数サーバーの情報も削除します。ただし、クラスタが定足数サーバーホストとの通信を失うと、定足数デバイスを削除しても、この情報がクリーンアップされません。

定足数サーバークラスタ情報は、次の状況で無効になります。

- `clquorum remove` コマンドを使用してクラスタ定足数デバイスを削除せずに、クラスタの運用を停止した場合。
- 定足数サーバーホストが停止している間に、`quorum_server` タイプの定足数デバイスをクラスタから削除した場合。



注意 - タイプ `quorumserver` の定足数デバイスがまだクラスタから削除されていない場合、この手順を使用して無効な定足数サーバーを削除すると、クラスタ定足数に障害が発生する可能性があります。

## ▼ 定足数サーバーの構成情報をクリーンアップする方法

始める前に 181 ページの「定足数デバイスを削除する方法」で説明されているとおりに、定足数サーバーの定足数デバイスを削除します。



注意 - クラスタがまだこの定足数サーバーを使用している場合、この手順を実行するとクラスタ定足数に障害が発生します。

- 1 定足数サーバーホストでスーパーユーザーになります。
- 2 **`clquorumserver clear`** コマンドを使用して、構成ファイルをクリーンアップします。  

```
clquorumserver clear -c clustername -I clusterID quorumserver [-y]
```

**`-c clustername`** 以前に定足数サーバーを定足数デバイスとして使用していたクラスタの名前です。

クラスタ名を取得するには、クラスタノード上で `cluster show` を実行します。

**`-I clusterID`** クラスタ ID です。

クラスタ ID は 8 桁の 16 進数です。クラスタ ID を取得するには、クラスタノード上で `cluster show` を実行します。

**`quorumserver`** 1 つまたは複数の定足数サーバーの識別子です。

定足数サーバーは、ポート番号かインスタンス名で識別できます。ポート番号は、クラスタノードが定足数サーバーと通信するために使用されます。インスタンス名は、定足数サーバーの構成ファイル `/etc/scqsd/scqsd.conf` で指定されます。

-y                    実行前に確認のプロンプトを表示することなく、`clquorumserver clear` コマンドに、構成ファイルからクラスタ情報をクリーンアップさせます。

期限切れのクラスタ情報を定足数サーバーから削除したいことが確かである場合のみ、このオプションを使用します。

- 3 (省略可能) このサーバーインスタンスでほかに定足数デバイスが構成されていない場合は、定足数サーバーを停止します。

#### 例 6-17 定足数サーバー構成からの期限切れのクラスタ情報のクリーンアップ

次の例は、`sc-cluster` という名前のクラスタについての情報を、ポート 9000 を使用する定足数サーバーから削除します。

```
clquorumserver clear -c sc-cluster -I 0x4308D2CF 9000
The quorum server to be unconfigured must have been removed from the cluster.
Unconfiguring a valid quorum server could compromise the cluster quorum. Do you
want to continue? (yes or no) y
```

# クラスタインターコネクトとパブリックネットワークの管理

---

この章では、Oracle Solaris Cluster インターコネクトとパブリックネットワークのソフトウェア上の作業手順について説明します。

クラスタインターコネクトとパブリックネットワークの管理には、ハードウェア上の作業とソフトウェア上の作業が含まれます。通常、初めてクラスタをインストールおよび構成するときには、IP ネットワークマルチパス (IP Network Multipathing) グループを含むクラスタインターコネクトとパブリックネットワークを構成します。マルチパスは Oracle Solaris 10 OS に自動的にインストールされますが、使用するには有効にする必要があります。あとで、クラスタインターコネクトネットワーク構成を変更する必要がある場合は、この章のソフトウェア手順を使用します。クラスタ内に IP Network Multipathing グループを構成する方法については、[215 ページの「パブリックネットワークの管理」](#)のセクションを参照してください。

この章では、次のトピックの手順について説明します。

- [200 ページの「クラスタインターコネクトの管理」](#)
- [215 ページの「パブリックネットワークの管理」](#)

この章の関連手順の詳細な説明については、[表 7-1](#) および [表 7-3](#) を参照してください。

クラスタインターコネクトおよびパブリックネットワークの背景情報および概要情報については、『[Oracle Solaris Cluster Concepts Guide](#)』というドキュメントを参照してください。

# クラスタインターコネクトの管理

ここでは、クラスタインターコネクト (クラスタトランスポートアダプタ、クラスタトランスポートケーブルなど) を再構成する手順を説明します。これらの手順では、Oracle Solaris Cluster ソフトウェアがインストールされている必要があります。

通常、`clsetup` ユーティリティを使用すると、クラスタインターコネクトのクラスタトランスポートを管理できます。詳細は、[clsetup\(1CL\)](#) のマニュアルページを参照してください。クラスタインターコネクトコマンドはすべて、グローバルクラスタ投票ノードで実行する必要があります。

クラスタソフトウェアをインストールする手順については、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』を参照してください。クラスタハードウェアコンポーネントをサービスする手順については、『[Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual](#)』を参照してください。

注-クラスタインターコネクト手順中、通常は、(適切であれば) デフォルトのポート名を選択してもかまいません。デフォルトのポート名は、ケーブルのアダプタ側が接続されているノードの内部ノード ID 番号と同じです。

表 7-1 タスクリスト:クラスタインターコネクトの管理

タスク	手順
<code>clsetup(1CL)</code> を使用することで、クラスタトランスポートを管理する	25 ページの「クラスタ構成ユーティリティにアクセスする方法」
<code>clinterconnect status</code> を使用することで、クラスタインターコネクトのステータスを確認する	202 ページの「クラスタインターコネクトのステータスを確認する方法」
<code>clsetup</code> を使用することで、クラスタトランスポートケーブル、トランスポートアダプタ、またはスイッチを追加する	203 ページの「クラスタトランスポートケーブル、トランスポートアダプタ、トランスポートスイッチを追加する方法」
<code>clsetup</code> を使用することで、クラスタトランスポートケーブル、トランスポートアダプタ、またはトランスポートスイッチを削除する	205 ページの「クラスタトランスポートケーブル、トランスポートアダプタ、トランスポートスイッチを削除する方法」
<code>clsetup</code> を使用することで、クラスタトランスポートケーブルを有効にする	208 ページの「クラスタトランスポートケーブルを有効にする方法」
<code>clsetup</code> を使用することで、クラスタトランスポートケーブルを無効にする	210 ページの「クラスタトランスポートケーブルを無効にする方法」
トランスポートアダプタのインスタンス番号の確認	212 ページの「トランスポートアダプタのインスタンス番号を確認する方法」



表 7-1 タスクリスト:クラスタインターコネクトの管理 (続き)

タスク	手順
IP アドレスまたは既存のクラスタのアドレス範囲の変更	212 ページの「既存のクラスタのプライベートネットワークアドレスまたはアドレス範囲を変更する方法」

## クラスタインターコネクトでの動的再構成

クラスタインターコネクト上で動的再構成 (DR) を実行するときには、いくつかの問題を考慮する必要があります。

- Oracle Solaris の DR 機能の説明で述べられているすべての必要条件、手順、制限は、Oracle Solaris Cluster の DR サポートにも適用されます (オペレーティングシステムでの休止操作は除く)。したがって、Oracle Solaris Cluster ソフトウェアで DR 機能を使用する前に、Oracle Solaris の DR 機能についての説明を参照してください。特に、DR の切り離し操作中に、ネットワークに接続されていない入出力デバイスに影響する問題について確認してください。
- Oracle Solaris Cluster ソフトウェアは、アクティブなプライベートインターコネクトインタフェース上で実行された DR ボード削除操作を拒否します。
- アクティブなクラスタインターコネクトで DR を実行するには、クラスタからアクティブなアダプタを完全に削除する必要があります。clsetup メニューまたは該当するコマンドを使用します。



注意 - Oracle Solaris Cluster ソフトウェアの個々のクラスタノードには、他のすべてのクラスタノードに対する有効なパスが、少なくとも 1 つは存在していなければなりません。したがって、個々のクラスタノードへの最後のパスをサポートするプライベートインターコネクトインタフェースを無効にしないでください。

パブリックネットワークインタフェース上で DR 操作を実行するときは、次の手順をその順番どおりに行います。

表 7-2 タスクマップ:パブリックネットワークインタフェースでの動的再構成

タスク	手順
1. アクティブなインターコネクトからインタフェースを無効にして削除	217 ページの「パブリックネットワークインタフェースでの動的再構成」
2. パブリックネットワークインタフェース上で DR 操作を実行	

▼ クラスタインターコネクトのステータスを確認する方法

この手順は、Oracle Solaris Cluster Manager GUI を使用しても実行できます。詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

この手順を実行するためにスーパーユーザーとしてログインする必要はありません。

- 1 クラスタインターコネクトのステータスを確認します。  
% clinterconnect status
- 2 一般的なステータスメッセージについては、以下の表を参照してください。

ステータスメッセージ	説明および可能な処置
Path online	パスが現在正常に機能しています。処置は必要ありません。
Path waiting	パスが現在初期化中です。処置は必要ありません。
Faulted	パスが機能していません。これは、パスが一時的に待機状態とオンライン状態の間にある状態の可能性があります。再び clinterconnect status を実行してもメッセージが繰り返される場合は、適切な処置を行なってください。

例 7-1 クラスタインターコネクトのステータスを確認する

次に、正常に機能しているクラスタインターコネクトのステータスの例を示します。

```
% clinterconnect status
-- Cluster Transport Paths --
 Endpoint Endpoint Status
 ----- -
Transport path: phys-schost-1:qfe1 phys-schost-2:qfe1 Path online
Transport path: phys-schost-1:qfe0 phys-schost-2:qfe0 Path online
Transport path: phys-schost-1:qfe1 phys-schost-3:qfe1 Path online
Transport path: phys-schost-1:qfe0 phys-schost-3:qfe0 Path online
Transport path: phys-schost-2:qfe1 phys-schost-3:qfe1 Path online
Transport path: phys-schost-2:qfe0 phys-schost-3:qfe0 Path online
```

## ▼ クラスタトランスポートケーブル、トランスポートアダプタ、トランスポートスイッチを追加する方法

クラスタのプライベートトランスポートの要件については、『[Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual](#)』の「[Interconnect Requirements and Restrictions](#)」を参照してください。

この手順は、Oracle Solaris Cluster Manager GUI を使用しても実行できます。詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタトランスポートケーブルが物理的に取り付けられていることを確認します。

クラスタトランスポートケーブルのインストール手順については、『[Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual](#)』を参照してください。

- 2 クラスタ内にある任意のノード上でスーパーユーザーになります。

- 3 **clsetup** ユーティリティを起動します。

```
clsetup
```

メインメニューが表示されます。

- 4 クラスタインターコネクトメニューを表示するためのオプションに対応する番号を入力します。

- 5 トランスポートケーブルを追加するためのオプションに対応する番号を入力します。

指示に従い、必要な情報を入力します。

- 6 トランスポートアダプタをノードに追加するためのオプションに対応する番号を入力します。

指示に従い、必要な情報を入力します。

クラスタインターコネクトで次のアダプタのいずれかを使用する予定の場合、関連するエントリを各クラスタノードの `/etc/system` ファイルに追加します。このエントリは、次のシステム再ブート後に有効になります。

アダプタ	エントリ
ce	set ce:ce_taskq_disable=1
ipge	set ipge:ipge_taskq_disable=1
ixge	set ixge:ixge_taskq_disable=1

7    トランスポートスイッチを追加するためのオプションに対応する番号を入力します。  
指示に従い、必要な情報を入力します。

8    クラスタトランスポートケーブル、トランスポートアダプタ、トランスポートスイッチが追加されたことを確認します。

```
clinterconnect show node:adapter,adapternode
clinterconnect show node:adapter
clinterconnect show node:switch
```

例 7-2    クラスタトランスポートケーブル、トランスポートアダプタ、トランスポートスイッチの追加

次の例に、clsetup ユーティリティを使用し、トランスポートケーブル、トランスポートアダプタ、トランスポートスイッチをノードに追加する方法を示します。

```
[Ensure that the physical cable is installed.]
[Start the clsetup utility:]
clsetup
[Select Cluster interconnect]

[Select either Add a transport cable,
Add a transport adapter to a node,
or Add a transport switch.]
[Answer the questions when prompted.]
[You Will Need:]
[Information: Example:]
node names phys-schost-1
adapter names qfe2
switch names hub2
transport type dlpi
[Verify that the clinterconnect
command completed successfully:]Command completed successfully.
Quit the clsetup Cluster Interconnect Menu and Main Menu.
[Verify that the cable, adapter, and switch are added:]
clinterconnect show phys-schost-1:qfe2,hub2
===Transport Cables ===
Transport Cable: phys-schost-1:qfe2@0,hub2
Endpoint1: phys-schost-2:qfe0@0
Endpoint2: ethernet-1@2 ??? Should this be hub2?
State: Enabled

clinterconnect show phys-schost-1:qfe2
=== Transport Adepters for qfe2
```

```

Transport Adapter: qfe2
Adapter State: Enabled
Adapter Transport Type: dlpi
Adapter Property (device_name): ce
Adapter Property (device_instance): 0
Adapter Property (lazy_free): 1
Adapter Property (dlpi_heartbeat_timeout): 10000
Adpater Property (dlpi_heartbeat_quantum): 1000
Adapter Property (nw_bandwidth): 80
Adapter Property (bandwidth): 70
Adapter Property (ip_address): 172.16.0.129
Adapter Property (netmask): 255.255.255.128
Adapter Port Names: 0
Adapter Port SState (0): Enabled

clinterconnect show phys-schost-1:hub2

=== Transport Switches ===
Transport Switch: hub2
Switch State: Enabled
Switch Type: switch
Switch Port Names: 1 2
Switch Port State(1): Enabled
Switch Port State(2): Enabled

```

次の手順 クラスタトランスポートケーブルのインターコネクトのステータスを確認するには、[202 ページの「クラスタインターコネクトのステータスを確認する方法」](#)を参照してください。

## ▼ クラスタトランスポートケーブル、トランスポートアダプタ、トランスポートスイッチを削除する方法

この手順は、Oracle Solaris Cluster Manager GUI を使用しても実行できます。詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

次の手順を使用して、ノード構成からクラスタトランスポートケーブル、トランスポートアダプタ、およびトランスポートスイッチを削除します。ケーブルを無効にした場合、このケーブルの2つのエンドポイントは構成されたままになります。トランスポートケーブルの終端として使用されているアダプタは削除できません。



注意-各クラスタノードには、他のすべてのクラスタノードに対する(機能している)トランスポートパスが少なくとも1つずつ必要です。2つのノードは必ず接続されており、お互いに分離されているノードは存在しません。ケーブルを無効にする前には、必ず、ノードのクラスタインターコネクトのステータスを確認してください。状態が冗長な場合、つまり別の接続が使用できる場合だけ、ケーブル接続を無効にします。ノードの最後の機能しているケーブルを無効にすると、そのノードはクラスタメンバーシップから外れます。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタ内にある任意のノード上でスーパーユーザーになります。
- 2 残りのクラスタトランスポートパスのステータスを確認します。

```
clinterconnect status
```



注意-2 ノードクラスタのいずれかのノードを削除しようとして「パス障害 (Path faulted)」などのエラーメッセージが表示された場合、この手順を続ける前に問題を調査してください。このような問題は、ノードパスが利用できないことを示しています。残りの動作中のパスを削除すると、このノードはクラスタメンバーシップから外れ、クラスタが再構成されます。

- 3 `clsetup` ユーティリティを起動します。

```
clsetup
```

メインメニューが表示されます。

- 4 クラスタインターコネクトメニューにアクセスするためのオプションに対応する番号を入力します。
- 5 トランスポートケーブルを無効にするためのオプションに対応する番号を入力します。

指示に従い、必要な情報を入力します。アプリケーションのノード名、アダプタ名、およびスイッチ名を知っておく必要があります。

- 6 トランスポートケーブルを削除するためのオプションに対応する番号を入力します。

指示に従い、必要な情報を入力します。アプリケーションのノード名、アダプタ名、およびスイッチ名を知っておく必要があります。

---

注-物理的にケーブル接続を解除する場合は、ポートと宛先デバイスをつないでいるケーブルを切り離します。

---

- 7 トランスポートアダプタをノードから削除するためのオプションに対応する番号を入力します。

指示に従い、必要な情報を入力します。アプリケーションのノード名、アダプタ名、およびスイッチ名を知っておく必要があります。

---

注-物理アダプタをノードから取り外す場合は、『[Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual](#)』のハードウェアサービス手順を参照してください。

---

- 8 トランスポートスイッチを削除するためのオプションに対応する番号を入力します。

指示に従い、必要な情報を入力します。アプリケーションのノード名、アダプタ名、およびスイッチ名を知っておく必要があります。

---

注-ポートがトランスポートケーブルの終端として使用されている場合、スイッチは削除できません。

---

- 9 ケーブル、アダプタ、またはスイッチが削除されたことを確認します。

```
clinterconnect show node:adapter,adapternode
clinterconnect show node:adapter
clinterconnect show node:switch
```

ノードからトランスポートケーブルやトランスポートアダプタが削除された場合は、このコマンドの出力には表示されません。

### 例 7-3 トランスポートケーブル、トランスポートアダプタ、トランスポートスイッチの削除

次の例に、`clsetup` コマンドを使用して、トランスポートケーブル、トランスポートアダプタ、またはトランスポートスイッチを削除する方法を示します。

```
[Become superuser on any node in the cluster.]
[Start the utility:]
clsetup
[Select Cluster interconnect.]
[Select either Remove a transport cable,
Remove a transport adapter to a node,
```

```

or Remove a transport switch.[
[Answer the questions when prompted.[
 You Will Need:
 Information Example:
 node names phys-schost-1
 adapter names qfe1
 switch names hub1
[Verify that the clinterconnect
command was completed successfully:]
Command completed successfully.
[Quit the clsetup utility Cluster Interconnect Menu and Main Menu.]
[Verify that the cable, adapter, or switch is removed:]
clinterconnect show phys-schost-1:qfe2,hub2
===Transport Cables===
Transport Cable: phys-schost-2:qfe2@0,hub2
 Cable Endpoint1: phys-schost-2:qfe0@0
 Cable Endpoint2: ethernet-1@2 ??? Should this be hub2???
 Cable State: Enabled

clinterconnect show phys-schost-1:qfe2
=== Transport Adepters for qfe2
Transport Adapter: qfe2
 Adapter State: Enabled
 Adapter Transport Type: dlpi
 Adapter Property (device_name): ce
 Adapter Property (device_instance): 0
 Adapter Property (lazy_free): 1
 Adapter Property (dlpi_heartbeat_timeout): 10000
 Adpater Property (dlpi_heartbeat_quantum): 1000
 Adapter Property (nw_bandwidth): 80
 Adapter Property (bandwidth): 70
 Adapter Property (ip_address): 172.16.0.129
 Adapter Property (netmask): 255.255.255.128
 Adapter Port Names: 0
 Adapter Port STate (0): Enabled

clinterconnect show phys-schost-1:hub2
=== Transport Switches===
Transport Switch: hub2
 Switch State: Enabled
 Switch Type: switch
 Switch Port Names: 1 2
 Switch Port State(1): Enabled
 Switch Port State(2): Enabled

```

## ▼ クラスタトランスポートケーブルを有効にする方法

この手順は、Oracle Solaris Cluster Manager GUI を使用しても実行できます。詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

このオプションは、既存のクラスタトランスポートケーブルを有効にするために使用します。



phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタ内にある任意のノード上でスーパーユーザーになります。
- 2 **clsetup** ユーティリティーを起動します。  
**# clsetup**  
 メインメニューが表示されます。
- 3 クラスタインターコネクトメニューにアクセスするためのオプションに対応する番号を入力し、**Return** キーを押します。
- 4 トランスポートケーブルを有効にするためのオプションに対応する番号を入力し、**Return** キーを押します。  
 プロンプトが表示されたなら、指示に従います。ケーブルのいずれかの終端のノード名およびアダプタ名の両方を入力する必要があります。
- 5 ケーブルが有効になっていることを確認します。  
**# clinterconnect show node:adapter,adapternode**

#### 例 7-4 クラスタトランスポートケーブルを有効にする

次の例は、ノード phys-schost-2 にあるアダプタ qfe-1 のクラスタトランスポートケーブルを有効にする方法を示しています。

```
[Become superuser on any node.]
[Start the clsetup utility:]
clsetup
[Select Cluster interconnect>Enable a transport cable.]

[Answer the questions when prompted.]
[You will need the following information.]
 You Will Need:
Information: Example:
node names phys-schost-2
adapter names qfe1
switch names hub1
[Verify that the scinterconnect
command was completed successfully:]

clinterconnect enable phys-schost-2:qfe1

Command completed successfully.
[Quit the clsetup Cluster Interconnect Menu and Main Menu.]
```

```
[Verify that the cable is enabled:]
clinterconnect show phys-schost-1:qfe2,hub2
Transport cable: phys-schost-2:qfe1@0 ethernet-1@2 Enabled
Transport cable: phys-schost-3:qfe0@1 ethernet-1@3 Enabled
Transport cable: phys-schost-1:qfe0@0 ethernet-1@1 Enabled
```

## ▼ クラスタトランスポートケーブルを無効にする方法

この手順は、Oracle Solaris Cluster Manager GUI を使用しても実行できます。詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

クラスタトランスポートケーブルを無効にして、クラスタインターコネクトパスを一時的に停止する必要がある場合があります。一時的な停止は、クラスタインターコネクトで発生する問題の解決や、クラスタインターコネクトのハードウェアの交換に便利です。

ケーブルを無効にした場合、このケーブルの2つのエンドポイントは構成されたままになります。トランスポートケーブルの終端として使用されているアダプタは削除できません。



注意 - 各クラスタノードには、他のすべてのクラスタノードに対する (機能している) トランスポートパスが少なくとも1つずつ必要です。2つのノードは必ず接続されており、お互いに分離されているノードは存在しません。ケーブルを無効にする前には、必ず、ノードのクラスタインターコネクトのステータスを確認してください。状態が冗長な場合、つまり別の接続が使用できる場合だけ、ケーブル接続を無効にします。ノードの最後の機能しているケーブルを無効にすると、そのノードはクラスタメンバーシップから外れます。

**phys-schost#** プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタ内にある任意のノード上でスーパーユーザーになります。
- 2 ケーブルを無効にする前に、クラスタインターコネクトのステータスを確認します。

```
clinterconnect status
```



注意-2 ノードクラスタのいずれかのノードを削除しようとして「パス障害 (Path faulted)」などのエラーメッセージが表示された場合、この手順を続ける前に問題を調査してください。このような問題は、ノードパスが利用できないことを示しています。残りの動作中のパスを削除すると、このノードはクラスタメンバーシップから外れ、クラスタが再構成されます。

- 3 **clsetup** ユーティリティを起動します。

```
clsetup
```

メインメニューが表示されます。

- 4 クラスタインターコネクトメニューにアクセスするためのオプションに対応する番号を入力し、**Return** キーを押します。

- 5 トランスポートケーブルを無効にするためのオプションに対応する番号を入力し、**Return** キーを押します。

指示に従い、必要な情報を入力します。このクラスタインターコネクトのすべてのコンポーネントは無効になります。ケーブルのいずれかの終端のノード名およびアダプタ名の両方を入力する必要があります。

- 6 ケーブルが無効になっていることを確認します。

```
clinterconnect show node:adapter,adapternode
```

### 例 7-5 クラスタトランスポートケーブルを無効にする

次の例は、ノード phys-schost-2 にあるアダプタ qfe-1 のクラスタトランスポートケーブルを無効にする方法を示しています。

```
[Become superuser on any node.]
[Start the clsetup utility:]
clsetup
[Select Cluster interconnect>Disable a transport cable.]

[Answer the questions when prompted.]
[You will need the following information.]
[You Will Need:]
Information: Example:
node names phys-schost-2
adapter names qfe1
switch names hub1
[Verify that the clinterconnect
command was completed successfully:]
Command completed successfully.
[Quit the clsetup Cluster Interconnect Menu and Main Menu.]
[Verify that the cable is disabled:]
clinterconnect show -p phys-schost-1:qfe2,hub2
Transport cable: phys-schost-2:qfe1@0 ethernet-1@2 Disabled
Transport cable: phys-schost-3:qfe0@1 ethernet-1@3 Enabled
Transport cable: phys-schost-1:qfe0@0 ethernet-1@1 Enabled
```

▼ トランスポートアダプタのインスタンス番号を確認する方法

clsetup コマンドを使用して正しいトランスポートアダプタの追加と削除を行うには、トランスポートアダプタのインスタンス番号を確認する必要があります。アダプタ名は、アダプタの種類とアダプタのインスタンス番号を組み合わせたものです。

- 1 スロット番号にもとづき、アダプタの名前を確認してください。  
次の画面は例であり、個々のハードウェアと一致しない可能性があります。

```
prtdiag
...
===== IO Cards =====
 Bus Max
 Bus Max
 IO Port Bus Freq Bus Dev,
 Type ID Side Slot MHz Freq Func State Name Model

XYZ 8 B 2 33 33 2,0 ok xyz11c8,0-xyz11c8,d665.11c8.0.0
XYZ 8 B 3 33 33 3,0 ok xyz11c8,0-xyz11c8,d665.11c8.0.0
...
```

- 2 アダプタのパスを使用して、アダプタのインスタンス番号を確認してください。  
次の画面は例であり、個々のハードウェアと一致しない可能性があります。

```
grep sci /etc/path_to_inst
"/xyz@1f,400/pci11c8,0@2" 0 "ttt"
"/xyz@1f,4000.pci11c8,0@4 "ttt"
```

- 3 アダプタの名前とスロット番号を使用してアダプタのインスタンス番号を確認してください。  
次の画面は例であり、個々のハードウェアと一致しない可能性があります。

```
prtconf
...
xyz, instance #0
 xyz11c8,0, instance #0
 xyz11c8,0, instance #1
...
```

▼ 既存のクラスタのプライベートネットワークアドレスまたはアドレス範囲を変更する方法

プライベートネットワークアドレスまたは使用されるネットワークアドレスの範囲、またはその両方を変更するには、次の手順に従います。

始める前に      スーパーユーザーのリモートシェルの rsh(1M) または Secure Shell の ssh(1) アクセスが、すべてのクラスタノードで有効になっていることを確認します。詳細は、[rsh\(1M\)](#) または [ssh\(1\)](#) のマニュアルページを参照してください。

- 1 各クラスタノード上で次のサブステップを実行することで、すべてのクラスタノードをリブートし、非クラスタモードにします。
  - a. 非クラスタモードで起動するクラスタノード上で、スーパーユーザーになるか、RBACの承認 `solaris.cluster.admin` を提供する役割になります。
  - b. `clnode evacuate` および `cluster shutdown` コマンドを使用してノードを停止します。  
`clnode evacuate` コマンドは、すべてのデバイスグループを、指定ノードから次に優先されるノードに切り替えます。またこのコマンドは、指定ノード上の投票または非投票ノードから、次に優先される投票または非投票ノードへ、すべてのリソースグループを切り替えます。  

```
clnode evacuate node
cluster shutdown -g0 -y
```
- 2 1つのノードから、`clsetup` ユーティリティを起動します。  
 非クラスタモードで動作している場合、`clsetup` ユーティリティは非クラスタモード動作のメインメニューを表示します。
- 3 「クラスタトランスポート用のネットワークアドレス指定と範囲の変更」のオプションに対応する番号を入力し、**Return** キーを押します。  
`clsetup` ユーティリティは現在のプライベートネットワーク構成を表示し、この構成を変更するかどうかを尋ねます。
- 4 プライベートネットワークのIPアドレスとIPアドレス範囲のいずれかを変更するには、**yes** と入力し、**Return** キーを押します。  
`clsetup` ユーティリティはデフォルトのプライベートネットワークIPアドレスである `172.16.0.0` を表示し、このデフォルトをそのまま使用してもよいかどうかを尋ねます。
- 5 プライベートネットワークIPアドレスを変更するか、そのまま使用します。
  - デフォルトのプライベートネットワークIPアドレスをそのまま使用し、IPアドレス範囲の変更に進むには、**yes** と入力し、**Return** キーを押します。  
`clsetup` ユーティリティは、デフォルトのネットマスクをそのまま使用してもよいかどうかを尋ねます。次の手順に進み、応答を入力します。

- デフォルトのプライベートネットワーク IP アドレスを変更するには、次のサブステップを実行します。
  - a. **clsetup** ユーティリティーの、デフォルトのアドレスをそのまま使用してもよいかどうかに関する質問に対しては **no** と入力し、**Return** キーを押します。  
**clsetup** ユーティリティーは、新しいプライベートネットワーク IP アドレスを入力するプロンプトを表示します。
  - b. 新しい IP アドレスを入力し、**Return** キーを押します。  
**clsetup** ユーティリティーはデフォルトのネットマスクを表示し、デフォルトのネットマスクをそのまま使用してもよいかどうかを尋ねます。
- 6 デフォルトのプライベートネットワーク IP アドレス範囲を変更するか、そのまま使用します。  
 デフォルトのネットマスクは 255.255.240.0 です。このデフォルトの IP アドレス範囲は、クラスタ内で最大 64 のノード、最大 12 のゾーンクラスタ、および最大 10 のプライベートネットワークをサポートします。
  - デフォルトの IP アドレス範囲をそのまま使用するには、「**yes**」と入力して、**Return** キーを押します。  
 そして次の手順に進みます。
  - IP アドレス範囲を変更するには、次のサブステップを実行します。
    - a. **clsetup** ユーティリティーの、デフォルトのアドレス範囲をそのまま使用してもよいかどうかに関する質問に対しては **no** と入力し、**Return** キーを押します。  
 デフォルトのネットマスクを使用しない場合、**clsetup** ユーティリティーは、クラスタで構成する予定のノード、プライベートネットワーク、およびゾーンクラスタの数を入力するプロンプトを表示します。
    - b. クラスタで構成する予定のノード、プライベートネットワーク、およびゾーンクラスタの数を入力します。  
 これらの数から、**clsetup** ユーティリティーは 2 つの推奨ネットマスクを計算します。
      - 第一のネットマスクは、ユーザーが指定したノード、プライベートネットワーク、およびゾーンクラスタの数をサポートする、最低限のネットマスクです。
      - 第二のネットマスクは、将来ありうる成長に対応するため、ユーザーが指定したノード、プライベートネットワーク、およびゾーンクラスタの数の 2 倍をサポートします。

- c. 計算されたネットマスクのいずれかを指定するか、ノード、プライベートネットワーク、およびゾーンクラスタの予定数をサポートする別のネットマスクを指定します。
- 7 更新を使用して続行するかに関する **clsetup** ユーティリティの質問に対しては、**yes** と入力します。
- 8 完了後 **clsetup** ユーティリティを終了します。
- 9 各クラスタノードに対して次のサブステップを実行することで、各クラスタノードをリブートし、クラスタモードに戻します。
  - a. ノードをブートします。
    - SPARC ベースのシステム上で、次のコマンドを実行します。  
`ok boot`
    - x86 ベースのシステム上で、次のコマンドを実行します。  
GRUB メニューが表示された時点で、適切な Oracle Solaris エントリを選択し、Enter キーを押します。次のような GRUB メニューが表示されます。  
  

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-----+
| Solaris 10 /sol_10_x86 |
| Solaris failsafe |
| |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```
- 10 ノードが問題なくブートし、オンラインであることを確認します。  
`# cluster status -t node`

## パブリックネットワークの管理

Oracle Solaris Cluster ソフトウェアは、Oracle Solaris ソフトウェアに実装されている、パブリックネットワーク用の Internet Protocol network Multipathing (IPMP) をサポートします。IPMP の基本的な管理は、クラスタ環境でも非クラスタ環境でも同じです。マルチパスは Oracle Solaris 10 OS をインストールすると自動的にインストールされますが、使用するには有効にする必要があります。マルチパスの管理については、適切な Oracle Solaris OS のドキュメントを参照してください。ただし、Oracle Solaris Cluster 環境で IPMP を管理する前にガイドラインを確認してください。

## クラスタで IP ネットワークマルチパスグループを管理する方法

IPMP 手順をクラスタ上で実行する前に、次のガイドラインについて考慮してください。

- 各パブリックネットワークアダプタは、IPMP グループに属している必要があります。
- `local-mac-address?` 変数には、Ethernet アダプタの値として `true` が指定されていなければなりません。
- クラスタでは、プローブベースの IPMP グループ、またはリンクベースの IPMP グループを使用できます。プローブベースの IPMP グループは、ターゲットの IP アドレスをテストし、優れた保護を提供しますが、可用性が損なわれる場合もあります。
- 次に示すタイプのマルチパスグループ内に存在するアダプタごとにテスト IP アドレスを構成する必要があります。
  - すべての多重アダプタマルチパスグループで、テスト IP アドレスが必要です。単一アダプタマルチパスグループでは、テスト IP アドレスは必要ありません。
- 同一マルチパスグループ内のすべてのアダプタ用のテスト IP アドレスは、単一の IP サブネットに属する必要があります。
- テスト IP アドレスは高可用性でないため、通常のアプリケーションが使用しないようにします。
- マルチパスグループの命名に制限はありません。しかし、リソースグループを構成するとき、`netiflist` には、任意のマルチパス名にノード ID 番号またはノード名が続くものを指定します。たとえば、マルチパスグループの名前が `sc_ipmp0` であるとき、ノード ID が 1 である `phys-schost-1` というノード上にアダプタが存在する場合、`netiflist` には `sc_ipmp0@1` と `sc_ipmp0@phys-schost-1` のどちらを指定してもかまいません。
- `if_mpadm` コマンドを使用して、削除するアダプタからグループ内の代替アダプタに IP アドレスを最初に切り替えてから、IP ネットワークマルチパスグループのアダプタを構成解除 (`unplumb`) または停止してください。詳細は、[if\\_mpadm\(1M\)](#) のマニュアルページを参照してください。
- 個々のマルチパスグループから削除する前に、アダプタを別のサブネットに配線しないようにします。
- 論理アダプタ操作は、マルチパスグループで監視中の場合でもアダプタに対して行うことができます。
- クラスタ内の各ノードについて、最低 1 つのパブリックネットワーク接続を維持しなければなりません。クラスタは、パブリックネットワーク接続がないとアクセスできません。



- クラスタ上の IP ネットワークマルチパスグループのステータスを表示するには、`clinterconnect status` コマンドを使用します。

IP ネットワークマルチパスの詳細は、Oracle Solaris OS システム管理ドキュメントセットの該当するドキュメントを参照してください。

表 7-3 タスクリスト:パブリックネットワークの管理

Oracle Solaris OS のリリース	手順
Oracle Solaris 10 OS	『Oracle Solaris の管理: IP サービス』の IP ネットワークマルチパスに関するトピック

クラスタソフトウェアをインストールする手順については、『Oracle Solaris Cluster ソフトウェアのインストール』を参照してください。パブリックネットワークハードウェアコンポーネントをサービスする手順については、『Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual』を参照してください。

## パブリックネットワークインタフェースでの動的再構成

クラスタ内のパブリックネットワークインタフェース上で動的再構成 (DR) を実行するときには、いくつかの問題を考える必要があります。

- Oracle Solaris の DR 機能の説明で述べられているすべての必要条件、手順、制限は、Oracle Solaris Cluster の DR サポートにも適用されます (オペレーティングシステムでの休止操作は除く)。したがって、Oracle Solaris Cluster ソフトウェアで DR 機能を使用する前に、Oracle Solaris の DR 機能についての説明を参照してください。特に、DR の切り離し操作中に、ネットワークに接続されていない入出力デバイスに影響する問題について確認してください。
- DR ボード削除操作は、パブリックネットワークインタフェースがアクティブでないときだけ成功します。アクティブなパブリックネットワークインタフェースを削除する前に、`if_mpadm(1M)` コマンドを使用して、マルチパスグループ内の削除する予定のアダプタから別のアダプタに IP アドレスを切り換えます。
- アクティブなネットワークインタフェースを適切に無効にせずにパブリックネットワークインタフェースカードを削除しようとした場合、Oracle Solaris Cluster はその操作を拒否して、その操作から影響を受けるインタフェースを識別します。



注意-2つのアダプタを持つマルチパスグループの場合、無効にしたネットワークアダプタ上でDR 削除操作を実行している間に残りのネットワークアダプタに障害が発生すると、可用性に影響が生じます。これは、DR 操作の間は、残りのネットワークアダプタのフェイルオーバー先が存在しないためです。

パブリックネットワークインタフェース上でDR 操作を実行するときは、次の手順をその順番どおりに行います。

表 7-4 タスクマップ:パブリックネットワークインタフェースでの動的再構成

タスク	手順
1.if_mpadmを使用して、マルチパスグループ内の削除する予定のアダプタから別のアダプタへのIPアドレスの切り換えを実行	<a href="#">if_mpadm(1M)</a> のマニュアルページ 『Oracle Solaris の管理:IP サービス』のパート V 「IPMP」
2.ifconfig コマンドを使用して、マルチパスグループからアダプタを削除	<a href="#">ifconfig(1m)</a> のマニュアルページ 『Oracle Solaris の管理:IP サービス』のパート V 「IPMP」
3.パブリックネットワークインタフェース上でDR 操作を実行	

## ノードの追加と削除

---

この章では、クラスタへのノードの追加とノードの削除方法を説明します。

- 219 ページの「クラスタへのノードの追加」
- 225 ページの「クラスタからのノードの削除」

クラスタのメンテナンスに関する情報は、第9章「クラスタの管理」を参照してください。

### クラスタへのノードの追加

このセクションでは、グローバルクラスタまたはゾーンクラスタにノードを追加する方法を説明します。新しいゾーンクラスタノードは、そのゾーンクラスタをホストするグローバルクラスタのノード上に作成できますが、それは、グローバルクラスタノードが、そのゾーンクラスタのノードをまだホストしていない場合に限られます。グローバルクラスタ上の既存の非投票ノードをゾーンクラスタノードに変換することはできません。

各ゾーンクラスタノードの IP アドレスおよび NIC の指定は任意です。

---

注- 各ゾーンクラスタノードで IP アドレスを構成しない場合、次の2つのことが発生します。

1. その特定のゾーンクラスタでは、ゾーンクラスタで使用するための NAS デバイスを構成することができません。NAS デバイスと通信するにはゾーンクラスタノードの IP アドレスを使用するため、IP アドレスを持たないクラスタは、NAS デバイスのフェンシングをサポートできません。
  2. クラスタソフトウェアによって、NIC の論理ホスト IP アドレスが有効化されます。
-

元のゾーンクラスタノード用 IP アドレスまたは NIC を指定しなかった場合、新しいゾーンクラスタノード用にその情報を指定する必要はありません。

この章での `phys-schost#` は、グローバルクラスタのプロンプトを表します。`clzonecluster` の対話型シェルプロンプトは `clzc:schost>` です。

次の表に、ノードを既存のクラスタに追加するときに行うタスクを示します。タスクは、示されている順に実行してください。

表 8-1 タスクマップ:既存のグローバルクラスタまたはゾーンクラスタへのノードの追加

タスク	手順
ホストアダプタのノードへの取り付けと、既存のクラスタインターコネクトが新しいノードをサポートできることの確認	『Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual』
共有ストレージの追加	『Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual』
認証ノードリストへのノードの追加	<code>/usr/cluster/bin/claccess allow -h node-being-added</code>
新しいクラスタノードへのソフトウェアのインストールと構成	『Oracle Solaris Cluster ソフトウェアのインストール』の第2章「グローバルクラスタノードへのソフトウェアのインストール」
既存のクラスタに新しいノードを追加する	220 ページの「既存のクラスタにノードを追加する方法」
クラスタが Oracle Solaris Cluster Geographic Edition のパートナーシップで構成されている場合、構成内のアクティブな参加メンバーとして新しいノードを構成する	『Oracle Solaris Cluster Geographic Edition System Administration Guide』の「How to Add a New Node to a Cluster in a Partnership」

## ▼ 既存のクラスタにノードを追加する方法

Oracle Solaris ホストまたは仮想マシンを既存のグローバルクラスタまたはゾーンクラスタに追加する前に、プライベートクラスタインターコネクトへの運用面の物理接続を含む、必要なハードウェアすべてがノードに正しく取り付けられ、構成されていることを確認してください。

ハードウェアのインストールについては、『Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual』またはサーバーに付属するハードウェアのドキュメントを参照してください。

この手順によって、マシンは自分自身をクラスタ内にインストールします。つまり、自分のノード名を当該クラスタの認証ノードリストに追加します。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 **表 8-1** のタスクマップに記載されている必要なハードウェアのインストールと構成タスクをすべて正しく完了していることを確認します。
- 2 **scinstall** ユーティリティを使用してスイッチ、アダプタ、およびケーブルが作成されたことを確認してください
- 3 追加されるノードで、**scinstall** ユーティリティを使用して、クラスタノードを追加するメニュー項目を選択し、プロンプトに従います。
- 4 ゾーンクラスタに手動でノードを追加するには、次の追加の手順に従います。

- a. **Oracle Solaris** ホストおよび仮想ノード名を指定します。また、各ノードでパブリックネットワーク通信に使用するネットワークリソースも指定してください。次の例では、ゾーン名は **sczone** で、**bge0** は両方のマシン上のパブリックネットワークアダプタです

```
clzc:sczone>add node
clzc:sczone:node>set physical-host=phys-cluster-1
clzc:sczone:node>set hostname=hostname1
clzc:sczone:node>add net
clzc:sczone:node:net>set address=hostname1
clzc:sczone:node:net>set physical=bge0
clzc:sczone:node:net>end
clzc:sczone:node>end
clzc:sczone>add node
clzc:sczone:node>set physical-host=phys-cluster-2
clzc:sczone:node>set hostname=hostname2
clzc:sczone:node>add net
clzc:sczone:node:net>set address=hostname2
clzc:sczone:node:net>set physical=bge0
clzc:sczone:node:net>end
clzc:sczone:node>end
```

- b. ノードの構成後、ノードをクラスタモードでリブートし、そのノードにゾーンクラスタをインストールします。

```
clzc install zone-clustername
```

ゾーンクラスタノードを構成する手順の詳細は、『Oracle Solaris Cluster Software Installation Guide』の『[Oracle Solaris Cluster ソフトウェアのインストール](#)』の「[ゾーンクラスタの構成](#)」を参照してください。

- 5 新しいマシンがクラスタに追加されないようにするために、新しいマシンを追加する要求を無視するようクラスタに指示するオプションの番号を **clsetup** ユーティリティーから入力します。

clsetup のプロンプトに従います。このオプションを設定すると、クラスタは、自分自身をクラスタに追加しようとする新しいマシンからのパブリックネットワーク経由の要求をすべて無視します。完了後 clsetup ユーティリティーを終了します。

### 例 8-1 認証ノードリストへのグローバルクラスタノードの追加

次に、ノード phys-schost-3 を既存のクラスタの認証ノードリストに追加する方法を示します。

```
[Become superuser and execute the clsetup utility.]
phys-schost# clsetup
[Select New nodes>Specify the name of a machine which may add itself.]
[Answer the questions when prompted.]
[Verify that the command completed successfully.]

claccess allow -h phys-schost-3

Command completed successfully.
[Select Prevent any new machines from being added to the cluster.]
[Quit the clsetup New Nodes Menu and Main Menu.]
[Install the cluster software.]
```

#### 参照 clsetup(1CL)

クラスタノードを追加するタスクの一連の手順については、[表 8-1](#)、「タスクマップ: クラスタノードの追加」を参照してください。

ノードを既存のリソースグループに追加する方法については、『[Oracle Solaris Cluster Data Services Planning and Administration Guide](#)』を参照してください。

## グローバルクラスタ内での非投票ノード (ゾーン) の作成

このセクションでは、グローバルクラスタノード上に、単にゾーンと呼ばれる非投票ノードを作成する場合の情報と手順を説明します。

### ▼ グローバルクラスタ内に非投票ノードを作成する方法

- 1 非投票ノードを作成するグローバルクラスタノードでスーパーユーザーになります。  
大域ゾーンで作業してください。

- ノードのサービスがまだオンラインでない場合は、次のステップに進む前に状態がオンラインに変わるまで待ちます。

新しいゾーンを構成、インストール、およびブートします。

Oracle Solaris のドキュメントの手順に従ってください。

- 4 ゾーンが **ready** 状態であることを確認します。

5 (省略可能) 共有 IP ゾーンでは、プライベート IP アドレスとプライベートホスト名をゾーンに割り当てます。

次のコマンドは、クラスタのプライベート IP アドレスの範囲から、使用可能な IP アドレスを選択し、割り当てます。またこのコマンドは、指定されたプライベートホスト名、またはホスト別名をゾーンに割り当て、割り当てられたプライベート IP アドレスにそれをマッピングします。

```
phys-schost# clnode set -p zprivatehostname=hostalias node:zone
```

-p プロパティを指定します。

`zprivatehostname=hostalias` ゾーンプライベートホスト名、またはホスト別名を指定します。

<i>node</i>	ノードの名前。
-------------	---------

*zone* グローバルクラスタの非投票ノードの名前。

6 初期内部ゾーン構成を実行します。

『Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン』の「初期内部ゾーン構成を実行する」の指示に従います。次のどちらかの方法を選択します。

- ゾーンにログインします。
- `/etc/sysidcfg` ファイルを使用します。

7 非投票ノードで、`nsswitch.conf` ファイルを変更します。

これらの変更により、クラスタ固有のホスト名と IP アドレスの検索をゾーンが解決できるようになります。

a. ゾーンにログインします。

```
phys-schost# zlogin -c zonename
```

b. 編集のために `/etc/nsswitch.conf` ファイルを開きます。

```
sczone# vi /etc/nsswitch.conf
```

c. `hosts` および `netmasks` エントリの検索の先頭に `cluster` スイッチを追加し、`files` スイッチを続けます。

変更したエントリは次のようになります。

```
...
hosts: cluster files nis [NOTFOUND=return]
...
netmasks: cluster files nis [NOTFOUND=return]
...
```

d. ほかのすべてのエントリで、エントリに記載される最初のスイッチが `files` スイッチであることを確認します。

e. ゾーンを終了します。

8 排他的 IP ゾーンを作成した場合は、ゾーン上の各 `/etc/hostname.interface` ファイルで IPMP グループを構成します。

ゾーン内のデータサービストラフィックに使用されているパブリックネットワークアダプタごとに、IPMP グループを構成します。この情報は、大域ゾーンから継承されません。IPMP グループをクラスタ内に構成する方法の詳細は、『Oracle Solaris Cluster ソフトウェアのインストール』の「パブリックネットワーク」を参照してください。



- 9    ゾーンにより使用されるすべての論理ホスト名リソースの名前とアドレスのマッピングを設定します。
- a.    ゾーン上の `/etc/inet/hosts` ファイルに名前とアドレスのマッピングを追加します。  
        この情報は、大域ゾーンから継承されません。
- b.    ネームサーバーを使用している場合は、名前とアドレスのマッピングを追加します。

## クラスタからのノードの削除

このセクションでは、グローバルクラスタまたはゾーンクラスタ上のノードを削除する方法について説明します。グローバルクラスタから特定のゾーンクラスタを削除することもできます。次の表に、ノードを既存のクラスタから削除するときに行うタスクを示します。タスクは、示されている順に実行してください。



注意-RAC 構成の場合、この手順のみを使用してノードを削除すると、リブート中のノードでパニックが発生する可能性があります。RAC 構成からノードを削除する方法については、『[Oracle Solaris Cluster Data Service for Oracle Real Application Clusters ガイド](#)』の「[選択したノードから Oracle RAC 用サポートを削除する](#)」を参照してください。このプロセスを完了したら、次の該当する手順に従います。

表 8-2    タスクマップ: ノードの削除

タスク	手順
削除するノードからすべてのリソースグループとデバイスグループを移動する。ゾーンクラスタがある場合は、ゾーンクラスタにログインし、アンインストールされる物理ノード上にあるゾーンクラスタノードを退避させます。その後、物理ノードを停止する前に、ゾーンクラスタからそのノードを削除します。	<code>clnode evacuate node</code> <a href="#">226 ページの「ゾーンクラスタからノードを削除する方法」</a>
許可されたホストをチェックして、ノードを削除できることを確認する。	<code>claccess show</code> <code>claccess allow -h node-to-remove</code>
ノードを削除できない場合に、クラスタ構成へのアクセス権をノードに付与する。	

表 8-2 タスクマップ: ノードの削除 (続き)	
タスク	手順
すべてのデバイスグループからノードを削除する。	135 ページの「デバイスグループからノードを削除する方法 (Solaris Volume Manager)」
削除するノードに接続されているすべての定足数デバイスを削除する。	2 ノードクラスタのノードを削除する場合、この手順はオプションです。  181 ページの「定足数デバイスを削除する方法」  次の手順では、ストレージデバイスを削除する前に定足数デバイスを削除する必要がありますが、定足数デバイスはその後直後に追加し直すことができます。  183 ページの「クラスタから最後の定足数デバイスを削除する方法」
削除するノードを非クラスタモードにする。	251 ページの「ノードを保守状態にする」
ゾーンクラスタからノードを削除する。	226 ページの「ゾーンクラスタからノードを削除する方法」
クラスタソフトウェア構成からノードを削除する。	227 ページの「クラスタソフトウェア構成からノードを削除する方法」
(オプション) Oracle Solaris Cluster ソフトウェアをクラスタノードからアンインストールする。	255 ページの「クラスタノードから Oracle Solaris Cluster ソフトウェアをアンインストールする」

## ▼ ゾーンクラスタからノードを削除する方法

ノードを停止してアンインストールし、構成からノードを削除することによって、ゾーンクラスタからノードを削除できます。あとでノードをゾーンクラスタに戻す場合は、表 8-1 の手順に従います。ここからの手順のほとんどは、グローバルクラスタノードから実行します。

始める前に この手順を実行する前に、表 8-2 を参照し、実行すべき手順があるかどうかを確認してください。

- 1 グローバルクラスタのノードのスーパーユーザーになります。
- 2 ノードとそのゾーンクラスタを指定して、削除するゾーンクラスタノードを停止します。

```
phys-schost# clzonecluster halt -n node zoneclustername
```

ゾーンクラスタ内で `clnode evacuate` コマンドと `shutdown` コマンドを使用することもできます。

- 3 ゾーンクラスタ内のすべてのリソースグループからノードを削除します。

```
phys-schost# clrg remove-node -n zonehostname -Z zoneclustername rg-name
```

- 4 ゾーンクラスタノードをアンインストールします。

```
phys-schost# clzonecluster uninstall -n node zoneclustername
```

- 5 ゾーンクラスタノードを構成から削除します。

次のコマンドを使用します。

```
phys-schost# clzonecluster configure zoneclustername
```

```
clzc:sczone> remove node physical-host=node
```

```
clzc:sczone> exit
```

- 6 ノードがゾーンクラスタから削除されたことを確認します。

```
phys-schost# clzonecluster status
```

## ▼ クラスタソフトウェア構成からノードを削除する方法

この手順を実行して、ノードをグローバルクラスタから削除します。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 この手順を実行する前に、ノードをすべてのリソースグループ、デバイスグループ、および定足数デバイスの構成から削除していること、および、このノードを保守状態にしていることを確認します。
- 2 削除するノードで、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.modify` を提供する役割になります。グローバルクラスタのノードから、次の手順のステップをすべて実行します。
- 3 削除するグローバルクラスタノードを非クラスタモードでブートします。ゾーンクラスタノードの場合は、この手順を実行する前に、[226 ページの「ゾーンクラスタからノードを削除する方法」](#)の手順を実行します。

- SPARC ベースのシステム上で、次のコマンドを実行します。

```
ok boot -x
```

- x86 ベースのシステム上で、次のコマンドを実行します。

```
shutdown -g -y -i0
```

Press any key to continue

- a. GRUB メニューで矢印キーを使用して該当する **Oracle Solaris** エントリを選択し、**e** と入力してコマンドを編集します。

次のような GRUB メニューが表示されます。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
```

```
+-----+
| Solaris 10 /sol_10_x86 |
| Solaris failsafe |
| |
+-----+
```

Use the ^ and v keys to select which entry is highlighted.

Press enter to boot the selected OS, 'e' to edit the commands before booting, or 'c' for a command-line.

GRUB ベースのブートの詳細については、『[Oracle Solaris の管理: 基本管理](#)』の「[GRUB を使用して x86 システムをブートする \(タスクマップ\)](#)」を参照してください。

- b. ブートパラメータの画面で、矢印キーを使用してカーネルエントリを選択し、**e** と入力してエントリを編集します。

次のような GRUB ブートパラメータの画面が表示されます。

```
GNU GRUB version 0.95 (615K lower / 2095552K upper memory)
```

```
+-----+
| root (hd0,0,a) |
| kernel /platform/i86pc/multiboot |
| module /platform/i86pc/boot_archive |
+-----+
```

Use the ^ and v keys to select which entry is highlighted.

Press 'b' to boot, 'e' to edit the selected command in the boot sequence, 'c' for a command-line, 'o' to open a new line after ('O' for before) the selected line, 'd' to remove the selected line, or escape to go back to the main menu.

- c. コマンドに **-x** を追加して、システムを非クラスタモードでブートするように指定します。

```
[Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time exits.]
```

```
grub edit> kernel /platform/i86pc/multiboot -x
```

- d. **Enter** キーを押して変更を受け入れ、ブートパラメータの画面に戻ります。

画面には編集されたコマンドが表示されます。

```
GNU GRUB version 0.95 (615K lower / 2095552K upper memory)
+-----+
| root (hd0,0,a) |
| kernel /platform/i86pc/multiboot -x |
| module /platform/i86pc/boot_archive |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.-
```

- e. **b** と入力して、ノードを非クラスタモードでブートします。

カーネルブートパラメータコマンドへのこの変更は、システムをブートすると無効になります。次にノードをリブートする際には、ノードはクラスタモードでブートします。クラスタモードではなく、非クラスタモードでブートするには、これらの手順を再度実行して、カーネルブートパラメータコマンドに `-x` オプションを追加します。

---

注-削除するノードが使用できない場合や、ブートできなくなっている場合は、アクティブな任意のクラスタノードで **`clnode clear -F <node-to-be-removed>`** コマンドを実行します。**`clnode status <nodename>`** を実行して、ノードが削除されていることを確認します。

---

- 4 クラスタからノードを削除します。

アクティブなノードから次のコマンドを実行します。

```
phys-schost# clnode clear -F nodename
```

`rg_system=true` が設定されているリソースグループがある場合、`clnode clear -F` コマンドが成功するためには、それらを `rg_system=false` に変更する必要があります。`clnode clear -F` を実行したあとに、そのリソースグループを `rg_system=true` に戻します。

削除するノードから次のコマンドを実行します。

```
phys-schost# clnode remove -F
```

---

注-クラスタ内の最後のノードを削除する場合は、そのノードがクラスタモードでないこと、およびクラスタ内にアクティブなノードがないことが必要です。

---

- 5 別のクラスタノードから、ノードの削除を確認します。

```
phys-schost# clnode status nodename
```

## 6 ノードの削除を完了します。

- 削除するノードから **Oracle Solaris Cluster** ソフトウェアをアンインストールする場合は、[255 ページの「クラスタノードから Oracle Solaris Cluster ソフトウェアをアンインストールする」](#)に進んでください。
- 削除するノードから **Oracle Solaris Cluster** ソフトウェアをアンインストールしない場合は、『[Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual](#)』で説明されているように、ハードウェア接続を削除することにより、クラスタからノードを物理的に削除できます。

### 例 8-2 クラスタソフトウェア構成からのノードの削除

次に、ノード `phys-schost-2` をクラスタから削除する方法を示します。`clnode remove` コマンドは、クラスタから削除するノード (`phys-schost-2`) から非クラスタモードで実行されます。

```
[Remove the node from the cluster:]
phys-schost-2# clnode remove
phys-schost-1# clnode clear -F phys-schost-2
[Verify node removal:]
phys-schost-1# clnode status
-- Cluster Nodes --
```

	Node name	Status
	-----	-----
Cluster node:	phys-schost-1	Online

**参照** 削除するノードから **Oracle Solaris Cluster** ソフトウェアをアンインストールする方法については、[255 ページの「クラスタノードから Oracle Solaris Cluster ソフトウェアをアンインストールする」](#)を参照してください。

ハードウェア手順については、『[Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual](#)』を参照してください。

クラスタノードを削除するタスクの総合的な一覧については、[表 8-2](#)を参照してください。

既存のクラスタにノードを追加するには、[220 ページの「既存のクラスタにノードを追加する方法」](#)を参照してください。

## ▼ グローバルクラスタから非投票ノード(ゾーン)を削除する方法

- 1 非投票ノードを作成したグローバルクラスタノードでスーパーユーザーになります。

- 2 システムから非投票ノードを削除します。

『Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン』の「システムから非大域ゾーンを削除する」の指示に従います。

## ▼ 2 ノード接続より大きなクラスタでアレイと単一ノード間の接続を削除する方法

3 ノードまたは4 ノード接続のクラスタでストレージアレイを単一クラスタノードから取り外すには、次の手順を使用します。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 取り外す予定のストレージアレイに関連付けられているすべてのデータベーステーブル、データサービス、ボリュームのバックアップを作成します。
- 2 切断する予定のノードで動作しているリソースグループとデバイスグループを判別します。

```
phys-schost# clresourcegroup status
phys-schost# cldevicegroup status
```

- 3 必要であれば、切断する予定のノードからすべてのリソースグループとデバイスグループを移動します。



**Caution (SPARC only)** - Oracle RAC ソフトウェアをクラスタで実行している場合、グループをノードから移動する前に、動作している Oracle RAC データベースのインスタンスを停止します。手順については、『Oracle Database Administration Guide』を参照してください。

```
phys-schost# clnode evacuate node
```

clnode evacuate コマンドは、すべてのデバイスグループを、指定ノードから次に優先されるノードに切り替えます。またこのコマンドは、指定ノード上の投票または非投票ノードから、次に優先される投票または非投票ノードへ、すべてのリソースグループを切り替えます。

- 4 デバイスグループを保守状態にします。  
デバイスグループを保守状態にする手順については、[251 ページの「ノードを保守状態にする」](#)を参照してください。

- 5 デバイスグループからノードを削除します。

raw ディスクを使用している場合は、`cldevicegroup` コマンドを使用して、デバイスグループを削除します。詳細については、[cldevicegroup\(1CL\)](#) のマニュアルページを参照してください。

- 6 **HASStoragePlus** リソースが含まれる各リソースグループで、リソースグループのノードリストからノードを削除します。

```
phys-schost# clresourcegroup remove-node -z zone -n node + | resourcegroup
node
```

ノードの名前。

`zone` リソースグループをマスターできる非投票ノードの名前。リソースグループを作成したときに非投票ノードを指定した場合にのみ、`zone` を指定します。

リソースグループのノードリストを変更する方法については、『[Oracle Solaris Cluster Data Services Planning and Administration Guide](#)』を参照してください。

---

注 - `clresourcegroup` を実行するときには、リソースタイプ、リソースグループ、およびリソースのプロパティ名には大文字と小文字の区別があります。

---

- 7 削除する予定のストレージアレイがノードに接続されている最後のストレージアレイである場合、当該ストレージアレイに接続されているハブまたはスイッチとノードの間にある光ケーブルを取り外します (そうでない場合、この手順は省略します)。
- 8 切断するノードからホストアダプタを削除する場合、ノードの電源を切ります。切断するノードからホストアダプタを削除する場合は、[手順 11](#)に進みます。
- 9 ノードからホストアダプタを削除します。  
ホストアダプタの削除手順については、ノード用ドキュメントを参照してください。
- 10 ブートが行われないようにして、ノードに電源を入れます。
- 11 **Oracle RAC** ソフトウェアがインストールされている場合、切断する予定のノードからそのパッケージを削除します。

```
phys-schost# pkgrm SUNWscum
```




---

**Caution (SPARC only)** – 切断したノードから Oracle RAC ソフトウェアを削除しない場合、そのノードをクラスタに導入し直すときにノードでパニックが発生し、データの可用性が失われる可能性があります。

---

- 12 クラスタモードでノードをブートします。



- SPARC ベースのシステム上で、次のコマンドを実行します。

```
ok boot
```

- x86 ベースのシステム上で、次のコマンドを実行します。

GRUB メニューが表示された時点で、適切な Oracle Solaris エントリを選択し、Enter キーを押します。次のような GRUB メニューが表示されます。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-----+
| Solaris 10 /sol_10_x86 |
| Solaris failsafe |
| |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

- 13 ノードの `/devices` と `/dev` エントリを更新して、デバイスの名前空間を更新します。

```
phys-schost# devfsadm -C
cldevice refresh
```

- 14 デバイスグループをオンラインに戻します。

デバイスグループをオンラインにする方法については、[253 ページの「ノードを保守状態から戻す」](#)を参照してください。

## ▼ エラーメッセージを修正する方法

クラスタノードの削除手順のいずれかを実行中に発生したエラーメッセージを修正するには、次の手順を実行します。

- 1 グローバルクラスタへのノードの再参加を試みます。この手順は、グローバルクラスタ上のみで実行します。

```
phys-schost# boot
```

- 2 ノードがクラスタに正常に再参加したかどうかを確認します。

- 再結合されていない場合は、[手順 b](#)に進みます。
- 再結合されている場合は、次の各手順を行なってノードをデバイスグループから削除します。

- a. ノードが正常にクラスタに再参加した場合は、残っているデバイスグループからノードを削除します。

[134 ページの「すべてのデバイスグループからノードを削除する方法」](#)の作業を行います。

- b. すべてのデバイスグループからノードを削除したあと、[255 ページの「クラスタノードから Oracle Solaris Cluster ソフトウェアをアンインストールする」](#)に戻り、その手順を繰り返します。
- 3 ノードがクラスタに再参加できなかった場合は、ノードの `/etc/cluster/ccr` ファイルを他の名前に変更します (たとえば、`ccr.old`)。  

```
mv /etc/cluster/ccr /etc/cluster/ccr.old
```
- 4 [255 ページの「クラスタノードから Oracle Solaris Cluster ソフトウェアをアンインストールする」](#)に戻り、その手順を繰り返します。

## クラスタの管理

---

この章では、グローバルクラスタやゾーンクラスタ全体に影響する管理手順について説明します。

- 235 ページの「クラスタの管理の概要」
- 267 ページの「ゾーンクラスタ管理タスクの実行」
- 275 ページの「トラブルシューティング」

クラスタへのノードの追加または削除に関する詳細は、第8章「ノードの追加と削除」を参照してください。

### クラスタの管理の概要

このセクションでは、グローバルクラスタやゾーンクラスタ全体の管理タスクを実行する方法を説明します。次の表に、これらの管理タスクと、関連する手順を示します。クラスタの管理タスクは通常は大域ゾーンで行います。ゾーンクラスタを管理するには、そのゾーンクラスタをホストするマシンが1台以上クラスタモードで起動している必要があります。すべてのゾーンクラスタノードが起動し動作している必要はありません。現在クラスタ外にあるノードがクラスタに再参加すると、構成の変更点が Oracle Solaris Cluster によって再現されます。

---

注-デフォルトでは、電源管理は無効になっているため、クラスタに干渉しません。単一ノードクラスタの電源管理を有効にすると、クラスタは引き続き動作していますが、数秒間使用できなくなる場合があります。電源管理機能はノードを停止しようとしませんが、停止されません。

---

この章での `phys-schost#` は、グローバルクラスタのプロンプトを表します。`clzonecluster` の対話型シェルプロンプトは `clzsc: schost>` です。

表 9-1 タスクリスト:クラスタの管理

タスク	手順
クラスタへのノードの追加または削除	第 8 章「ノードの追加と削除」
クラスタ名を変更	236 ページの「クラスタ名を変更する方法」
ノード ID およびそれらの対応するノード名の一覧の表示	238 ページの「ノード ID をノード名にマップする方法」
クラスタへの新しいノードの追加を許可または拒否	238 ページの「新しいクラスタノード認証で作業する方法」
NTP (Network Time Protocol) を使用してクラスタの時間を変更	240 ページの「クラスタの時刻をリセットする方法」
ノードを停止し、SPARC ベースのシステムでは OpenBoot PROM ok プロンプト、x86 ベースのシステムでは GRUB メニューで「Press any key to continue」というメッセージを表示	243 ページの「SPARC: ノードで OpenBoot PROM (OBP) を表示する方法」
プライベートホスト名の追加または変更	246 ページの「グローバルクラスタの非投票ノードにプライベートホスト名を追加する方法」 243 ページの「ノードのプライベートホスト名を変更する」
クラスタノードを保守状態に変更	251 ページの「ノードを保守状態にする」
ノード名の変更	249 ページの「ノード名を変更する」
クラスタノードを保守状態から復帰	253 ページの「ノードを保守状態から戻す」
クラスタノードからソフトウェアをアンインストール	255 ページの「クラスタノードから Oracle Solaris Cluster ソフトウェアをアンインストールする」
SNMP Event MIB の追加および管理	260 ページの「SNMP イベント MIB を有効にする」 264 ページの「SNMP ユーザーをノードに追加する」
各ノードの負荷制限の構成	266 ページの「ノードに負荷制限を構成する」
ゾーンクラスタの移動、アプリケーション用ゾーンクラスタの準備、ゾーンクラスタの削除	267 ページの「ゾーンクラスタ管理タスクの実行」

▼ クラスタ名を変更する方法

必要に応じて、初期インストール後にクラスタ名を変更できます。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 グローバルクラスタ内にある任意のノード上でスーパーユーザーになります。
- 2 **clsetup** ユーティリティを起動します。  

```
phys-schost# clsetup
```

メインメニューが表示されます。
- 3 クラスタ名を変更するには、クラスタその他のプロパティのオプションに対応する番号を入力します。  
「クラスタその他のプロパティ」メニューが表示されます。
- 4 メニューから選択を行なって、画面の指示に従います。
- 5 **Oracle Solaris Cluster** のサービスタグに新しいクラスタ名を反映させる場合は、既存の **Oracle Solaris Cluster** タグを削除してクラスタを再起動します。 **Oracle Solaris Cluster** サービスタグインスタンスを削除するには、クラスタ内のすべてのノードで次のサブステップを完了します。
  - a. すべてのサービスタグの一覧を表示します。  

```
phys-schost# stclient -x
```
  - b. **Oracle Solaris Cluster** サービスタグインスタンス番号を見つけて、次のコマンドを実行します。  

```
phys-schost# stclient -d -i service_tag_instance_number
```
  - c. クラスタ内のすべてのノードをリブートします。  

```
phys-schost# reboot
```

### 例 9-1 クラスタ名の変更

次の例に、新しいクラスタ名 `dromedary` へ変更するために、`clsetup` ユーティリティから生成される `cluster` コマンドを示します。

詳細は、[cluster\(1CL\)](#) および [clsetup\(1CL\)](#) のマニュアルページを参照してください。

```
phys-schost# cluster rename -c dromedary
```

## ▼ ノード ID をノード名にマップする方法

Oracle Solaris Cluster のインストール中、ノードにはそれぞれ一意のノード ID 番号が自動で割り当てられます。このノード ID 番号は、最初にクラスタに参加したときの順番でノードに割り当てられます。ノード ID 番号が割り当てられたあとでは、番号は変更できません。ノード ID 番号は、通常、エラーメッセージが発生したクラスタノードを識別するために、エラーメッセージで使用されます。この手順を使用し、ノード ID とノード名間のマッピングを判別します。

グローバルクラスタまたはゾーンクラスタ用の構成情報を表示するために、スーパーユーザーになる必要はありません。グローバルクラスタのノードから、このプロシージャの 1 ステップが実行されます。他のステップはゾーンクラスタノードから実行されます。

- 1 **clnode** コマンドを使用して、グローバルクラスタに対するクラスタ構成情報を一覧表示します。

```
phys-schost# clnode show | grep Node
```

詳細は、[clnode\(1CL\)](#) のマニュアルページを参照してください。

- 2 1 つのゾーンクラスタに対して、複数のノード ID を一覧表示することも可能です。ゾーンクラスタノードは、実行中のグローバルクラスタノードと同じノード ID を持っています。

```
phys-schost# zlogin sczone clnode -v | grep Node
```

### 例 9-2 ノード名のノード ID へのマップ

次の例は、グローバルクラスタに対するノード ID の割り当てを示しています。

```
phys-schost# clnode show | grep Node
=== Cluster Nodes ===
Node Name: phys-schost1
Node ID: 1
Node Name: phys-schost2
Node ID: 2
Node Name: phys-schost3
Node ID: 3
```

## ▼ 新しいクラスタノード認証で作業する方法

Oracle Solaris Cluster では、新しいノードをグローバルクラスタに追加できるようにするかどうかと、使用する認証の種類を指定できます。パブリックネットワーク上のクラスタに参加する新しいノードを許可したり、新しいノードがクラスタに参加することを拒否したり、クラスタに参加するノードを特定できます。新しいノードは、標準 UNIX または Diffie-Hellman (DES) 認証を使用し、認証することができま

す。DES 認証を使用して認証する場合、ノードが参加するには、すべての必要な暗号化鍵を構成する必要があります。詳細は、[keyserv\(1M\)](#) と [publickey\(4\)](#) のマニュアルページを参照してください。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 グローバルクラスタ内にある任意のノード上でスーパーユーザーになります。
- 2 **clsetup** ユーティリティを起動します。  
`phys-schost# clsetup`  
メインメニューが表示されます。
- 3 クラスタ認証の作業を行うには、新規ノードのオプションに対応する番号を入力します。  
「新規ノード」メニューが表示されます。
- 4 メニューから選択を行なって、画面の指示に従います。

#### 例 9-3 新しいマシンがグローバルクラスタに追加されないようにする

`clsetup` ユーティリティにより、`claccess` コマンドを生成します。次の例は、新しいマシンがクラスタに追加されないようにする `claccess` コマンドを示しています。

```
phys-schost# claccess deny -h hostname
```

#### 例 9-4 すべての新しいマシンがグローバルクラスに追加されることを許可する

`clsetup` ユーティリティにより、`claccess` コマンドを生成します。次の例は、すべての新しいマシンをクラスタに追加できるようにする `claccess` コマンドを示しています。

```
phys-schost# claccess allow-all
```

#### 例 9-5 グローバルクラスタに追加される新しいマシンを指定する

`clsetup` ユーティリティにより、`claccess` コマンドを生成します。次の例は、1 台の新しいマシンをクラスタに追加できるようにする `claccess` コマンドを示しています。

```
phys-schost# claccess allow -h hostname
```

#### 例 9-6 認証を標準 UNIX に設定する

clsetup ユーティリティーにより、claccess コマンドを生成します。次の例は、クラスタに参加している新規ノードの標準 UNIX 認証に対し、リセットを行う claccess コマンドを示しています。

```
phys-schost# claccess set -p protocol=sys
```

#### 例 9-7 認証を DES に設定する

clsetup ユーティリティーにより、claccess コマンドを生成します。次の例は、クラスタに参加している新規ノードの DES 認証を使用する claccess コマンドを示しています。

```
phys-schost# claccess set -p protocol=des
```

DES 認証を使用する場合、クラスタにノードが参加するには、すべての必要な暗号化鍵を構成します。詳細は、[keyserv\(1M\)](#) と [publickey\(4\)](#) のマニュアルページを参照してください。

## ▼ クラスタの時刻をリセットする方法

Oracle Solaris Cluster ソフトウェアは、NTP (Network Time Protocol) を使用して、クラスタノード間で時間を同期させています。グローバルクラスタの時刻の調整は、ノードが時刻を同期するときに、必要に応じて自動的に行われます。詳細は、『[Oracle Solaris Cluster Concepts Guide](#)』および <http://docs.oracle.com/cd/E19065-01/servers.10k/> の Network Time Protocol のユーザーズガイドを参照してください。



注意 - NTP を使用する場合、クラスタの稼動中はクラスタの時刻を調整しないでください。date、rdate、または svcadm コマンドを使用した対話形式で、または cron スクリプト内で、時間を調整しないでください。詳細については、[date\(1\)](#)、[rdate\(1M\)](#)、[xntpd\(1M\)](#)、[svcadm\(1M\)](#)、または [cron\(1M\)](#) のマニュアルページを参照してください。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。



- 1 グローバルクラスタ内にある任意のノード上でスーパーユーザーになります。
- 2 グローバルクラスタを停止します。  
phys-schost# **cluster shutdown -g0 -y -i 0**
- 3 SPARC ベースのシステムではノードが **ok** プロンプトを表示し、x86 ベースのシステムでは **GRUB** メニューで「**Press any key to continue**」というメッセージが表示されていることを確認します。

- 4 非クラスタモードでノードをブートします。

- SPARC ベースのシステム上で、次のコマンドを実行します。

```
ok boot -x
```

- x86 ベースのシステム上で、次のコマンドを実行します。

```
shutdown -g -y -i0
```

```
Press any key to continue
```

- a. **GRUB** メニューで矢印キーを使用して該当する **Oracle Solaris** エントリを選択し、**e** と入力してコマンドを編集します。

次のような **GRUB** メニューが表示されます。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
```

```
+-----+
| Solaris 10 /sol_10_x86 |
| Solaris failsafe |
| |
+-----+
```

```
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

**GRUB** ベースのブートの詳細については、『[Oracle Solaris の管理: 基本管理](#)』の「[GRUB を使用して x86 システムをブートする \(タスクマップ\)](#)」を参照してください。

- b. ブートパラメータの画面で、矢印キーを使用してカーネルエントリを選択し、**e** を入力してエントリを編集します。

次のような **GRUB** ブートパラメータの画面が表示されます。

```
GNU GRUB version 0.95 (615K lower / 2095552K upper memory)
```

```
+-----+
| root (hd0,0,a) |
| kernel /platform/i86pc/multiboot |
| module /platform/i86pc/boot_archive |
+-----+
```

```
Use the ^ and v keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.
```

- c. コマンドに **-x** を追加して、システムを非クラスタモードでブートするように指定します。

```
[Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time exits.]
```

```
grub edit> kernel /platform/i86pc/multiboot -x
```

- d. **Enter** キーを押して変更を受け入れ、ブートパラメータの画面に戻ります。  
画面には編集されたコマンドが表示されます。

```
GNU GRUB version 0.95 (615K lower / 2095552K upper memory)
```

```
+-----+
| root (hd0,0,a) |
| kernel /platform/i86pc/multiboot -x |
| module /platform/i86pc/boot_archive |
+-----+
```

```
Use the ^ and v keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.-
```

- e. **b** と入力して、ノードを非クラスタモードでブートします。

---

注-カーネルブートパラメータコマンドへのこの変更は、システムをブートすると無効になります。次にノードをリブートする際には、ノードはクラスタモードでブートします。クラスタモードではなく、非クラスタモードでブートするには、これらの手順を再度実行して、カーネルブートパラメータコマンドに **-x** オプションを追加します。

---

- 5 単一のノードで、**date** コマンドを実行して時刻を設定します。

```
phys-schost# date HHMM.SS
```

- 6 ほかのマシンで **rddate** コマンドを実行して、時間をそのノードと同期させます。

```
phys-schost# rdate hostname
```

詳細は、[rddate\(1M\)](#) のマニュアルページを参照してください。

- 7 各ノードを起動し、クラスタを再起動します。

```
phys-schost# reboot
```

- 8 すべてのクラスタノードで変更が行われたことを確認します。

各ノードで、**date** コマンドを実行します。

```
phys-schost# date
```

## ▼ SPARC: ノードで OpenBoot PROM (OBP) を表示する方法

OpenBoot™ PROM 設定を構成または変更する必要がある場合は、この手順を使用します。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 停止するノード上でコンソールに接続します。

```
telnet tc_name tc_port_number
```

**tc\_name** 端末集配信装置 (コンセントレータ) の名前を指定します。

**tc\_port\_number** 端末集配信装置のポート番号を指定します。ポート番号は構成に依存します。通常、ポート 2 (5002) とポート 3 (5003) は、サイトで最初に設置されたクラスタで使用されています。

- 2 **clnode evacuate** コマンドを使用してから、**shutdown** コマンドを使用することで、クラスタノードを正常に停止します。**clnode evacuate** コマンドは、すべてのデバイスグループを、指定ノードから次に優先されるノードに切り替えます。またこのコマンドは、グローバルクラスタの指定した投票ノードまたは非投票ノードから、次に優先される投票ノードまたは非投票ノードへ、すべてのリソースグループを切り替えます。

```
phys-schost# clnode evacuate node
shutdown -g0 -y
```



注意 - クラスタコンソールで **send brk** を使用して、クラスタノードをシャットダウンしないでください。

- 3 OBP コマンドを実行します。

## ▼ ノードのプライベートホスト名を変更する

インストール完了後、クラスタノードのプライベートホスト名を変更するには、次の手順に従います。

デフォルトのプライベートホスト名は、クラスタの初期インストール時に割り当てられます。デフォルトのプライベートホスト名の形式は、**clusternode<nodeid>-priv**

です (clusternode3-priv など)。名前がすでにドメイン内で使用されている場合にかぎり、プライベートホスト名を変更します。



注意 - 新しいプライベートホスト名には IP アドレスを割り当てないでください。クラスタソフトウェアがそれらを割り当てます。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 クラスタ内のすべてのノード上で、プライベートホスト名をキャッシュする可能性があるデータサービスリソースやアプリケーションをすべて無効にします。

```
phys-schost# clresource disable resource[,...]
```

無効にするアプリケーションには次のようなものがあります。

- HA-DNS と HA-NFS サービス (構成している場合)
- プライベートホスト名を使用するためにカスタム構成されたアプリケーション
- クライアントがプライベートインターコネクト経由で使用しているアプリケーション

clresource コマンドの使用については、[clresource\(1CL\)](#) のマニュアルページおよび『[Oracle Solaris Cluster Data Services Planning and Administration Guide](#)』を参照してください。

- 2 NTP 構成ファイルが、変更しようとしているプライベートホスト名を参照している場合、クラスタの各ノード上で NTP (Network Time Protocol) デーモンを停止します。

svcadm コマンドを使用して、NTP (Network Time Protocol) デーモンを停止します。NTP デーモンについての詳細は、[svcadm\(1M\)](#) のマニュアルページを参照してください。

```
phys-schost# svcadm disable ntp
```

- 3 [clsetup\(1CL\)](#) ユーティリティーを実行して、適切なノードのプライベートホスト名を変更します。

ユーティリティーは、クラスタ内の 1 つのノードからのみ実行します。

注 - 新しいプライベートホスト名を選択するときには、その名前がクラスタノード内で一意であることを確認してください。

- 4 プライベートホスト名のオプションに対応する番号を入力します。

- 5 プライベートホスト名を変更するためのオプションに対応する番号を入力します。  
表示される質問に答えます。変更しようとしているプライベートホスト名のノード名 (`clusternode<nodeid> -priv`) および新しいプライベートホスト名を入力してください。
- 6 ネームサービスキャッシュをフラッシュします。  
クラスタの各ノードで次の手順を実行します。フラッシュすることによって、クラスタアプリケーションとデータサービスが古いプライベートホスト名にアクセスしないようにします。  

```
phys-schost# nscd -i hosts
```
- 7 NTP 構成ファイルでプライベートホスト名を変更した場合は、各ノードの NTP 構成ファイル (`ntp.conf` または `ntp.conf.cluster`) を更新します。
  - a. 任意のエディタを使用してください。  
この手順をインストール時に行う場合は、構成するノードの名前を削除する必要があります。デフォルトテンプレートは 16 のノードで事前構成されています。通常、`ntp.conf.cluster` ファイルは各クラスタノードで同一です。
  - b. すべてのクラスタノードから新しいプライベートホスト名に **ping** を実行できることを確認します。
  - c. NTP デーモンを再起動します。  
クラスタの各ノードで次の手順を実行します。  
NTP デーモンを再起動するには、`svcadm` コマンドを使用します。  

```
svcadm enable ntp
```
- 8 **手順 1** で無効にしたデータサービスリソースとそのほかのアプリケーションをすべて有効にします。  

```
phys-schost# clresource enable resource[,...]
```

  
`clresource` コマンドの使用については、`clresource(1CL)` のマニュアルページおよび『[Oracle Solaris Cluster Data Services Planning and Administration Guide](#)』を参照してください。

## 例 9-8 プライベートホスト名を変更する

次に、ノード `phys-schost-2` 上のプライベートホスト名 `clusternode2-priv` を `clusternode4-priv` に変更する例を示します。

```
[Disable all applications and data services as necessary.]
phys-schost-1# /etc/init.d/xntpd stop
phys-schost-1# clnode show | grep node
```

```
...
private hostname: clusternode1-priv
private hostname: clusternode2-priv
private hostname: clusternode3-priv
...
phys-schost-1# clsetup
phys-schost-1# nscd -i hosts
phys-schost-1# vi /etc/inet/ntp.conf
...
peer clusternode1-priv
peer clusternode4-priv
peer clusternode3-priv
phys-schost-1# ping clusternode4-priv
phys-schost-1# /etc/init.d/xntpd start
[Enable all applications and data services disabled at the beginning of the procedure.]
```

## ▼ グローバルクラスタの非投票ノードにプライベートホスト名を追加する方法

インストール完了後にグローバルクラスタの非投票ノードにプライベートホスト名を追加するには、この手順を使用します。この章の手順の `phys-schost#` は、グローバルクラスタプロンプトを表します。この手順は、グローバルクラスタ上のみで実行します。

- 1 **clsetup(1CL)** ユーティリティーを実行して、適切なゾーンでプライベートホスト名を追加します。

```
phys-schost# clsetup
```

- 2 プライベートホスト名のオプションに対応する番号を入力し、**Return** キーを押します。
- 3 ゾーンのプライベートホスト名を追加するためのオプションに対応する番号を入力し、**Return** キーを押します。

表示される質問に答えます。グローバルクラスタの非投票ノードには、デフォルトのプライベートホスト名はありません。ホスト名を入力する必要があります。

## ▼ グローバルクラスタの非投票ノードでプライベートホスト名を変更する方法

インストール完了後に非投票ノードのプライベートホスト名を変更するには、この手順を使用します。

プライベートホスト名は、クラスタの初期インストール時に割り当てられます。プライベートホスト名の形式は、`clusternode<nodeid>-priv` です (`clusternode3-priv` など)。名前がすでにドメイン内で使用されている場合にかぎり、プライベートホスト名を変更します。



注意-新しいプライベートホスト名にはIPアドレスを割り当てないでください。クラスタソフトウェアがそれらを割り当てます。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 グローバルクラスタ内のすべてのノード上で、プライベートホスト名をキャッシュする可能性があるデータサービスリソースやほかのアプリケーションをすべて無効にします。

```
phys-schost# clresource disable resource1, resource2
```

無効にするアプリケーションには次のようなものがあります。

- HA-DNS と HA-NFS サービス (構成している場合)
- プライベートホスト名を使用するためにカスタム構成されたアプリケーション
- クライアントがプライベートインターコネクト経由で使用しているアプリケーション

`clresource` コマンドの使用については、`clresource(1CL)` のマニュアルページおよび『[Oracle Solaris Cluster Data Services Planning and Administration Guide](#)』を参照してください。

- 2 `clsetup` ユーティリティを実行して、グローバルクラスタ上の適切な非投票ノードのプライベートホスト名を変更します。

```
phys-schost# clsetup
```

この手順は、クラスタ内の1つのノードからのみ実行する必要があります。詳細は、`clsetup(1CL)` のマニュアルページを参照してください。

注-新しいプライベートホスト名を選択するときには、その名前がクラスタ内で一意であることを確認してください。

- 3 プライベートホスト名のオプションに対応する番号を入力し、**Return** キーを押します。

- 4 ゾーンのプライベートホスト名を追加するためのオプションに対応する番号を入力し、**Return** キーを押します。  
グローバルクラスタの非投票ノードには、デフォルトのプライベートホスト名は存在しません。ホスト名を入力する必要があります。
- 5 ゾーンのプライベートホスト名を変更するためのオプションに対応する番号を入力します。  
表示される質問に答えます。プライベートホスト名を変更する非投票ノードの名前 (clusternode<nodeid> -priv) と新しいプライベートホスト名の入力を求められます。
- 6 ネームサービスキャッシュをフラッシュします。  
クラスタの各ノードで次の手順を実行します。フラッシュすることによって、クラスタアプリケーションとデータサービスが古いプライベートホスト名にアクセスしないようにします。  

```
phys-schost# nscd -i hosts
```
- 7 **手順1**で無効にしたデータサービスリソースとほかのアプリケーションをすべて有効にします。

## ▼ グローバルクラスタの非投票ノードのプライベートホスト名を削除する方法

グローバルクラスタの非投票ノードのプライベートホスト名を削除するには、この手順を使用します。この手順は、グローバルクラスタ上のみで実行します。

- 1 **clsetup** ユーティリティーを実行して、適切なゾーン上のプライベートホスト名を削除します。  
詳細は、[clsetup\(1CL\)](#)のマニュアルページを参照してください。
- 2 ゾーンのプライベートホスト名のオプションに対応する番号を入力します。
- 3 ゾーンのプライベートホスト名を削除するためのオプションに対応する番号を入力します。
- 4 削除する非投票ノードのプライベートホスト名の名前を入力します。



## ▼ ノード名を変更する

Oracle Solaris Cluster 構成の一部であるノードの名前を変更できます。ノード名を変更する前に Oracle Solaris ホスト名を変更する必要があります。ノード名を変更するには、`clnode rename` コマンドを使用します。

次の説明は、グローバルクラスタで動作しているすべてのアプリケーションに該当します。

- 1 グローバルクラスタでは、スーパーユーザーになるか、`solaris.cluster.modify` RBAC の承認を提供する役割になります。
- 2 Oracle Solaris 構成のパートナーシップにある Oracle Solaris Cluster Geographic Edition クラスタのノードの名前を変更する場合は、追加の手順を行う必要があります。名前変更手順を行っているクラスタが保護グループのプライマリクラスタで、保護グループのアプリケーションをオンラインにしておく場合は、名前変更手順を行っている間、保護グループをセカンダリクラスタに切り替えることができます。Geographic Edition クラスタおよびノードの詳細は、『[Oracle Solaris Cluster Geographic Edition System Administration Guide](#)』の第 5 章「[Administering Cluster Partnerships](#)」を参照してください。
- 3 『[Solaris のシステム管理 \(上級編\)](#)』の「[システムのホスト名の変更方法](#)」の手順を完了して (ただし、手順の最後にあるリポートは実行しない)、Oracle Solaris のホスト名を変更します。代わりに、この手順を完了したあと、クラスタの停止を実行します。
- 4 すべてのクラスタノードを非クラスタモードでブートします。  

```
ok> boot -x
```
- 5 Oracle Solaris のホスト名を変更したノード上で、非クラスタモードでノードの名前を変更し、名前を変更したホストごとに `cmd` コマンドを実行します。1 回につき 1 つのノード名を変更します。  

```
clnode rename -n newnodename oldnodename
```
- 6 クラスタで実行されるアプリケーション内の以前のホスト名への既存の参照を更新します。
- 7 コマンドメッセージとログファイルをチェックして、ノード名が変更されたことを確認します。
- 8 すべてのノードをクラスタモードでリブートします。  

```
sync;sync;sync;/etc/reboot
```

- 9 ノードに新しい名前が表示されていることを確認します。

```
clnode status -v
```

- 10 **GeographicEdition** クラスタノード上で名前を変更していて、名前が変更されたノードを含むクラスタのパートナークラスタが引き続き以前のノード名を参照している場合、保護グループの同期ステータスはエラーとして表示されます。**geopg update <pg>** を使用して、名前が変更されたノードが含まれているパートナークラスタのいずれかのノードから保護グループを更新する必要があります。その手順の完了後、**geopg start -e global <pg>** コマンドを実行します。あとで、名前が変更されたノードのクラスタに保護グループを再び切り替えることができます。
- 11 論理ホスト名リソースの **hostnamelist** プロパティを変更することを選択できます。このオプションな手順については、[250 ページの「既存の Oracle Solaris Cluster の論理ホスト名リソースで使用されている論理ホスト名を変更する」](#)を参照してください。

## ▼ 既存の **Oracle Solaris Cluster** の論理ホスト名リソースで使用されている論理ホスト名を変更する

[249 ページの「ノード名を変更する」](#)の手順に従ってノード名を変更する前または変更したあとに、論理ホスト名リソースの **hostnamelist** プロパティを変更することもできます。この手順はオプションです。

- 1 グローバルクラスタでは、スーパーユーザーになるか、**solaris.cluster.modify RBAC** の承認を提供する役割になります。
- 2 必要に応じて、既存の **Oracle Solaris Cluster** の論理ホスト名リソースのいずれかで使用されている論理ホスト名を変更できます。  
次の手順は、新しい論理ホスト名と連動するように **apache-lh-res** リソースを構成する方法を示したもので、クラスタモードで実行する必要があります。
  - a. クラスタモードで、論理ホスト名を含む **Apache** リソースグループをオフラインにします。  

```
clrg offline apache-rg
```
  - b. **Apache** 論理ホスト名リソースを無効にします。  

```
clrs disable appache-lh-res
```
  - c. 新しいホスト名リストを指定します。  

```
clrs set -p HostnameList=test-2 apache-lh-res
```

- d. `hostnameList` プロパティの以前のエントリに対するアプリケーションの参照を、新しいエントリを参照するように変更します。
- e. 新しい **Apache** 論理ホスト名リソースを有効にします。  
`# clrs enable apache-lh-res`
- f. **Apache** リソースグループをオンラインにします。  
`# clrg online -eM apache-rg`
- g. 次のコマンドを実行してクライアントをチェックし、アプリケーションが正しく起動したことを確認します。  
`# clrs status apache-rs`

## ▼ ノードを保守状態にする

サービスからグローバルクラスタノードを長時間外す場合は、そのノードを保守状態にします。保守状態のノードは、サービス対象中に定足数確立の投票に参加しません。ノードを保守状態にするには、`clnode evacuate` および `cluster shutdown` コマンドを使用して、ノードをシャットダウンしておく必要があります。詳細は、[clnode\(1CL\)](#) および [cluster\(1CL\)](#) のマニュアルページを参照してください。

---

注 - ノードを1つだけ停止する場合は、Oracle Solaris の `shutdown` コマンドを使用します。クラスタ全体を停止する場合にだけ、`cluster shutdown` コマンドを使用します。

---

クラスタノードが停止されて保守状態になると、そのノードのポートで構成されるすべての定足数デバイスの、定足数投票数 (quorum vote count) が1つ減ります。このノードが保守状態から移動してオンラインに戻されると、ノードおよび定足数デバイスの投票数は1つ増えます。

`clquorum disable` コマンドを使用して、クラスタノードを保守状態にします。詳細は、[clquorum\(1CL\)](#) のマニュアルページを参照してください。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 保守状態にするグローバルクラスタノード上で、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.modify` を提供する役割になります。

- 2 ノードからすべてのリソースグループおよびデバイスグループを退避させます。**clnode evacuate** コマンドは、すべての非投票ノードを含むすべてのリソースグループおよびデバイスグループを、指定ノードから次に優先されるノードに切り替えます。

```
phys-schost# clnode evacuate node
```

- 3 退避させたノードをシャットダウンします。

```
phys-schost# shutdown -g0 -y -i 0
```

- 4 クラスタ内の別のノード上で、スーパーユーザーになるか、RBACの承認 **solaris.cluster.modify** を提供する役割になり、[手順3](#)でシャットダウンしたノードを保守状態にします。

```
phys-schost# clquorum disable node
```

node                      保守モードにするノードの名前を指定します。

- 5 グローバルクラスタノードが保守状態にあることを確認します。

```
phys-schost# clquorum status node
```

保守状態にしたノードのStatusはofflineで、そのPresentとPossibleの定足数投票数は0(ゼロ)である必要があります。

## 例9-9 グローバルクラスタノードを保守状態にする

次に、クラスタノードを保守状態にして、その結果を確認する例を示します。clnode status の出力では、phys-schost-1 のノードのNode votes は0(ゼロ)で、そのステータスはOfflineです。Quorum Summary では、投票数も減っているはずですが。構成によって異なりますが、Quorum Votes by Device の出力では、いくつかの定足数ディスクデバイスもオフラインである可能性があります。

[On the node to be put into maintenance state:]

```
phys-schost-1# clnode evacuate phys-schost-1
```

```
phys-schost-1# shutdown -g0 -y -i0
```

[On another node in the cluster:]

```
phys-schost-2# clquorum disable phys-schost-1
```

```
phys-schost-2# clquorum status phys-schost-1
```

```
-- Quorum Votes by Node --
```

Node Name	Present	Possible	Status
-----	-----	-----	-----
phys-schost-1	0	0	Offline
phys-schost-2	1	1	Online
phys-schost-3	1	1	Online

参照 ノードをオンライン状態に戻す方法については、[253 ページ](#)の「ノードを保守状態から戻す」を参照してください。

## ▼ ノードを保守状態から戻す

次の手順を使用して、グローバルクラスタノードをオンラインに戻し、定足数投票数をリセットしてデフォルト設定に戻します。クラスタノードのデフォルトの投票数は1です。定足数デバイスのデフォルトの投票数は $N-1$ です。 $N$ は、投票数が0以外で、定足数デバイスが構成されているポートを持つノードの数を示します。

ノードが保守状態になると、そのノードの投票数は1つ減ります。また、このノードのポートに構成されているすべての定足数デバイスの投票数も(1つ)減ります。投票数がリセットされ、ノードが保守状態から戻されると、ノードの投票数と定足数デバイスの投票数の両方が1つ増えます。

保守状態にしたグローバルクラスタノードを保守状態から戻した場合は、必ずこの手順を実行してください。



注意 - `globaldev` または `node` オプションのどちらも指定しない場合、定足数投票数はクラスタ全体でリセットされます。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 グローバルクラスタの、保守状態のノード以外の任意のノード上で、スーパーユーザーになるか、RBACの承認 `solaris.cluster.modify` を提供する役割になります。
- 2 グローバルクラスタ構成内にあるノードの数に応じて、次の手順のいずれかを実行します。
  - クラスタ構成内に2つのノードがある場合は、[手順4](#)に進みます。
  - クラスタ構成内に3つ以上のノードがある場合は、[手順3](#)に進みます。
- 3 保守状態から解除するノードに定足数デバイスがある場合は、保守状態にあるノード以外のノードからクラスタ定足数のカウントをリセットします。

保守状態ではないノードの定足数投票数をリセットするのは、そのノードをリブートする前である必要があります。そうしないと、定足数の確立を待機中にハングアップすることがあります。

```
phys-schost# clquorum reset
```

reset                      定足数をリセットする変更フラグです。

- 4 保守状態を解除するノードをブートします。
- 5 定足数投票数を確認します。

phys-schost# **clquorum status**

保守状態を解除したノードの状態はonlineであり、PresentとPossibleの定足数投票数は適切な値である必要があります。

例 9-10 クラスタノードの保守状態を解除して、定足数投票数をリセットする

次に、クラスタノードの定足数投票数をリセットして、その定足数デバイスをデフォルトに戻し、その結果を確認する例を示します。cluster status の出力では、phys-schost-1 のNode votes は1で、そのステータスはonlineです。Quorum Summary では投票数も増加しているはずです。

phys-schost-2# **clquorum reset**

- SPARC ベースのシステム上で、次のコマンドを実行します。

ok **boot**

- x86 ベースのシステム上で、次のコマンドを実行します。

GRUB メニューが表示された時点で、適切な Oracle Solaris エントリを選択し、Enter キーを押します。次のような GRUB メニューが表示されます。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-----+
| Solaris 10 /sol_10_x86 |
| Solaris failsafe |
| |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

phys-schost-1# **clquorum status**

--- Quorum Votes Summary ---

Needed	Present	Possible
-----	-----	-----
4	6	6

--- Quorum Votes by Node ---

Node Name	Present	Possible	Status
-----	-----	-----	-----
phys-schost-2	1	1	Online
phys-schost-3	1	1	Online

--- Quorum Votes by Device ---

Device Name	Present	Possible	Status
-----	-----	-----	-----
/dev/did/rdsk/d3s2	1	1	Online
/dev/did/rdsk/d17s2	0	1	Online
/dev/did/rdsk/d31s2	1	1	Online

## ▼ クラスタノードから Oracle Solaris Cluster ソフトウェアをアンインストールする

完全に確立されているクラスタ構成からグローバルクラスタノードを切断する前に、この手順を実行して、グローバルクラスタノードから Oracle Solaris Cluster ソフトウェアをアンインストールします。この手順では、クラスタに存在する最後のノードからソフトウェアをアンインストールできます。

注-クラスタにまだ参加していない、あるいはまだインストールモードであるノードから Oracle Solaris Cluster ソフトウェアをアンインストールする場合、この手順を使用してはいけません。その代わりに、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』の、Oracle Solaris Cluster ソフトウェアを正しくアンインストールし、インストールの問題を修正する方法の説明に進みます。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 タスクマップにあるクラスタノードを削除するための前提タスクがすべて完了していることを確認します。

[表 8-2](#) を参照してください。

注- この手順を続ける前に、`clnode remove` を使用してクラスタ構成からノードを削除します。

- 2 アンインストールするグローバルクラスタノード以外のグローバルクラスタのアクティブなメンバー上で、スーパーユーザーになります。この手順はグローバルクラスタノードから実行します。

- 3 アクティブなクラスタメンバーから、アンインストールするノードを、クラスタのノード認証リストに追加します。

```
phys-schost# claccess allow -h hostname
```

-h ノードの認証リストに追加するノードの名前を指定します。

あるいは、`clsetup` ユーティリティも使用できます。手順については、`clsetup(1CL)` のマニュアルページと [220 ページ](#) の「既存のクラスタにノードを追加する方法」を参照してください。

- 4 アンインストールするノードでスーパーユーザーになります。

- 5 ゾーンクラスタがある場合は、アンインストールします。

```
phys-schost# clzonecluster uninstall -F zoneclustername
```

具体的な手順については、[270 ページ](#) の「ゾーンクラスタを削除する」を参照してください。

- 6 ノードにグローバルデバイス名前空間用の専用パーティションがある場合、グローバルクラスタノードを非クラスタモードでリブートします。

- SPARC ベースのシステム上で、次のコマンドを実行します。

```
shutdown -g0 -y -i0ok boot -x
```

- x86 ベースのシステム上で、次のコマンドを実行します。

```
shutdown -g0 -y -i0
```

```
...
```

```
<<< Current Boot Parameters >>>
```

```
Boot path: /pci@0,0/pci8086,2545@3/pci8086,1460@1d/pci8086,341a@7,1/
```

```
sd@0,0:a
```

```
Boot args:
```

```
Type b [file-name] [boot-flags] <ENTER> to boot with options
or i <ENTER> to enter boot interpreter
or <ENTER> to boot with defaults
```

```
<<< timeout in 5 seconds >>>
```

```
Select (b)oot or (i)nterpreter: b -x
```

- 7 `/etc/vfstab` ファイルから、グローバルにマウントされるすべてのファイルシステムエントリを削除します。ただし、`/global/.devices` グローバルマウントを除きます。



- 8 このノードに **Oracle Solaris Cluster** ソフトウェアを再インストールする予定の場合、**Sun Java Enterprise System (Java ES)** 製品レジストリから **Oracle Solaris Cluster** エントリを削除します。

Java ES 製品レジストリに Oracle Solaris Cluster ソフトウェアがインストールされたというレコードが含まれている場合、Java ES インストーラは Oracle Solaris Cluster コンポーネントをグレー表示し、再インストールを許可しません。

- a. **Java ES** アンインストーラを起動します。

次のコマンドを実行します。*ver* は、Oracle Solaris Cluster ソフトウェアをインストールした Java ES ディストリビューションのバージョンです。

```
/var/sadm/prod/SUNWentsys ver /uninstall
```

- b. プロンプトに従って、アンインストールする **Oracle Solaris Cluster** を選択します。

`uninstall` コマンドの使用に関する詳細については、『[Sun Java Enterprise System 5 Update 1 Installation Guide for UNIX](#)』の第 8 章「[Uninstalling](#)」を参照してください。

- 9 このクラスタ上で **Oracle Solaris Cluster** ソフトウェアを再インストールしない場合は、ほかのクラスタデバイスからトランスポートケーブルとトランスポートスイッチを切断します (存在する場合)。

- a. アンインストールしたノードが、並列 **SCSI** インタフェースを使用する記憶装置デバイスに接続されている場合は、トランスポートケーブルを切り離した後で、この記憶装置デバイスのオープン **SCSI** コネクタに **SCSI** ターミネータを取り付ける必要があります。

アンインストールしたノードが、Fibre Channel インタフェースを使用する記憶装置デバイスに接続されている場合は、終端処理は必要ありません。

- b. 切断の手順については、ホストアダプタおよびサーバーに付属しているドキュメントに従います。

---

ヒント-ループバックファイルインタフェース (lofi) デバイスを使っている場合、Java ES のアンインストーラは、`/globaldevices` という名前の lofi ファイルを自動的に削除します。lofi へのグローバルデバイス名前空間の移行の詳細については、[126 ページの「グローバルデバイス名前空間を移行する」](#)を参照してください。

---

## ノードのアンインストールのトラブルシューティング

ここでは、`clnode remove` コマンドを実行したときに出力される可能性があるエラーメッセージとその対処方法について説明します。

### 削除されないクラスタファイルシステムのエントリ

次のエラーメッセージは、削除したグローバルクラスタノードに、`vfstab` ファイルから参照されているクラスタファイルシステムがまだあることを示しています。

```
Verifying that no unexpected global mounts remain in /etc/vfstab ... failed
clnode: global-mount1 is still configured as a global mount.
clnode: global-mount1 is still configured as a global mount.
clnode: /global/dg1 is still configured as a global mount.
```

```
clnode: It is not safe to uninstall with these outstanding errors.
clnode: Refer to the documentation for complete uninstall instructions.
clnode: Uninstall failed.
```

このエラーを修正するためには、[255 ページの「クラスタノードから Oracle Solaris Cluster ソフトウェアをアンインストールする」](#)に戻って、その手順を繰り返す必要があります。[手順 7](#) コマンドを再度実行する前に、この Step 7 が正しく行われているか確認してください。

### デバイスグループに削除されていないリストがある場合

次のエラーメッセージは、削除したノードが依然としてデバイスグループにリストされていることを示しています。

```
Verifying that no device services still reference this node ... failed
clnode: This node is still configured to host device service "
service".
clnode: This node is still configured to host device service "
service2".
clnode: This node is still configured to host device service "
service3".
clnode: This node is still configured to host device service "
dg1".
```

```
clnode: It is not safe to uninstall with these outstanding errors.
clnode: Refer to the documentation for complete uninstall instructions.
clnode: Uninstall failed.
```

### アンインストールスクリプトがない

削除しようとしている Sun Cluster または Oracle Solaris Cluster ソフトウェアのインストールまたはアップグレードに `installer` プログラムを使用しなかった場合、そのソ

ソフトウェアバージョンで使用するアンインストールスクリプトは存在しません。その代わり、次の手順を実行してソフトウェアをアンインストールします。

## ▼ アンインストールスクリプトなしで **Sun Cluster 3.1** および **3.2** ソフトウェアをアンインストールする方法

- 1 スーパーユーザーになるか、RBAC 承認 `solaris.cluster.modify` を提供する役割になります。

- 2 **Sun Cluster** パッケージに関連付けられていないディレクトリに変更します。

```
cd /directory
```

- 3 ノードから **Sun Cluster** ソフトウェアをアンインストールします。

```
scinstall -r
```

- 4 `productregistry` ファイルの名前を変更し、将来ありえるソフトウェアの再インストールが可能のようにします。

```
mv /var/sadm/install/productregistry /var/sadm/install/productregistry.sav
```

## Oracle Solaris Cluster SNMP イベント MIB の作成、設定、および管理

ここでは、SNMP イベント管理情報ベース (MIB) を作成、設定、および管理する方法を説明します。またこのセクションでは、Oracle Solaris Cluster SNMP イベント MIB を有効化、無効化、および変更する方法も説明します。

Oracle Solaris Cluster ソフトウェアでは現在、イベント MIB という MIB を 1 つサポートしています。SNMP マネージャーソフトウェアがクラスタイベントをリアルタイムでトラップします。有効な場合、SNMP マネージャーはトラップ通知を `clsnmphost` コマンドによって定義されているすべてのホストに自動的に送信します。MIB には、最新の 50 イベントの読み取り専用のテーブルが保持されます。クラスタは多数の通知を生成するので、重要度が `warning` 以上のイベントだけがトラップ通知として送信されます。この情報は、リポートが実行されると消失します。

SNMP イベント MIB は、`sun-cluster-event-mib.mib` ファイルで定義されており、`/usr/cluster/lib/mib` ディレクトリにあります。この定義を使用して、SNMP トラップ情報を解釈できます。

イベント SNMP モジュールのデフォルトのポート番号は 11161 で、SNMP トラップのデフォルトのポートは 11162 です。これらのポート番号は、共通エージェントコンテ

ナのプロパティファイル (/etc/cacao/instances/default/private/cacao.properties) を変更することによって変更できます。

Oracle Solaris Cluster SNMP イベント MIB の作成、設定、および管理には、次のタスクが含まれます。

表 9-2 タスクマップ: Oracle Solaris Cluster SNMP イベント MIB の作成、設定、および管理

タスク	手順
SNMP イベント MIB の有効化	260 ページの「SNMP イベント MIB を有効にする」
SNMP イベント MIB の無効化	261 ページの「SNMP イベント MIB を無効にする」
SNMP イベント MIB の変更	261 ページの「SNMP イベント MIB を変更する」
MIB のトラップ通知を受信するホストリストへの SNMP ホストの追加	262 ページの「SNMP ホストがノード上の SNMP トラップを受信できるようにする」
SNMP ホストの削除	263 ページの「SNMP ホストがノード上の SNMP トラップを受信できないようにする」
SNMP ユーザーの追加	264 ページの「SNMP ユーザーをノードに追加する」
SNMP ユーザーの削除	265 ページの「SNMP ユーザーをノードから削除する」

▼ SNMP イベント MIB を有効にする

この手順では、SNMP イベント MIB を有効化する方法を説明します。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 スーパーユーザーになるか、RBAC 承認 `solaris.cluster.modify` を提供する役割になります。
- 2 SNMP イベント MIB を有効にします。

```
phys-schost-1# clsnmpmib enable [-n node] MIB
```

`[-n node]`           有効にするイベント MIB がある *node* を指定します。ノード ID またはノード名を指定できます。このオプションを指定しないと、デフォルトで現在のノードが使用されます。

*MIB*                有効にする MIB の名前を指定します。この場合、MIB 名は event にしてください。

## ▼ SNMP イベント MIB を無効にする

この手順では、SNMP イベント MIB を無効化する方法を説明します。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 スーパーユーザーになるか、RBAC 承認 **solaris.cluster.modify** を提供する役割になります。
- 2 **SNMP イベント MIB** を無効にします。

```
phys-schost-1# clsnmpmib disable -n node MIB
```

**-n node**                      無効にするイベント MIB がある *node* を指定します。ノード ID またはノード名を指定できます。このオプションを指定しないと、デフォルトで現在のノードが使用されます。

**MIB**                          無効にする MIB の種類を指定します。この場合、**event** を指定してください。

## ▼ SNMP イベント MIB を変更する

この手順では、SNMP イベント MIB のプロトコルを変更する方法を説明します。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 スーパーユーザーになるか、RBAC 承認 **solaris.cluster.modify** を提供する役割になります。
- 2 **SNMP イベント MIB** のプロトコルを変更します。

```
phys-schost-1# clsnmpmib set -n node -p version=value MIB
```

**-n node**                      変更するイベント MIB がある *node* を指定します。ノード ID またはノード名を指定できます。このオプションを指定しないと、デフォルトで現在のノードが使用されます。

-p version=*value*

MIB で使用する SNMP プロトコルのバージョンを指定します。 *value* は次のように指定します。

- version=SNMPv2
- version=snmpv2
- version=2
- version=SNMPv3
- version=snmpv3
- version=3

#### MIB

サブコマンドが適用される単数または複数の MIB の名前を指定します。この場合、event を指定してください。このオペランドを指定しない場合は、サブコマンドが、すべての MIB を意味するデフォルトのプラス記号 (+) を使用します。MIB オペランドを使用する場合は、ほかのすべてのコマンド行オプションのあとで、MIB を空白区切りのリスト内に指定します。

## ▼ SNMP ホストがノード上の SNMP トラップを受信できるようにする

この手順では、ノード上の SNMP ホストを、MIB のトラップ通知を受信するホストのリストに追加する方法を説明します。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 スーパーユーザーになるか、RBAC 承認 **solaris.cluster.modify** を提供する役割になります。
- 2 ホストを、別のノード上のコミュニティの SNMP ホストリストに追加します。

```
phys-schost-1# clsnmphost add -c SNMPcommunity [-n node] host
```

-c *SNMPcommunity*

ホスト名とともに使用される SNMP コミュニティ名を指定します。

ホストを public 以外のコミュニティに追加する場合は、コミュニティ名 *SNMPcommunity* を指定してください。add サブコマンドを -c オプションなしで使用すると、このサブコマンドは public をデフォルトのコミュニティ名として使用します。

指定されたコミュニティ名が存在しない場合、このコマンドはそのコミュニティを作成します。

*-n node*

クラスタ内の SNMP MIB に対するアクセス権を付与されている SNMP ホストの *node* の名前を指定します。ノード名またはノード ID を指定できます。このオプションを指定しないと、デフォルトで現在のノードが使用されます。

*host*

クラスタ内の SNMP MIB に対するアクセス権が付与されたホストの名前、IP アドレス、または IPv6 アドレスを指定します。

## ▼ SNMP ホストがノード上の **SNMP** トラップを受信できないようにする

この手順では、ノード上の SNMP ホストを、MIB のトラップ通知を受信するホストのリストから削除する方法を説明します。

*phys-schost#* プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 スーパーユーザーになるか、RBAC 承認 **solaris.cluster.modify** を提供する役割になります。
- 2 指定のノード上のコミュニティの **SNMP** ホストリストからホストを削除します。

```
phys-schost-1# clsnmphost remove -c SNMPcommunity -n node host
```

*remove*

指定のノードから指定の SNMP ホストを削除します。

*-c SNMPcommunity*

SNMP ホストを削除する SNMP コミュニティの名前を指定します。

*-n node*

構成から削除される SNMP ホストの *node* の名前を指定します。ノード名またはノード ID を指定できます。このオプションを指定しないと、デフォルトで現在のノードが使用されます。

*host*

構成から削除されるホストの名前、IP アドレス、または IPv6 アドレスを指定します。

指定の SNMP コミュニティ内のすべてのホストを削除するには、*-c* オプション付きの *host* に正符号 (+) を使用します。すべてのホストを削除するには、*host* に正符号 + を使用します。

## ▼ SNMP ユーザーをノードに追加する

この手順では、ノード上の SNMP ユーザー構成に SNMP ユーザーを追加する方法を説明します。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 スーパーユーザーになるか、RBAC 承認 `solaris.cluster.modify` を提供する役割になります。
- 2 SNMP ユーザーを追加します。

```
phys-schost-1# clsnmpuser create -n node -a authentication \
-f password user
```

**-n node** SNMP ユーザーが追加されるノードを指定します。ノード ID またはノード名を指定できます。このオプションを指定しないと、デフォルトで現在のノードが使用されます。

**-a authentication** ユーザーの承認に使用する認証プロトコルを指定します。認証プロトコルの値は SHA または MD5 です。

**-f password** SNMP ユーザーパスワードを含むファイルを指定します。新しいユーザーを作成する際にこのオプションを指定しないと、コマンドはパスワードを求めるプロンプトを表示します。このオプションは、`add` サブコマンドとだけ有効です。

ユーザーパスワードは、次の形式で、独立した行の上に指定します。

```
user:password
```

パスワードには次に示す文字または空白文字を含めることはできません。

- ; (セミコロン)
- : (コロン)
- \ (バックスラッシュ)
- \n (復帰改行)

**user** 追加する SNMP ユーザーの名前を指定します。



## ▼ SNMP ユーザーをノードから削除する

この手順では、ノード上の SNMP ユーザー構成から SNMP ユーザーを削除する方法を説明します。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 スーパーユーザーになるか、RBAC 承認 `solaris.cluster.modify` を提供する役割になります。
- 2 SNMP ユーザーを削除します。

```
phys-schost-1# clsnmpuser delete -n node user
```

`-n node` SNMP ユーザーが削除されるノードを指定します。ノード ID またはノード名を指定できます。このオプションを指定しないと、デフォルトで現在のノードが使用されます。

`user` 削除する SNMP ユーザーの名前を指定します。

## 負荷制限の構成

負荷制限を設定することによって、ノードまたはゾーン間でのリソースグループの負荷の自動分散を有効にできます。一連の負荷制限はクラスタノードごとに設定できます。リソースグループに負荷係数を割り当てると、その負荷係数はノードの定義済み負荷制限に対応します。デフォルトの動作では、リソースグループの負荷がそのリソースグループのノードリスト内の使用可能なすべてのノードに均等に分散されます。

リソースグループは RGM によってリソースグループのノードリストのノード上で起動されるため、ノードの負荷制限を超えることはありません。RGM によってリソースグループがノードに割り当てられると、各ノードのリソースグループの負荷係数が合計され、合計負荷が算出されます。次に、合計負荷がそのノードの負荷制限と比較されます。

負荷制限は次の項目から構成されます。

- ユーザーが割り当てた名前。
- 弱い制限値 – 弱い負荷制限は一時的に超えることができます。
- 強い負荷制限 – 強い負荷制限は超えることはできず、厳格に適用されます。

1つのコマンドで強い制限と弱い制限の両方を設定できます。いずれかの制限が明示的に設定されていない場合は、デフォルト値が使用されます。各ノードの強い負荷制限値と弱い負荷制限値の作成と変更には、`clnode create-loadlimit`、`clnode set-loadlimit`、および `clnode delete-loadlimit` コマンドを使用します。詳細については、[clnode\(1CL\)](#) のマニュアルページを参照してください。

高い優先度を持つようにリソースグループを構成すると、特定のノードから移動させられる可能性が低くなります。`preemption_mode` プロパティを設定して、ノードの過負荷が原因でリソースグループが優先度の高いリソースによってノードから横取りされるかどうかを判定することもできます。`concentrate_load` プロパティを使用して、リソースグループの負荷をできるだけ少ないノードに集中させることもできます。`concentrate_load` プロパティのデフォルト値は、`FALSE` です。

---

注- 負荷制限は、グローバルクラスタまたはゾーンクラスタのノード上で構成できます。負荷制限を設定するには、コマンド行、`clsetup` ユーティリティ、または Oracle Solaris Cluster Manager インタフェースを使用します。次の手順は、コマンド行を使用して負荷制限を構成する方法を示したものです。

---

## ▼ ノードに負荷制限を構成する

- 1 グローバルクラスタの任意のノードで、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.modify` を提供する役割になります。

- 2 負荷分散を使用するノードに対して、負荷制限を作成および設定します。

```
clnode create-loadlimit -p limitname=mem_load -Z zc1 -p
softlimit=11 -p hardlimit=20 node1 node2 node3
```

この例では、ゾーンクラスタ名は `zc1` です。サンプルプロパティは `mem_load` で、弱い負荷制限は 11、強い負荷制限は 20 です。強い制限と弱い制限は省略可能な引数で、特に定義しなかった場合、デフォルトは無制限です。詳細については、[clnode\(1CL\)](#) のマニュアルページを参照してください。

- 3 負荷係数値を各リソースグループに割り当てます。

```
clresourcegroup set -p load_factors=mem_load@50,factor2@1 rg1 rg2
```

この例では、2つのリソースグループ (`rg1` と `rg2`) で負荷係数が設定されています。負荷係数の設定は、ノードの定義済み負荷制限に対応します。この手順は、リソースグループの作成中に `clresourcegroup create` コマンドを使用して実行することもできます。詳細は、[clresourcegroup\(1CL\)](#) のマニュアルページを参照してください。

- 4 必要に応じて、既存の負荷を再分散できます (`clrg remaster`)。

```
clresourcegroup remaster rg1 rg2
```

このコマンドにより、リソースグループを現在のマスターからほかのノードに移動し、均等な負荷分散を実現できます。

- 5 必要に応じて、一部のリソースグループに、ほかのリソースグループより高い優先度を与えることができます。

```
clresourcegroup set -p priority=600 rg1
```

デフォルトの優先度は500です。優先度の値が高いリソースグループは、ノードの割り当てにおいて、優先度の値が低いリソースグループよりも優先されます。

- 6 必要に応じて、**Preemption\_mode** プロパティを設定できます。

```
clresourcegroup set -p Preemption_mode=No_cost rg1
```

HAS\_COST、NO\_COST、およびNEVER オプションについては、[clresourcegroup\(1CL\)](#)のマニュアルページを参照してください。

- 7 必要に応じて、**Concentrate\_load** プロパティを設定できます。

```
cluster set -p Concentrate_load=TRUE
```

- 8 必要に応じて、リソースグループ間のアフィニティを指定できます。

強い正または負のアフィニティは負荷分散より優先されます。強いアフィニティや強い負荷制限が無効になることはありません。強いアフィニティと強い負荷制限の両方を設定すると、両方の制限が満たされなかった場合に一部のリソースグループが強制的にオフラインのままになることがあります。

次の例では、ゾーンクラスタ **zc1** のリソースグループ **rg1** とゾーンクラスタ **zc2** のリソースグループ **rg2** の間の強い正のアフィニティを指定しています。

```
clresourcegroup set -p RG_affinities==+zc2:rg2 zc1:rg1
```

- 9 クラスタ内のすべてのグローバルクラスタノードとゾーンクラスタノードのステータスを確認します。

```
clnode status -Z all -v
```

出力には、ノードまたは非大域ゾーンで定義された負荷制限設定がすべて含まれます。

## ゾーンクラスタ管理タスクの実行

ゾーンクラスタに関するほかの管理タスク (ゾーンパスの移動、アプリケーションを実行するためのゾーンクラスタの準備、ゾーンクラスタの複製など) を実行できます。これらのコマンドは、必ずグローバルクラスタの投票ノードから実行してください。

clsetup ユーティリティを使用してゾーンクラスタの構成ウィザードを起動すると、新しいゾーンクラスタを作成したり、ファイルシステムまたはストレージデバイスを追加したりできます。clzonecluster install -c を実行してプロファイルを構成すると、ゾーンクラスタのゾーンが構成されます。clsetup ユーティリティまたは -c config\_profile オプションの使用手順については、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』の「[ゾーンクラスタの確立](#)」を参照してください。

注- グローバルクラスタ内の投票ノードからのみ実行する Oracle Solaris Cluster コマンドは、ゾーンクラスタには使用できません。各種ゾーンでのコマンドの有効な使用方法については、Oracle Solaris Cluster の該当するマニュアルページを参照してください。

表 9-3 その他のゾーンクラスタのタスク

タスク	手順
新規ゾーンパスへのゾーンパスの移動	clzonecluster move -f zonepath zoneclustername
アプリケーション実行用のゾーンクラスタの準備	clzonecluster ready -n nodename zoneclustername
ゾーンクラスタの複製	clzonecluster clone -Z target- zoneclustername [ -m copymethod] source-zoneclustername  clone サブコマンドを使用する前に、ソース ゾーンクラスタを停止してください。複製先 のゾーンクラスタは、構成済みであることが 必要です。
ゾーンクラスタへのネットワークアドレスの追加	269 ページの「 <a href="#">ゾーンクラスタにネットワーク アドレスを追加する方法</a> 」
ゾーンクラスタの削除	270 ページの「 <a href="#">ゾーンクラスタを削除する</a> 」
ゾーンクラスタからファイルシステムを削除	271 ページの「 <a href="#">ゾーンクラスタからファイルシ ステムを削除する</a> 」
ゾーンクラスタからストレージデバイスを削除	273 ページの「 <a href="#">ゾーンクラスタからストレージ デバイスを削除する</a> 」
ノードのアンインストールに関するトラブル シューティング	258 ページの「 <a href="#">ノードのアンインストールのト ラブルシューティング</a> 」
Oracle Solaris Cluster SNMP イベント MIB の作 成、設定、および管理	259 ページの「 <a href="#">Oracle Solaris Cluster SNMP イベ ント MIB の作成、設定、および管理</a> 」

## ▼ ゾーンクラスタにネットワークアドレスを追加する方法

この手順では、既存のゾーンクラスタで使用するネットワークアドレスを追加します。ネットワークアドレスは、論理ホストまたは共有 IP アドレスリソースをゾーンクラスタに構成する場合に使用します。clsetup ユーティリティを複数回実行して、必要な数のネットワークアドレスを追加できます。

- 1 目的のゾーンクラスタをホストしているグローバルクラスタのノードで、スーパーユーザーになります。
- 2 グローバルクラスタ上で、ゾーンクラスタで使用するクラスタファイルシステムを構成します。  
clsetup ユーティリティを起動します。  
phys-schost# clsetup  
メインメニューが表示されます。
- 3 「ゾーンクラスタ」メニュー項目を選択します。
- 4 「ゾーンクラスタにネットワークアドレスを追加」メニュー項目を選択します。
- 5 ネットワークアドレスを追加するゾーンクラスタを選択します。

- 6 追加するネットワークアドレスを指定するためのプロパティを選択します。  
address=value 論理ホストまたは共有 IP アドレスリソースをゾーンクラスタに構成する場合に使用されるネットワークアドレスを指定します。例: 192.168.100.101。

次の種類のネットワークアドレスがサポートされています。

- 有効な IPv4 アドレス (オプションで/および接頭辞長が続く)。
- 有効な IPv6 アドレス (/および接頭辞長が続く必要があります)。
- IPv4 アドレスに解決するホスト名。IPv6 アドレスに解決するホスト名はサポートされていません。

ネットワークアドレスの詳細は、[zonecfg\(1M\)](#)のマニュアルページを参照してください。

- 7 ネットワークアドレスを追加するには、**a**を入力します。

- 8 cを入力して、構成の変更を保存します。

構成変更の結果が表示されます。例:

```
>>> Result of Configuration Change to the Zone Cluster(sczone) <<<

Adding network address to the zone cluster...

The zone cluster is being created with the following configuration

/usr/cluster/bin/clzonecluster configure sczone
add net
set address=phys-schost-1
end

All network address added successfully to sczone.
```

- 9 完了後 **clsetup** ユーティリティーを終了します。

## ▼ ゾーンクラスタを削除する

グローバルクラスタ上に構成されているゾーンクラスタは、特定の1つのゾーンクラスタを削除することも、ワイルドカードを使用してすべてのゾーンクラスタを削除することもできます。構成されていないゾーンクラスタは、削除できません。

- 1 グローバルクラスタのノードで、スーパーユーザーになるか、**RBAC**の承認 **solaris.cluster.modify**を提供する役割になります。グローバルクラスタのノードから、次の手順のステップをすべて実行します。
- 2 ゾーンクラスタからすべてのリソースグループとそのリソースを削除します。

```
phys-schost# clresourcegroup delete -F -Z zoneclustername +
```

---

注- この手順は、グローバルクラスタノードから実行されます。この手順をゾーンクラスタのノードから実行するには、ゾーンクラスタノードにログインし、コマンドの「-Z zonecluster」を省略します。

---

- 3 ゾーンクラスタを停止します。  
phys-schost# **clzonecluster halt zoneclustername**
- 4 ゾーンクラスタをアンインストールします。  
phys-schost# **clzonecluster uninstall zoneclustername**
- 5 ゾーンクラスタを構成解除します。  
phys-schost# **clzonecluster delete zoneclustername**

## 例 9-11 グローバルクラスタからのゾーンクラスタの削除

```
phys-schost# clresourcegroup delete -F -Z sczone +
```

```
phys-schost# clzonecluster halt sczone
```

```
phys-schost# clzonecluster uninstall sczone
```

```
phys-schost# clzonecluster delete sczone
```

## ▼ ゾーンクラスタからファイルシステムを削除する

ファイルシステムをゾーンクラスタにエクスポートするには、直接マウントまたはループバックマウントを使用します。

ゾーンクラスタでは、次の直接マウントがサポートされます。

- UFS ローカルファイルシステム
- Oracle Automatic Storage Management Cluster File System (Oracle ACFS) ファイルシステム
- QFS スタンドアロンファイルシステム
- QFS 共有ファイルシステム (Oracle RAC のサポートに使用する場合のみ)
- ZFS (データセットとしてエクスポートされたもの)
- サポートされている NAS デバイスの NFS

ゾーンクラスタでは、次のループバックマウントを管理できます。

- UFS ローカルファイルシステム
- QFS スタンドアロンファイルシステム
- QFS 共有ファイルシステム (Oracle RAC のサポートに使用する場合のみ)
- UFS クラスタファイルシステム

ファイルシステムのマウントを管理する `HASStoragePlus` または `ScalMountPoint` リソースを構成します。ファイルシステムをゾーンクラスタに追加する手順については、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』の「[ゾーンクラスタにファイルシステムを追加する](#)」を参照してください。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 目的のゾーンクラスタをホストしているグローバルクラスタのノードで、スーパーユーザーになります。この手順のいくつかのステップはグローバルクラスタのノードから行います。他のステップは、ゾーンクラスタのノードから実行されます。
- 2 削除するファイルシステムに関連するリソースを削除します。
  - a. 削除するゾーンクラスタのファイルシステム用に構成されている **Oracle Solaris Cluster** のリソースタイプ (**HASStoragePlus**、**SUNW.ScalMountPoint** など) を特定し、削除します。

```
phys-schost# clresource delete -F -Z zoneclustername fs_zone_resources
```

- b. 削除するファイルシステム用のグローバルクラスタ内に構成されている **SUNW.qfs** タイプの **Oracle Solaris Cluster** リソースがあれば、そのリソースを特定し、削除します。

```
phys-schost# clresource delete -F fs_global_resources
```

-F オプションを指定すると、前もって無効にしていないリソースも含め、指定したリソースがすべて強制的に削除されるため、このオプションは注意して使用してください。すべての指定リソースが、ほかのリソースのリソース関係設定から削除されるため、クラスタ内のサービスが失われることがあります。削除されていない依存リソースは、無効な状態やエラー状態になる可能性があります。詳細は、**clresource(1CL)** のマニュアルページを参照してください。

---

ヒント-削除したリソースのリソースグループがあとで空になると、そのリソースグループを安全に削除できます。

---

- 3 ファイルシステムのマウントポイントディレクトリのパスを調べます。例:

```
phys-schost# clzonecluster configure zoneclustername
```

- 4 ファイルシステムをゾーンクラスタの構成から削除します。

```
phys-schost# clzonecluster configure zoneclustername
```

```
clzc:zoneclustername> remove fs dir=filesystemdirectory
```

```
clzc:zoneclustername> commit
```

ファイルシステムのマウントポイントは、**dir=** で指定します。

- 5 ファイルシステムが削除されたことを確認します。

```
phys-schost# clzonecluster show -v zoneclustername
```



**例 9-12** ゾーンクラスタ内の高可用性ファイルシステムの削除

この例は、sczone というゾーンクラスタ内に構成された、マウントポイントディレクトリ (/local/ufs-1) のあるファイルシステムを削除する方法を示しています。リソースは hasp-rs で、そのタイプは HAStoragePlus です。

```
phys-schost# clzonecluster show -v sczone
...
Resource Name: fs
 dir: /local/ufs-1
 special: /dev/md/ds1/dsk/d0
 raw: /dev/md/ds1/rdisk/d0
 type: ufs
 options: [logging]
...
phys-schost# clresource delete -F -Z sczone hasp-rs
phys-schost# clzonecluster configure sczone
clzc:sczone> remove fs dir=/local/ufs-1
clzc:sczone> commit
phys-schost# clzonecluster show -v sczone
```

**例 9-13** ゾーンクラスタ内の高可用性 ZFS ファイルシステムの削除

この例は、リソース hasp-rs、タイプ SUNW.HAStoragePlus の sczone ゾーンクラスタ内で構成された、HAzpool という ZFS プール内の ZFS ファイルシステムを削除する方法を示します。

```
phys-schost# clzonecluster show -v sczone
...
Resource Name: dataset
 name: HAzpool
...
phys-schost# clresource delete -F -Z sczone hasp-rs
phys-schost# clzonecluster configure sczone
clzc:sczone> remove dataset name=HAzpool
clzc:sczone> commit
phys-schost# clzonecluster show -v sczone
```

## ▼ ゾーンクラスタからストレージデバイスを削除する

ゾーンクラスタから、SVM ディスクセットや DID デバイスなどのストレージデバイスを削除できます。この手順は、ゾーンクラスタからストレージデバイスを削除する場合に実行します。

- 1 目的のゾーンクラスタをホストしているグローバルクラスタのノードで、スーパーユーザーになります。この手順のいくつかのステップはグローバルクラスタのノードから行います。ほかのステップは、ゾーンクラスタのノードから実行することが可能です。
- 2 削除するデバイスに関連するリソースを削除します。削除するゾーンクラスタのデバイス用に構成されている **Oracle Solaris Cluster** のリソースタイプ (**SUNW.HAStoragePlus**、**SUNW.ScalDeviceGroup** など) を特定し、削除します。

```
phys-schost# clresource delete -F -Z zoneclustername dev_zone_resources
```

- 3 削除するデバイスに対して一致するエントリを調べます。

```
phys-schost# clzonecluster show -v zoneclustername
...
Resource Name: device
match: <device_match>
...
```

- 4 デバイスをゾーンクラスタの構成から削除します。

```
phys-schost# clzonecluster configure zoneclustername
clzc:zoneclustername> remove device match=<devices_match>
clzc:zoneclustername> commit
clzc:zoneclustername> end
```

- 5 ゾーンクラスタをリブートします。

```
phys-schost# clzonecluster reboot zoneclustername
```

- 6 デバイスの削除を確認します。

```
phys-schost# clzonecluster show -v zoneclustername
```

#### 例 9-14 SVM ディスクセットをゾーンクラスタから削除する

この例は、sczone というゾーンクラスタ内に構成された apachedg という SVM ディスクセットを削除する方法を示しています。apachedg ディスクセットのセット番号は 3 です。このデバイスは、クラスタに構成された zc\_rs のリソースにより使用されます。

```
phys-schost# clzonecluster show -v sczone
...
Resource Name: device
match: /dev/md/apachedg/*dsk/*
Resource Name: device
match: /dev/md/shared/3/*dsk/*
...
phys-schost# clresource delete -F -Z sczone zc_rs

phys-schost# ls -l /dev/md/apachedg
lrwxrwxrwx 1 root root 8 Jul 22 23:11 /dev/md/apachedg -> shared/3
phys-schost# clzonecluster configure sczone
```

```
clzc:sczone> remove device match=/dev/md/apachedg/*dsk/*
clzc:sczone> remove device match=/dev/md/shared/3/*dsk/*
clzc:sczone> commit
clzc:sczone> end
phys-schost# clzonecluster reboot sczone
phys-schost# clzonecluster show -v sczone
```

#### 例 9-15 DID デバイスをゾーンクラスタから削除する

この例は、DID デバイス d10 および d11 を削除する方法を示しています。このデバイスは、sczone というゾーンクラスタに構成されています。このデバイスは、クラスタに構成された zc\_rs のリソースにより使用されます。

```
phys-schost# clzonecluster show -v sczone
...
Resource Name: device
match: /dev/did/*dsk/d10*
Resource Name: device
match: /dev/did/*dsk/d11*
...
phys-schost# clresource delete -F -Z sczone zc_rs
phys-schost# clzonecluster configure sczone
clzc:sczone> remove device match=/dev/did/*dsk/d10*
clzc:sczone> remove device match=/dev/did/*dsk/d11*
clzc:sczone> commit
clzc:sczone> end
phys-schost# clzonecluster reboot sczone
phys-schost# clzonecluster show -v sczone
```

## トラブルシューティング

この節では、テスト用に使用できるトラブルシューティング手順について説明します。

### グローバルクラスタ外でのアプリケーションの実行

#### ▼ 非クラスタモードでブートしたノードから **Solaris Volume Manager** メタセットを取得する

この手順を使用して、テスト用にグローバルクラスタ外でアプリケーションを実行します。

- 1 **Solaris Volume Manager** メタセットで定足数デバイスが使用されているかどうかを確認し、定足数デバイスが **SCSI2** または **SCSI3** 予約を使用するかどうかを確認します。

```
phys-schost# clquorum show
```

- a. 定足数デバイスが **Solaris Volume Manager** メタセットにある場合は、あとで非クラスタモードにするメタセットには含まれない、新しい定足数デバイスを追加します。

```
phys-schost# clquorum add did
```

- b. 古い定足数デバイスを削除します。

```
phys-schost# clquorum remove did
```

- c. 定足数デバイスが **SCSI2** 予約を使用する場合は、古い定足数からの **SCSI2** 予約をスクラブして、**SCSI2** 予約が残らないことを確認します。

次のコマンドは、PGRE (Persistent Group Reservation Emulation) 鍵を検索します。ディスク上に鍵が存在しない場合は、`errno=22` メッセージが表示されます。

```
/usr/cluster/lib/sc/pgre -c pgre_inkeys -d /dev/did/rdisk/dids2
```

鍵が見つかったら、PGRE 鍵をスクラブします。

```
/usr/cluster/lib/sc/pgre -c pgre_scrub -d /dev/did/rdisk/dids2
```



注意 - アクティブな定足数デバイス鍵をディスクからスクラブすると、次の再構成時にクラスタでパニックが発生し、「操作可能な定足数を失いました」というメッセージが表示されます。

- 2 非クラスタモードでブートするグローバルクラスタノードを退避します。

```
phys-schost# clresourcegroup evacuate -n targetnode
```

- 3 **HAStorage** または **HAStoragePlus** リソースを含み、あとで非クラスタモードにするメタセットの影響を受けるデバイスまたはファイルシステムを含む、1つまたは複数のリソースグループをオフラインにします。

```
phys-schost# clresourcegroup offline resourcegroupname
```

- 4 オフラインにしたリソースグループ内のすべてのリソースを無効にします。

```
phys-schost# clresource disable resourcename
```

- 5 リソースグループを非管理状態に切り替えます。

```
phys-schost# clresourcegroup unmanage resourcegroupname
```

- 6 対応する1つまたは複数のデバイスグループをオフラインにします。

```
phys-schost# cldevicegroup offline devicegroupname
```

- 7 1つまたは複数のデバイスグループを無効にします。  
`phys-schost# cldevicegroup disable devicegroupname`
- 8 パッシブノードを非クラスタモードでブートします。  
`phys-schost# reboot -x`
- 9 続ける前にパッシブノードでブートプロセスが完了していることを確認します。  
`phys-schost# svcs -x`
- 10 メタセット内のディスクに **SCSI3** 予約があるかどうかを調べます。メタセットのすべてのディスクで次のコマンドを実行します。  
`phys-schost# /usr/cluster/lib/sc/scsi -c inkeys -d /dev/did/rdisk/dids2`
- 11 ディスクに **SCSI3** 予約が存在する場合は、それらをスクラブします。  
`phys-schost# /usr/cluster/lib/sc/scsi -c scrub -d /dev/did/rdisk/dids2`
- 12 退避したノードでメタセットを取得します。  
`phys-schost# metaset -s name -C take -f`
- 13 メタセットで定義されたデバイスが含まれている1つまたは複数のファイルシステムをマウントします。  
`phys-schost# mount device mountpoint`
- 14 アプリケーションを起動し、目的のテストを行います。テストが終了したら、アプリケーションを停止します。
- 15 ノードをリブートし、ブートプロセスが終了するまで待ちます。  
`phys-schost# reboot`
- 16 1つまたは複数のデバイスグループをオンラインにします。  
`phys-schost# cldevicegroup online -e devicegroupname`
- 17 1つまたは複数のリソースグループを起動します。  
`phys-schost# clresourcegroup online -eM resourcegroupname`

## 破損したディスクセットの復元

この手順は、ディスクセットが破損している場合、またはクラスタのノードがディスクセットの所有権を取得できない状態になっている場合に使用します。状態を明らかにしようとしたができなかった場合は、ディスクセットを修正するための最後の試みとして次の手順に従います。

次の手順は、Solaris Volume Manager のメタセットおよび複数所有者 Solaris Volume Manager のメタセットに適用します。

## ▼ Solaris Volume Manager ソフトウェア構成を保存する方法

最初からディスクセットを復元すると、時間がかかり、エラーが発生しやすくなります。代わりの方法として適切なのは、`metastat` コマンドを使用して定期的に複製をバックアップするか、Oracle Explorer (SUNWexplo) を使用してバックアップを作成する方法です。その後、保存された構成を使用して、ディスクセットを再作成します。(prtvtoc および `metastat` コマンドを使用して) 現在の構成をファイルに保存し、ディスクセットとそのコンポーネントを再作成します。279 ページの「[Solaris Volume Manager ソフトウェア構成を再作成する方法](#)」を参照してください。

- 1 ディスクセット内の各ディスクのパーティションテーブルを保存します。

```
/usr/sbin/prtvtoc /dev/global/rdisk/diskname > /etc/lvm/diskname.vtoc
```

- 2 Solaris Volume Manager ソフトウェア構成を保存します。

```
/bin/cp /etc/lvm/md.tab /etc/lvm/md.tab_ORIGINAL
```

```
/usr/sbin/metastat -p -s setname >> /etc/lvm/md.tab
```

---

注 - /etc/vfstab ファイルなどのほかの構成ファイルが、Solaris Volume Manager ソフトウェアを参照する場合があります。この手順では、同一の Solaris Volume Manager ソフトウェア構成を再構築することを想定しているため、マウント情報は同じです。セットの所有権を持つノード上で Oracle Explorer (SUNWexplo) を実行すると、prtvtoc および `metaset -p` の情報が取得されます。

---

## ▼ 破損したディスクセットを削除する

1 つのノードまたはすべてのノードからセットを削除すると、構成が削除されます。ノードからディスクセットを削除するには、ノードにディスクセットの所有権があってははいけません。

- 1 すべてのノードで削除コマンドを実行します。

```
/usr/sbin/metaset -s setname -P
```

このコマンドを実行すると、データベースの複製から、ディスクセット情報のほか、Oracle Solaris Cluster リポジトリが削除されます。`-P` および `-c` オプションを使用すると、Solaris Volume Manager 環境を完全に再構築しなくても、ディスクセットを削除できます。

---

注- ノードがクラスタモードからブートされたときに複数所有者のディスクセットが削除された場合、dcs 構成ファイルから情報を削除する必要があることがあります。

```
/usr/cluster/lib/sc/dcs_config -c remove -s setname
```

詳細は、[dcs\\_config\(1M\)](#) のマニュアルページを参照してください。

---

- 2 データベースの複製からディスクセット情報のみを削除する場合は、次のコマンドを使用します。

```
/usr/sbin/metaset -s setname -C purge
```

通常は、-c オプションではなく、-P オプションを使用するようにしてください。-c オプションを使用すると、Oracle Solaris Cluster ソフトウェアは引き続きディスクセットを認識するため、ディスクセットの再作成時に問題が発生する場合があります。

- a. **metaset** コマンドで -c オプションを使用した場合は、問題が発生しないかどうかを確認するために、まずディスクセットを作成します。

- b. 問題が発生した場合は、dcs 構成ファイルから情報を削除してください。

```
/usr/cluster/lib/sc/dcs_config -c remove -s setname
```

purge オプションが失敗した場合は、最新のカーネルとメタデバイスの更新がインストールされていることを確認し、[My Oracle Support](#) にアクセスします。

## ▼ Solaris Volume Manager ソフトウェア構成を再作成する方法

この手順に従うのは、Solaris Volume Manager ソフトウェア構成が完全に失われた場合のみです。この手順では、現在の Solaris Volume Manager 構成とそのコンポーネントが保存され、破損したディスクセットが削除されていることを想定しています。

---

注- メディエータは、2 ノードクラスタでのみ使用するようにしてください。

---

- 1 新しいディスクセットを作成します。

```
/usr/sbin/metaset -s setname -a -h nodename1 nodename2
```

これが複数所有者ディスクセットの場合は、次のコマンドを使用して新しいディスクセットを作成します。

```
/usr/sbin/metaset -s setname -aM -h nodename1 nodename2
```

- 2 セットが作成されたのと同じホストで、必要に応じてメディアータホストを追加します (2 ノードのみ)。

```
/usr/sbin/metaset -s setname -a -m nodename1 nodename2
```

- 3 この同じホストからディスクセットに同じディスクをふたたび追加します。

```
/usr/sbin/metaset -s setname -a /dev/did/rdisk/diskname /dev/did/rdisk/diskname
```

- 4 削除したディスクセットを再作成する場合は、ボリュームの目次 (**Volume Table of Contents, VTOC**) がディスクに残っているため、この手順は省略できます。ただし、復元するセットを再作成する場合は、**/etc/lvm/diskname.vtoc** ファイルに保存されている構成に従ってディスクをフォーマットするようにしてください。例:

```
/usr/sbin/fmthard -s /etc/lvm/d4.vtoc /dev/global/rdisk/d4s2
```

```
/usr/sbin/fmthard -s /etc/lvm/d8.vtoc /dev/global/rdisk/d8s2
```

このコマンドはどのノードでも実行できます。

- 5 メタデバイスごとに、既存の **/etc/lvm/md.tab** ファイルの構文を確認します。

```
/usr/sbin/metainit -s setname -n -a metadvice
```

- 6 保存されている構成から各メタデバイスを作成します。

```
/usr/sbin/metainit -s setname -a metadvice
```

- 7 メタデバイスにファイルシステムが存在する場合は、**fsck** コマンドを実行します。

```
/usr/sbin/fsck -n /dev/md/setname/rdisk/metadvice
```

**fsck** コマンドが、スーパーブロック数など少数のエラーのみを表示した場合、デバイスは正しく再構築されている可能性が高くなります。その後、**fsck** コマンドを **-n** オプションを指定せずに実行できます。多数のエラーが表示された場合は、メタデバイスが正しく再構築されているかどうかを確認します。正しく再構築されている場合は、**fsck** エラーを確認して、ファイルシステムが回復可能かどうかを判断します。回復できない場合は、バックアップからデータを復元するようにしてください。

- 8 すべてのクラスタノード上のほかのすべてのメタセットを **/etc/lvm/md.tab** ファイルに連結してから、ローカルディスクセットに連結します。

```
/usr/sbin/metastat -p >> /etc/lvm/md.tab
```



## CPU 使用率の制御の構成

---

CPU の使用率を制御したい場合は、CPU 制御機能を構成します。CPU 制御機能の構成についての詳細は、[rg\\_properties\(5\)](#) のマニュアルページを参照してください。この章では、次のトピックについて説明します。

- 281 ページの「CPU 制御の概要」
- 283 ページの「CPU 制御の構成」

### CPU 制御の概要

Oracle Solaris Cluster ソフトウェアを使用すると、CPU の使用率を制御できます。

CPU 制御機能は、Oracle Solaris OS で利用可能な機能に基づいて構築されています。ゾーン、プロジェクト、リソースプール、プロセッサセット、およびスケジューリングクラスについては、『[Oracle Solaris の管理: Oracle Solaris コンテナ - リソース管理と Oracle Solaris ゾーン](#)』を参照してください。

Oracle Solaris OS では、次の作業を実行できます。

- CPU シェアをリソースグループに割り当てる
- プロセッサをリソースグループに割り当てる

### シナリオの選択

構成の選択肢と、選択するオペレーティングシステムのバージョンに応じて、さまざまなレベルの CPU 制御を行うことができます。この章で説明する CPU 制御のすべての局面は、リソースグループプロパティ `RG_SLM_TYPE` が `automated` に設定されていることに依存します。

表 10-1 で、使用可能なさまざまな構成シナリオを説明します。

表 10-1 CPU 制御のシナリオ

説明	手順
<p>リソースグループは、グローバルクラスタ投票ノードで動作します。</p> <p>CPU シェアをリソースグループとゾーンに割り当て、<code>project.cpu-shares</code> および <code>zone.cpu-shares</code> の値を指定します。</p> <p>この手順は、グローバルクラスタの非投票ノードが構成されているかどうかにかかわらず実行できます。</p>	<p><a href="#">283 ページの「グローバルクラスタの投票ノードで CPU 使用率を制御する方法」</a></p>
<p>リソースグループは、デフォルトプロセッサセットを使用して、グローバルクラスタの非投票ゾーンで動作します。</p> <p>CPU シェアをリソースグループとゾーンに割り当て、<code>project.cpu-shares</code> および <code>zone.cpu-shares</code> の値を指定します。</p> <p>この手順は、プロセッサセットのサイズを制御する必要がない場合に実行します。</p>	<p><a href="#">285 ページの「デフォルトプロセッサセットを使用してグローバルクラスタの非投票ノードで CPU 使用率を制御する方法」</a></p>
<p>リソースグループは、専用プロセッサセットを使用して、グローバルクラスタの非投票ゾーンで動作します。</p> <p>CPU シェアをリソースグループに割り当てて、<code>project.cpu-shares</code>、<code>zone.cpu-shares</code>、および専用プロセッサセット中の最大プロセッサ数の値を指定します。</p> <p>専用プロセッサセット中のプロセッサセットの最小数を設定します。</p> <p>この手順は、CPU シェアとプロセッサセットのサイズを制御する場合に実行します。この制御は、専用プロセッサセットを使用することで、グローバルクラスタの非投票ノードでのみ行えます。</p>	<p><a href="#">288 ページの「専用プロセッサセットを使用してグローバルクラスタの非投票ノードで CPU 使用率を制御する方法」</a></p>

## 公平配分スケジューラ

CPU シェアをリソースグループに割り当てる手順の最初のステップは、システムのスケジューラを公平配分スケジューラ (FSS) に設定することです。デフォルトでは、Oracle Solaris OS のスケジューリングクラスはタイムシェアスケジューラ (TS) です。スケジューラを FSS に設定し、シェア構成を有効にします。

選択するスケジューラクラスに関係なく、専用のプロセッサセットを作成できます。

## CPU 制御の構成

このセクションでは次の作業について説明します。

- 283 ページの「グローバルクラスタの投票ノードで CPU 使用率を制御する方法」
- 285 ページの「デフォルトプロセッサセットを使用してグローバルクラスタの非投票ノードで CPU 使用率を制御する方法」
- 288 ページの「専用プロセッサセットを使用してグローバルクラスタの非投票ノードで CPU 使用率を制御する方法」

### ▼ グローバルクラスタの投票ノードで CPU 使用率を制御する方法

グローバルクラスタの投票ノードで実行されるリソースグループに CPU シェアを割り当てるには、この手順を実行します。

リソースグループに CPU シェアが割り当てられている場合、Oracle Solaris Cluster ソフトウェアは、グローバルクラスタの投票ノードでリソースグループの 1 つのリソースを起動する際に、次のタスクを実行します。

- 投票ノードに割り当てられている CPU シェア (`zone.cpu-shares`) の数を、指定された CPU シェアの数だけ増やします (まだ行われていない場合)。
- 投票ノードに `SCSLM_resourcegroup_name` という名前のプロジェクトを作成します (まだ行われていない場合)。このプロジェクトはリソースグループに固有で、指定された数の CPU シェア (`project.cpu-shares`) が割り当てられています。
- `SCSLM_resourcegroup_name` プロジェクトのリソースを起動する。

CPU 制御機能の構成についての詳細は、[rg\\_properties\(5\)](#) のマニュアルページを参照してください。

- 1 システムのデフォルトのスケジューラを、公平配分スケジューラ (FSS) に設定します。

```
dispadmin -d FSS
```

次のリブート時に、FSS がデフォルトのスケジューラになります。この構成をすぐに有効にするには、`priocntl` コマンドを使用します。

```
priocntl -s -C FSS
```

`priocntl` コマンドと `dispadmin` コマンドを組み合わせることで、FSS がすぐにデフォルトのスケジューラになり、リブート後もそのままになります。スケジューリングクラスの設定についての詳細は、[dispadmin\(1M\)](#) および [priocntl\(1\)](#) のマニュアルページを参照してください。

---

注-FSS がデフォルトのスケジューラでない場合、CPU シェアの割り当ては有効になりません。

---

- 2 各ノードで **CPU 制御** を使用するため、グローバルクラスタの投票ノードに対するシェア数と、デフォルトのプロセッサセットで使用可能な **CPU** の最小数を構成します。

これらのパラメータを設定すると、投票ノードで動作中のプロセスが、非投票ノードで動作中のプロセスと CPU で競合しないように保護するのに役立ちます。 `globalzoneshares` および `defaultpsetmin` プロパティに値を割り当てない場合、これらのプロパティはデフォルト値をとります。

```
clnode set [-p globalzoneshares=integer] \
[-p defaultpsetmin=integer] \
node
```

`-p defaultpsetmin= defaultpsetmininteger`      デフォルトプロセッサセットで使用可能な CPU の最小数を設定します。デフォルト値は 1 です。

`-p globalzoneshares= integer`      投票ノードに割り当てられるシェアの数を設定します。デフォルト値は 1 です。

`node`      プロパティを設定するノードを指定します。

これらのプロパティを設定する際には、投票ノードのプロパティを設定しています。これらのプロパティを設定しないと、非投票ノードで `RG_SLM_PSET_TYPE` プロパティの効果を達成することができません。

- 3 これらのプロパティを正しく設定したことを確認します。

```
clnode show node
```

指定するノードに対して、`clnode` コマンドは、設定されているプロパティ、およびこれらのプロパティに設定されている値を出力します。`clnode` を使用して CPU 制御プロパティを設定しないと、これらはデフォルト値をとります。

- 4 CPU 制御機能を構成します。

```
clresourcegroup create -p RG_SLM_TYPE=automated \
[-p RG_SLM_CPU_SHARES=value] resource_group_name
```

`-p RG_SLM_TYPE=automated`      CPU 使用率を管理できるようにし、システムリソース管理用に Oracle Solaris OS を構成する手順の一部を自動化します。

`-p RG_SLM_CPU_SHARES=value` リソースグループ固有のプロジェクトに割り当てられる CPU シェアの数 (`project.cpu-shares`) を指定し、投票ノードに割り当てられる CPU シェアの数 (`zone.cpu-shares`) を決定します。

`resource_group_name` リソースグループの名前を指定します。

この手順では、`RG_SLM_PSET_TYPE` プロパティは設定しません。投票ノードでは、このプロパティは値 `default` をとります。

このステップによりリソースグループが作成されます。また、`clresourcegroup set` コマンドを使用して既存のリソースグループを変更することもできます。

## 5 構成の変更を有効にします。

```
clresourcegroup online -eM resource_group_name
```

`resource_group_name` リソースグループの名前を指定します。

---

注-SCSLM `resource_group_name` プロジェクトは削除または変更しないでください。手動で、たとえば `project.max-lwps` プロパティを構成することにより、プロジェクトにさらにリソース制御を追加できます。詳細は、[projmod\(1M\)](#) のマニュアルページを参照してください。

---

## ▼ デフォルトプロセッサセットを使用してグローバルクラスタの非投票ノードで CPU 使用率を制御する方法

グローバルクラスタの非投票ノード内のリソースグループに CPU シェアを割り当てるが、専用プロセッサセットを作成する必要はない場合にこの手順を実行します。

リソースグループに CPU シェアが割り当てられている場合、Oracle Solaris Cluster ソフトウェアは、非投票ノードでそのリソースグループの 1 つのリソースを起動する際に、次のタスクを実行します。

- `SCSLM_resource_group_name` という名前のプールを作成します (まだ行われていない場合)。
- `SCSLM_pool_zone_name` プールをデフォルトプロセッサセットに関連付けます。
- 非投票ノードを `SCSLM_pool_zone_name` プールに動的にバインドします。
- 非投票ノードに割り当てられている CPU シェア (`zone.cpu-shares`) の数を、指定された CPU シェアの数だけ増やします (まだ行われていない場合)。

- 非投票ノードに `SCSLM_resourcegroup_name` という名前のプロジェクトを作成します (まだ行われていない場合)。このプロジェクトはリソースグループに固有で、指定された数の CPU シェア (`project.cpu-shares`) が割り当てられています。
- `SCSLM_resourcegroup_name` プロジェクトのリソースを起動する。

CPU 制御機能の構成についての詳細は、[rg\\_properties\(5\)](#) のマニュアルページを参照してください。

- 1 システムのデフォルトのスケジューラを、公平配分スケジューラ (**FSS**) に設定します。

```
dispadmin -d FSS
```

次のリブート時に、FSS がデフォルトのスケジューラになります。この構成をすぐに有効にするには、`priocntl` コマンドを使用します。

```
priocntl -s -C FSS
```

`priocntl` コマンドと `dispadmin` コマンドを組み合わせることで、FSS がすぐにデフォルトのスケジューラになり、リブート後もそのままになります。スケジューリングクラスの設定についての詳細は、[dispadmin\(1M\)](#) および [priocntl\(1\)](#) のマニュアルページを参照してください。

---

注-FSS がデフォルトのスケジューラでない場合、CPU シェアの割り当ては有効になりません。

---

- 2 各ノードで CPU 制御を使用するため、グローバルクラスタの投票ノードに対するシェア数と、デフォルトプロセッサセットで使用可能な CPU の最小数を構成します。

これらのパラメータを設定すると、投票ノードで動作中のプロセスが、グローバルクラスタの非投票ノードで動作中のプロセスと CPU で競合しないように保護するのに役立ちます。`globalzoneshares` および `defaultpsetmin` プロパティに値を割り当てない場合、これらのプロパティはデフォルト値をとります。

```
clnode set [-p globalzoneshares=integer] \
[-p defaultpsetmin=integer] \
node
```

```
-p globalzoneshares= integer
```

投票ノードに割り当てられるシェアの数を設定します。デフォルト値は 1 です。

```
-p defaultpsetmin= defaultpsetmininteger
```

デフォルトプロセッサセットで使用可能な CPU の最小数を設定します。デフォルト値は 1 です。

```
node
```

プロパティを設定するノードを指定します。

これらのプロパティを設定する際には、投票ノードのプロパティを設定しています。

- 3 これらのプロパティを正しく設定したことを確認します。

```
clnode show node
```

指定するノードに対して、`clnode` コマンドは、設定されているプロパティ、およびこれらのプロパティに設定されている値を出力します。`clnode` を使用して CPU 制御プロパティを設定しないと、これらはデフォルト値をとります。

- 4 CPU 制御機能を構成します。

```
clresourcegroup create -p RG_SLM_TYPE=automated \
 [-p RG_SLM_CPU_SHARES=value] resource_group_name
```

`-p RG_SLM_TYPE=automated` CPU 使用率を管理できるようにし、システムリソース管理用に Oracle Solaris OS を構成する手順の一部を自動化します。

`-p RG_SLM_CPU_SHARES=value` リソースグループ固有のプロジェクトに割り当てられる CPU シェアの数 (`project.cpu-shares`) を指定し、グローバルクラスタの非投票ノードに割り当てられる CPU シェアの数 (`zone.cpu_shares`) を決定します。

`resource_group_name` リソースグループの名前を指定します。

このステップによりリソースグループが作成されます。また、`clresourcegroup set` コマンドを使用して既存のリソースグループを変更することもできます。

ゾーン構成内にデフォルトプール以外のプールがある場合や、ゾーンがデフォルトプール以外のプールに動的にバインドされている場合は、非投票ノードで `RG_SLM_TYPE` を `automated` に設定することはできません。ゾーンの構成とプールのバインドについては、それぞれ [zonecfg\(1M\)](#) および [poolbind\(1M\)](#) のマニュアルページを参照してください。次のようにゾーンの構成を表示します。

```
zonecfg -z zone_name info pool
```

---

注 - 非投票ノードで起動するように構成された `HAStoragePlus` リソースや `LogicalHostname` リソースなどのリソースは、`GLOBAL_ZONE` プロパティが `TRUE` に設定されていると投票モードで起動されます。`RG_SLM_TYPE` プロパティを `automated` に設定した場合でも、このリソースは CPU シェア構成の効果が得られず、`RG_SLM_TYPE` が手動に設定されたリソースグループ内にあるように扱われます。

---

この手順では、`RG_SLM_PSET_TYPE` プロパティを設定しません。Oracle Solaris Cluster はデフォルトプロセッサセットを使用します。



## 5 構成の変更を有効にします。

```
clresourcegroup online -eM resource_group_name
```

`resource_group_name` リソースグループの名前を指定します。

RG\_SLM\_PSET\_TYPE を default に設定すると、Oracle Solaris Cluster はプール SCSLM\_pool\_zone\_name を作成しますが、プロセッサセットは作成しません。この場合、SCSLM\_pool\_zone\_name はデフォルトプロセッサセットに関連付けられます。

非投票ノードで、オンラインリソースグループの CPU 制御が構成されなくなった場合、非投票ノードの CPU シェアの値は、ゾーン構成内の zone.cpu-shares の値をとります。このパラメータの値はデフォルトで 1 です。ゾーン構成についての詳細は、[zonecfg\(1M\)](#) のマニュアルページを参照してください。

---

注 - SCSLM\_resource\_group\_name プロジェクトは削除または変更しないでください。たとえば project.max-lwps プロパティを構成することにより、手動でさらにプロジェクトにリソース制御を追加できます。詳細は、[projmod\(1M\)](#) のマニュアルページを参照してください。

---

## ▼ 専用プロセッサセットを使用してグローバルクラスタの非投票ノードで CPU 使用率を制御する方法

この手順は、リソースグループを専用プロセッサセットで実行する場合に実行します。

専用プロセッサセットで実行するようにリソースグループが構成されている場合、Oracle Solaris Cluster ソフトウェアは、グローバルクラスタの非投票ノードでリソースグループのリソースを起動する際に、次のタスクを実行します。

- SCSLM\_pool\_zone\_name という名前のプールを作成します (まだ行われていない場合)。
- 専用プロセッサセットを作成します。プロセッサセットのサイズは、RG\_SLM\_CPU\_SHARES プロパティと RG\_SLM\_PSET\_MIN プロパティを使用して決定します。
- 作成したプロセッサセットに、SCSLM\_pool\_zone\_name プールを関連付けます。
- 非投票ノードを SCSLM\_pool\_zone\_name プールに動的にバインドします。
- 非投票ノードに割り当てられている CPU シェアの数、指定された CPU シェアの数だけ増やします (まだ行われていない場合)。
- 非投票ノードに SCSLM\_resource\_group\_name という名前のプロジェクトを作成します (まだ行われていない場合)。このプロジェクトはリソースグループに固有で、指定された数の CPU シェア (project.cpu-shares) が割り当てられています。



- `SCSLM_resourcegroup_name` プロジェクトのリソースを起動する。

## 1 システムのスケジューラを、公平配分スケジューラ (FSS) に設定します。

```
dispadmin -d FSS
```

次のリブート時に、FSSがデフォルトのスケジューラになります。この構成をすぐに有効にするには、`priocntl` コマンドを使用します。

```
priocntl -s -C FSS
```

`priocntl` コマンドと `dispadmin` コマンドを組み合わせることで、FSSがすぐにデフォルトのスケジューラになり、リブート後もそのままになります。スケジューリングクラスの設定についての詳細は、[dispadmin\(1M\)](#) および [priocntl\(1\)](#) のマニュアルページを参照してください。

---

注-FSSがデフォルトのスケジューラでない場合、CPUシェアの割り当ては有効になりません。

---

## 2 各ノードでCPU制御を使用するため、グローバルクラスタの投票ノードに対するシェア数と、デフォルトプロセッサセットで使用可能なCPUの最小数を構成します。

これらのパラメータを設定すると、投票ノードで動作中のプロセスが、非投票ノードで動作中のプロセスとCPUで競合しないように保護するのに役立ちます。`globalzonestshares` および `defaultpsetmin` プロパティに値を割り当てない場合、これらのプロパティはデフォルト値をとります。

```
clnode set [-p globalzonestshares=integer] \
[-p defaultpsetmin=integer] \
node
```

`-p defaultpsetmin= defaultpsetmininteger`      デフォルトプロセッサセットで使用可能なCPUの最小数を設定します。デフォルトは1です。

`-p globalzonestshares= integer`      投票ノードに割り当てられるシェアの数を設定します。デフォルトは1です。

`node`      プロパティを設定するノードを指定します。

これらのプロパティを設定する際には、投票ノードのプロパティを設定しています。

## 3 これらのプロパティを正しく設定したことを確認します。

```
clnode show node
```

指定するノードに対して、`clnode` コマンドは、設定されているプロパティ、およびこれらのプロパティに設定されている値を出力します。`clnode` を使用して CPU 制御プロパティを設定しないと、これらはデフォルト値をとります。

#### 4 CPU 制御機能を構成します。

```
clresourcegroup create -p RG_SLM_TYPE=automated \
 [-p RG_SLM_CPU_SHARES=value] \
 -p -y RG_SLM_PSET_TYPE=value \
 [-p RG_SLM_PSET_MIN=value] resource_group_name
```

<code>-p RG_SLM_TYPE=automated</code>	CPU 使用率を管理できるようにし、システムリソース管理用に Oracle Solaris OS を構成する手順の一部を自動化します。
<code>-p RG_SLM_CPU_SHARES=value</code>	リソースグループ固有のプロジェクトに割り当てられる CPU シェアの数 ( <code>project.cpu-shares</code> ) を指定し、非投票ノードに割り当てられる CPU シェアの数 ( <code>zone.cpu-shares</code> ) と、プロセッサセット内の最大プロセッサ数を決定します。
<code>-p RG_SLM_PSET_TYPE= value</code>	専用のプロセッサセットの作成を可能にします。専用プロセッサセットを使用する場合は、このプロパティを <code>strong</code> または <code>weak</code> に設定できます。値 <code>strong</code> と値 <code>weak</code> は同時に指定できません。つまり、一部のリソースグループが <code>strong</code> で、それ以外が <code>weak</code> であるように、同じゾーン内のリソースグループを構成することはできません。
<code>-p RG_SLM_PSET_MIN= value</code>	プロセッサセット中のプロセッサの最小数を決定します。
<code>resource_group_name</code>	リソースグループの名前を指定します。

このステップによりリソースグループが作成されます。また、`clresourcegroup set` コマンドを使用して既存のリソースグループを変更できます。

ゾーン構成内にデフォルトプール以外のプールがある場合や、ゾーンがデフォルトプール以外のプールに動的にバインドされている場合は、非投票ノードで `RG_SLM_TYPE` を `automated` に設定することはできません。ゾーンの構成とプールのバインドについては、それぞれ [zonecfg\(1M\)](#) および [poolbind\(1M\)](#) のマニュアルページを参照してください。次のようにゾーンの構成を表示します。

```
zonecfg -z zone_name info pool
```

---

注-非投票ノードで起動するように構成された HASToragePlus リソースや LogicalHostname リソースなどのリソースは、GLOBAL\_ZONE プロパティが TRUE に設定されていると投票モードで起動されます。RG\_SLM\_TYPE プロパティを automated に設定した場合でも、このリソースは CPU シェアおよび専用プロセッサセットの構成の効果が得られず、RG\_SLM\_TYPE が手動に設定されたリソースグループ内にあるように扱われます。

---

## 5 構成の変更を有効にします。

```
clresourcegroup online -eM resource_group_name
```

`resource_group_name` リソースグループの名前を指定します。

---

注-SCSLM\_resource\_group\_name プロジェクトは削除または変更しないでください。たとえば `project.max-lwps` プロパティを構成することにより、手動でさらにプロジェクトにリソース制御を追加できます。詳細は、[projmod\(1M\)](#) のマニュアルページを参照してください。

---

リソースグループがオンラインの間に RG\_SLM\_CPU\_SHARES と RG\_SLM\_PSET\_MIN に行われた変更は、動的に考慮されます。しかし、RG\_SLM\_PSET\_TYPE に strong が設定されている場合、および、変更に対応するための十分な CPU を使用できない場合、RG\_SLM\_PSET\_MIN にリクエストされた変更は適用されません。この場合は、警告メッセージが表示されます。次のスイッチオーバーでは、RG\_SLM\_PSET\_MIN で構成した値を適用するのに十分な CPU を使用できない場合、不十分な CPU が理由のエラーが発生する可能性があります。

非投票ノードで、オンラインリソースグループの CPU 制御が構成されなくなった場合、非投票ノードの CPU シェアの値は `zone.cpu-shares` の値をとります。このパラメータの値はデフォルトで 1 です。



# Oracle Solaris Cluster ソフトウェアおよび ファームウェアのパッチ適用

---

この章では、Oracle Solaris Cluster 構成のためにパッチの追加と削除を行う手順を説明します。手順は次のセクションに含まれています。

- 293 ページの「Oracle Solaris Cluster パッチ適用の概要」
- 295 ページの「Oracle Solaris Cluster ソフトウェアのパッチ適用」

## Oracle Solaris Cluster パッチ適用の概要

クラスタの特性のため、クラスタが適切に動作するためには、すべてのクラスタメンバーノードが同じパッチレベルにある必要があります。場合によっては、Oracle Solaris Cluster でノードにパッチを適用するときに、クラスタメンバーシップからノードを一時的に削除するか、クラスタ全体を停止してからパッチをインストールする必要があることもあります。このセクションでは、これらの手順について説明します。

Oracle Solaris Cluster パッチを適用する前に、パッチの README ファイルを確認します。また、ストレージデバイスのアップグレード要件を確認し、どのパッチの方法が必要かを判別します。

---

注 - Oracle Solaris Cluster のパッチについては、必ずパッチの README ファイルと SunSolve の手順に従ってください。これらはこの章の手順に優先します。

---

すべてのクラスタノードでのパッチのインストールは、次のいずれかのシナリオで説明できます。

リブートするパッチ(ノード)

パッチまたはファームウェアを適用する前に、コマンド `boot -sx` または `shutdown -g -y -i0` を使用して、ノードをシングルユーザーモードでブートしてから、クラスタに参加するようにリブートする必要があります。最初に、パッチを適用する

ノードのリソースグループまたはデバイスグループをすべて別のクラスタメンバーに切り替えることによって、ノードを「停止」状態にする必要があります。また、パッチまたはファームウェアを1回につき1つのクラスタノードに適用し、クラスタ全体が停止するのを回避します。

個々のノードは一時的に使用できませんが、クラスタ自体は、このタイプのパッチの適用中も使用可能なままです。パッチを適用したノードは、ほかのノードがまだ同じパッチレベルになくても、メンバーノードとしてクラスタに再参加できます。

#### リブートするパッチ(クラスタ)

ソフトウェアまたはファームウェアパッチを適用するには、コマンド `boot -sx` または `shutdown -g -y -i0` を使用して、クラスタを停止し、各ノードをシングルユーザーモードでブートする必要があります。その後、ノードをリブートしてクラスタに再参加します。このタイプのパッチの場合、パッチの適用中にクラスタを使用することはできません。

#### リブートしないパッチ

ノードが「停止」状態になっている必要はなく(リソースグループまたはデバイスグループをマスターし続けることができる)、パッチ適用時にリブートする必要もありません。ただし、1回につき1つのノードにパッチを適用し、別のノードに適用する前にパッチが機能していることを確認する必要があります。

---

注- 基になるクラスタプロトコルがパッチのために変更されることはありません。

---

`patchadd` コマンドを使用してパッチをクラスタに適用し、`patchrm` を使用してパッチを削除します(可能な場合)。

## Oracle Solaris Cluster パッチのヒント

Oracle Solaris Cluster パッチをより効率的に管理するために、次のヒントを使用してください。

- パッチを適用する前に、必ずパッチの README ファイルを読みます。

- ストレージデバイスのアップグレード要件を確認し、どのパッチの方法が必要かを判別します。
- 本稼働環境でクラスタを実行する前に、すべてのパッチ (必須および推奨) を適用します。
- ハードウェアのファームウェアレベルを確認し、必要と思われる必須ファームウェアアップデートをインストールします。
- クラスタメンバーとして機能するノードには、すべて同じパッチを適用する必要があります。
- クラスタサブシステムのパッチを最新の状態に保ちます。これらのパッチには、ボリューム管理、ストレージデバイスのファームウェア、クラスタトランスポートなどが含まれます。
- 四半期に一度など、定期的にパッチレポートを確認し、推奨されるパッチスイートを使用して、Oracle Solaris Cluster 構成にパッチを適用します。
- Enterprise Services で推奨されるとおり、選択されたパッチを適用します。
- メジャーパッチの更新後はフェイルオーバーをテストします。クラスタの動作が低下または悪化した場合に備えて、パッチを取り消す準備をしておきます。

# Oracle Solaris Cluster ソフトウェアのパッチ適用

表 11-1 タスクマップ: クラスタのパッチ適用

タスク	手順
ノードを停止せずに、リブートしない Oracle Solaris Cluster パッチを 1 回につき 1 つのノードに適用する	<a href="#">304 ページの「リブートしない Oracle Solaris Cluster パッチを適用する方法」</a>
クラスタメンバーを非クラスタモードにしてから、リブートする Oracle Solaris Cluster パッチを適用する	<a href="#">296 ページの「リブートするパッチを適用する方法 (ノード)」</a> <a href="#">300 ページの「リブートするパッチを適用する方法 (クラスタ)」</a>
フェイルオーバーゾーンを使用するノードにはシングルユーザーモードでパッチを適用する	<a href="#">305 ページの「フェイルオーバーゾーンを使用するノードにシングルユーザーモードでパッチを適用する方法」</a>
Oracle Solaris Cluster パッチを削除する	<a href="#">309 ページの「Oracle Solaris Cluster パッチの変更」</a>

## ▼ リブートするパッチを適用する方法(ノード)

パッチプロセス中にクラスタ自体を動作可能に保つため、クラスタ内では1回につき1つのノードにパッチを適用します。この手順では、最初にクラスタ内のノードを停止し、`boot -sx` または `shutdown -g -y -i0` コマンドを使用してシングルユーザーモードでノードをブートしてから、パッチを適用します。

`phys-schost#` プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 パッチを適用する前に、インストール前またはインストール後の特別な指示がないか、**Oracle Solaris Cluster 製品 Web サイト**を確認してください。
- 2 パッチを適用するノードで、スーパーユーザーになるか、**RBAC**の承認 `solaris.cluster.admin` を提供する役割になります。
- 3 パッチが適用されるノードで、リソースグループとデバイスグループを一覧表示します。

```
clresourcegroup status -Z all -n node[...]
```

`node`      パッチが適用されるノード上にあるグローバルクラスタノードまたはゾーンクラスタノードの名前。

```
cldevicegroup status -n node
```

`node`      パッチが適用されるグローバルクラスタノードの名前。

---

注-デバイスグループはゾーンクラスタに関連付けられていません。

---

- 4 すべてのリソースグループ、リソース、およびデバイスグループを、パッチを適用するノードから別のクラスタメンバーに切り替えます。

```
clnode evacuate -n node
```

`evacuate`      すべてのデバイスグループとリソースグループ(すべてのグローバルクラスタ非投票ノードを含む)を待避させます。

`-n node`      リソースグループとデバイスグループの切り替え元のノードを指定します。

- 5 ノードを停止します。

```
shutdown -g0 [-y]
[-i0]
```



6 非クラスタのシングルユーザーモードでノードをブートします。

- SPARC ベースのシステム上で、次のコマンドを実行します。

```
ok boot -sx
```

- x86 ベースのシステム上で、次のコマンドを実行します。

```
phys-schost# shutdown -g -y -i0
```

Press any key to continue

- a. GRUB メニューで矢印キーを使用して該当する **Oracle Solaris** エントリを選択し、**e** と入力してコマンドを編集します。

次のような GRUB メニューが表示されます。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
```

```
+-----+
| Solaris 10 /sol_10_x86 |
| Solaris failsafe |
| |
+-----+
```

Use the ^ and v keys to select which entry is highlighted.  
Press enter to boot the selected OS, 'e' to edit the  
commands before booting, or 'c' for a command-line.

GRUB ベースのブートの詳細については、『[Oracle Solaris の管理: 基本管理](#)』の「[GRUB を使用して x86 システムをブートする \(タスクマップ\)](#)」を参照してください。

- b. ブートパラメータの画面で、矢印キーを使用してカーネルエントリを選択し、**e** を入力してエントリを編集します。

次のような GRUB ブートパラメータの画面が表示されます。

```
GNU GRUB version 0.95 (615K lower / 2095552K upper memory)
```

```
+-----+
| root (hd0,0,a) |
| kernel /platform/i86pc/multiboot |
| module /platform/i86pc/boot_archive |
+-----+
```

Use the ^ and v keys to select which entry is highlighted.  
Press 'b' to boot, 'e' to edit the selected command in the  
boot sequence, 'c' for a command-line, 'o' to open a new line  
after ('O' for before) the selected line, 'd' to remove the  
selected line, or escape to go back to the main menu.

- c. コマンドに **-sx** を追加して、システムが非クラスタモードでブートするよう指定します。

```
[Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time exits.]
```

```
grub edit> kernel /platform/i86pc/multiboot -sx
```

- d. **Enter** キーを押して変更を受け入れ、ブートパラメータの画面に戻ります。

画面には編集されたコマンドが表示されます。

```
GNU GRUB version 0.95 (615K lower / 2095552K upper memory)
+-----+
| root (hd0,0,a) |
| kernel /platform/i86pc/multiboot -sx |
| module /platform/i86pc/boot_archive |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.-
```

- e. **b** と入力して、ノードを非クラスタモードでブートします。

---

注-カーネルブートパラメータコマンドへのこの変更は、システムをブートすると無効になります。次にノードをリブートする際には、ノードはクラスタモードでブートします。そうでなく非クラスタモードでブートするには、これらの手順を再度実行して、カーネルブートパラメータコマンドに **-sx** オプションを追加します。

---

- 7 ソフトウェアパッチまたはファームウェアパッチを適用します。

```
patchadd -M patch-dir patch-id
```

*patch-dir*                   パッチのディレクトリの場所を指定します。

*patch-id*                    特定のパッチのパッチ番号を指定します。

---

注-必ずパッチディレクトリにある指示に従ってください。それらがこの章の手順に優先します。

---

- 8 パッチが正常にインストールされたことを確認します。

```
showrev -p | grep patch-id
```

- 9 ノードをリブートしてクラスタに戻します。

```
reboot
```

- 10 パッチが機能していること、およびノードとクラスタが正常に動作していることを確認します。

- 11 残りのクラスタノードすべてに対して、[手順2](#)から[手順10](#)を繰り返します。

- 12 必要に応じてリソースグループとデバイスグループを切り替えます。

すべてのノードをリブートしたあと、最後にリブートしたノードにはオンラインのリソースグループとデバイスグループがありません。

```
cldevicegroup switch -n node + | devicegroup ...
clresourcegroup switch -n node[:zone][,...] + | resource-group ...
```

*node* リソースグループとデバイスグループの切り替え先ノードの名前。

*zone* リソースグループをマスターできるグローバルクラスタの非投票ノード (node) の名前。リソースグループを作成したときに非投票ノードを指定した場合にのみ、ゾーンを指定します。

```
clresourcegroup switch -Z zoneclustername -n zcnode[...] + | resource-group ...
```

*zoneclustername* リソースグループの切り替え先ゾーンクラスタの名前。

*zcnode* リソースグループをマスターできるゾーンクラスタの名前。

---

注-デバイスグループはゾーンクラスタに関連付けられていません。

---

- 13 **scversions** コマンドを使用して、パッチソフトウェアをコミットする必要があるかどうかを確認します。

```
/usr/cluster/bin/scversions
```

次のいずれかの結果が表示されます。

```
Upgrade commit is needed.
```

```
Upgrade commit is NOT needed. All versions match.
```

- 14 コミットが必要な場合は、パッチソフトウェアをコミットします。

```
scversions -c
```

---

注-scversions を実行すると、状況に応じて1つ以上のCMM再構成が発生します。

---

### 例 11-1 リブートするパッチの適用 (ノード)

次に、リブートする Oracle Solaris Cluster パッチをノードに適用する例を示します。

```
clresourcegroup status -n rg1
...Resource Group Resource

rg1 rs-2
rg1 rs-3
...
cldevicegroup status -n nodedg-schost-1
...
```

```
Device Group Name: dg-schost-1
```

```
...
clnode evacuate phys-schost-2
shutdown -g0 -y -i0
...
```

非クラスタのシングルユーザーモードでノードをブートします。

- SPARC: 次のように入力します。

```
ok boot -sx
```

- x86: 非クラスタのシングルユーザーモードでノードをブートします。後続の手順でブート手順を参照してください。

```
patchadd -M /var/tmp/patches 234567-05
...
showrev -p | grep 234567-05

...
reboot
...
cldevicegroup switch -n phys-schost-1 dg-schost-1
clresourcegroup switch -n phys-schost-1 schost-sa-1
scversions
Upgrade commit is needed.
scversions -c
```

参照 パッチの適用を取り消す必要がある場合は、[309 ページの「Oracle Solaris Cluster パッチの変更」](#)を参照してください。

## ▼ リブートするパッチを適用する方法(クラスタ)

この手順では、最初にクラスタを停止し、`boot -sx` または `shutdown -g -y -i0` コマンドを使用してシングルユーザーモードで各ノードをブートしてから、パッチを適用します。

- 1 パッチを適用する前に、インストール前またはインストール後の特別な指示がないか、**Oracle Solaris Cluster** 製品 **Web** サイトを確認してください。
- 2 クラスタ内にある任意のノード上でスーパーユーザーになります。
- 3 クラスタを停止します。

```
cluster shutdown -y -g grace-period "message"
```

-y 確認のプロンプトに、回答 `yes` を指定します。

-g `grace-period` シャットダウンする前に待機する時間を秒単位で指定します。デフォルトの猶予期間は 60 秒です。

*message*                     ブロードキャストする警告メッセージを指定します。*message* に複数の単語が含まれる場合は引用符を使用します。

- 4 各ノードを非クラスタのシングルユーザーモードでブートします。  
各ノードのコンソールで、次のコマンドを実行します。

- SPARC ベースのシステム上で、次のコマンドを実行します。

```
ok boot -sx
```

- x86 ベースのシステム上で、次のコマンドを実行します。

```
phys-schost# shutdown -g -y -i0
```

```
Press any key to continue
```

- a. GRUB メニューで矢印キーを使用して該当する **Oracle Solaris** エントリを選択し、**e** と入力してコマンドを編集します。

次のような GRUB メニューが表示されます。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
```

```
+-----+
| Solaris 10 /sol_10_x86 |
| Solaris failsafe |
| |
+-----+
```

Use the ^ and v keys to select which entry is highlighted.

Press enter to boot the selected OS, 'e' to edit the  
commands before booting, or 'c' for a command-line.

GRUB ベースのブートの詳細については、『[Oracle Solaris の管理: 基本管理](#)』の「[GRUB を使用して x86 システムをブートする \(タスクマップ\)](#)」を参照してください。

- b. ブートパラメータの画面で、矢印キーを使用してカーネルエントリを選択し、**e** と入力してエントリを編集します。

次のような GRUB ブートパラメータの画面が表示されます。

```
GNU GRUB version 0.95 (615K lower / 2095552K upper memory)
```

```
+-----+
| root (hd0,0,a) |
| kernel /platform/i86pc/multiboot |
| module /platform/i86pc/boot_archive |
+-----+
```

Use the ^ and v keys to select which entry is highlighted.

Press 'b' to boot, 'e' to edit the selected command in the  
boot sequence, 'c' for a command-line, 'o' to open a new line  
after ('O' for before) the selected line, 'd' to remove the  
selected line, or escape to go back to the main menu.

- c. コマンドに **-sx** を追加して、システムが非クラスタモードでブートするよう指定します。

[ Minimal BASH-like line editing is supported. For the first word, TAB lists possible command completions. Anywhere else TAB lists the possible completions of a device/filename. ESC at any time exits. ]

```
grub edit> kernel /platform/i86pc/multiboot -sx
```

- d. **Enter** キーを押して変更を受け入れ、ブートパラメータの画面に戻ります。

画面には編集されたコマンドが表示されます。

```
GNU GRUB version 0.95 (615K lower / 2095552K upper memory)
+-----+
| root (hd0,0,a) |
| kernel /platform/i86pc/multiboot -sx |
| module /platform/i86pc/boot_archive |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.-
```

- e. **b** と入力して、ノードを非クラスタモードでブートします。

---

注-カーネルブートパラメータコマンドへのこの変更は、システムをブートすると無効になります。次にノードをリブートする際には、ノードはクラスタモードでブートします。そうでなく非クラスタモードでブートするには、これらの手順を再度実行して、カーネルブートパラメータコマンドに **-sx** オプションを追加します。

---

- 5 ソフトウェアパッチまたはファームウェアパッチを適用します。

1 回につき 1 つのノードで、次のコマンドを実行します。

```
patchadd -M patch-dir patch-id
```

*patch-dir*                      パッチのディレクトリの場所を指定します。

*patch-id*                      特定のパッチのパッチ番号を指定します。

---

注-必ずパッチディレクトリにある指示に従ってください。それらがこの章の手順に優先します。

---

- 6 各ノードにパッチが正常にインストールされたことを確認します。

```
showrev -p | grep patch-id
```

- 7 パッチをすべてのノードに適用したら、ノードをクラスタ内でリブートします。  
各ノードで次のコマンドを実行します。  
**# reboot**
- 8 **scversions** コマンドを使用して、パッチソフトウェアをコミットする必要があるかどうかを確認します。  
**# /usr/cluster/bin/scversions**  
次のいずれかの結果が表示されます。  
  
Upgrade commit is needed.  
  
Upgrade commit is NOT needed. All versions match.
- 9 コミットが必要な場合は、パッチソフトウェアをコミットします。  
**# scversions -c**

---

注 - **scversions** を実行すると、状況に応じて1つ以上のCMM再構成が発生します。

---

- 10 パッチが機能していること、およびノードとクラスタが正常に動作していることを確認します。

## 例 11-2 リブートするパッチの適用 (クラスタ)

次に、リブートする Oracle Solaris Cluster パッチをクラスタに適用する例を示します。

```
cluster shutdown -g0 -y
...
```

非クラスタのシングルユーザーモードでクラスタをブートします。

- SPARC: 次のように入力します。

```
ok boot -sx
```

- x86: 各ノードを非クラスタのシングルユーザーモードでブートします。後続の手順で手順を参照してください。

```
...
patchadd -M /var/tmp/patches 234567-05
(Apply patch to other cluster nodes)
...
showrev -p | grep 234567-05
reboot
scversions
Upgrade commit is needed.
scversions -c
```

参照 パッチの適用を取り消す必要がある場合は、309 ページの「[Oracle Solaris Cluster パッチの変更](#)」を参照してください。

## ▼ リブートしない Oracle Solaris Cluster パッチを適用する方法

クラスタ内で1回につき1つのノードにパッチを適用します。リブートしないパッチを適用する場合、パッチを受け取るノードを最初に停止する必要はありません。

- 1 パッチを適用する前に、インストール前またはインストール後の特別な指示がないか、**Oracle Solaris Cluster 製品 Web** ページを確認してください。

- 2 1つのノードでパッチを適用します。

```
patchadd -M patch-dir patch-id
```

*patch-dir*                      パッチのディレクトリの場所を指定します。

*patch-id*                      特定のパッチのパッチ番号を指定します。

- 3 パッチが正常にインストールされたことを確認します。

```
showrev -p | grep patch-id
```

- 4 パッチが機能していること、およびノードとクラスタが正常に動作していることを確認します。

- 5 残りのクラスタノードに対して、[手順2](#)から[手順4](#)を繰り返します。

- 6 **scversions** コマンドを使用して、パッチソフトウェアをコミットする必要があるかどうかを確認します。

```
/usr/cluster/bin/scversions
```

次のいずれかの結果が表示されます。

```
Upgrade commit is needed.
```

```
Upgrade commit is NOT needed. All versions match.
```

- 7 コミットが必要な場合は、パッチソフトウェアをコミットします。

```
scversions -c
```

---

注 – **scversions** を実行すると、状況に応じて1つ以上の CMM 再構成が発生します。

---



## 例 11-3 リブートしない Oracle Solaris Cluster パッチの適用

```
patchadd -M /tmp/patches 234567-05
...
showrev -p | grep 234567-05
scversions
Upgrade commit is needed.
scversions -c
```

参照 パッチの適用を取り消す必要がある場合は、[309 ページ](#)の「[Oracle Solaris Cluster パッチの変更](#)」を参照してください。

## ▼ フェイルオーバーゾーンを使用するノードにシングルユーザーモードでパッチを適用する方法

フェイルオーバーゾーンの使用時にシングルユーザーモードでパッチを適用するには、このタスクを実行します。このパッチ方法が必要とされるのは、Oracle Solaris Cluster ソフトウェアを使用するフェイルオーバー構成で Solaris コンテナ用に Oracle Solaris Cluster データサービスを使用している場合です。

- 1 この手順で手動により取得したゾーンパスを含むディスクセットの一部である共有ストレージとして使用される LUN の 1 つに対して、定足数デバイスが構成されていないことを確認します。

- a. ゾーンパスを含むディスクセットで定足数デバイスが使用されているかどうかを確認し、定足数デバイスが **SCSI2** または **SCSI3** 予約を使用するかどうかを確認します。

```
clquorum show
```

- b. 定足数デバイスがディスクセットの **LUN** 内にある場合、新しい **LUN** を、ゾーンパスを含むどのディスクセットの一部にもなっていない定足数デバイスとして追加します。

```
clquorum add new-didname
```

- c. 古い定足数デバイスを削除します。

```
clquorum remove old-didname
```

- d. **SCSI2** 予約が古い定足数デバイスのために使用されている場合、古い定足数から **SCSI2** 予約をスクラブして、**SCSI2** 予約が残っていないことを確認します。

次のコマンドは、PGRE (Persistent Group Reservation Emulation) 鍵を検索します。ディスク上に鍵が存在しない場合は、`errno=22` メッセージが表示されます。

```
/usr/cluster/lib/sc/pgre -c pgre_inkeys -d /dev/did/rdisk/dids2
```

鍵が見つかったら、PGRE 鍵をスクラブします。

```
/usr/cluster/lib/sc/pgre -c pgre_scrub -d /dev/did/rdisk/dids2
```



注意-アクティブな定足数デバイス鍵をディスクからスクラブすると、次の再構成時にクラスタでパニックが発生し、「操作可能な定足数を失いました」というメッセージが表示されます。

- 2 パッチを適用するノードを退避させます。

```
clresourcegroup evacuate -n node1
```

- 3 リソースグループ、または HA Solaris コンテナリソースを含むリソースグループをオフラインにします。

```
clresourcegroup offline resourcegroupname
```

- 4 オフラインにしたリソースグループ内のすべてのリソースを無効にします。

```
clresource disable resourcename
```

- 5 オフラインにしたリソースグループを非管理状態にします。

```
clresourcegroup unmanage resourcegroupname
```

- 6 対応する1つまたは複数のデバイスグループをオフラインにします。

```
cldevicegroup offline cldevicegroupname
```

注-ゾーンパス用の zpool があるフェイルオーバーゾーンにパッチを適用する場合は、この手順と手順7をスキップします。

- 7 オフラインにしたデバイスグループを無効にします。

```
cldevicegroup disable devicegroupname
```

- 8 クラスタにあるパッシブノードをブートします。

```
reboot -- -x
```

注-ゾーンパス用の zpool があるフェイルオーバーゾーンにパッチを適用する場合は、次のコマンドを使用します。

```
reboot -- -xs
```

- 9 続ける前にパッシブノードで SMF 起動メソッドが完了していることを確認します。

```
svcs -x
```

---

注- ゾーンバス用の `zpool` があるフェイルオーバーゾーンにパッチを適用する場合は、この手順をスキップします。

---

- 10 アクティブノードですべての再構成プロセスが完了したことを確認します。

```
cluster status
```

- 11 ディスクセット内のディスクに **SCSI-2** 予約が存在するかどうかを調べ、キーを解放します。**SCSI-2** 予約が存在するかどうかを調べる次の手順に従ってから、予約を解放します。

- ディスクセット内のすべてのディスクに対して、次のコマンドを実行します:  
`/usr/cluster/lib/sc/scsi -c disfailfast -d /dev/did/rdisk/d#s2`
- キーが一覧表示される場合は、次のコマンドを実行してキーを解放します:  
`/usr/cluster/lib/sc/scsi -c release -d /dev/did/rdisk/d#s2`

予約キーの解放が終了したら、手順 12 をスキップして手順 13 に進みます。

- 12 ディスクセット内のディスクに **SCSI3** 予約があるかどうかを調べます。

- a. ディスクセットのすべてのディスクで次のコマンドを実行します。

```
/usr/cluster/lib/sc/scsi -c inkeys -d /dev/did/rdisk/didnames2
```

- b. キーが一覧表示された場合は、それらをスクラブします。

```
/usr/cluster/lib/sc/scsi -c scrub -d /dev/did/rdisk/didnames2
```

- 13 パッシブノード上のメタセットの所有権を取得します。

```
metaset -s disksetname -C take -f
```

---

注- ゾーンバス用の `zpool` があるフェイルオーバーゾーンにパッチを適用する場合は、次のコマンドを使用します。

---

```
zpool import -R / pool_name
```

---

- 14 パッシブノード上の、ゾーンバスが含まれる 1 つまたは複数のファイルシステムをマウントします。

```
mount device mountpoint
```

---

注- ゾーンバス用の `zpool` があるフェイルオーバーゾーンにパッチを適用する場合は、この手順と [手順 15](#) をスキップします。

---

- 15 パッシブノードでシングルユーザーモードに切り替えます。

```
init s
```

- 16 Solaris コンテナ用の Oracle Solaris Cluster データサービスの制御下でない可能性があるブート済みゾーンをすべて停止します。

```
zoneadm -z zonename halt
```

- 17 (省略可能) 複数のパッチをインストールする場合は、パフォーマンス上の理由で、シングルユーザーモードの構成済みゾーンをすべてブートすることを選択できます。

```
zoneadm -z zonename boot -s
```

- 18 パッチを適用します。

- 19 ノードをリブートし、SMF 起動メソッドが終了するまで待機します。svcs -a コマンドは、ノードがリブートされたあとにのみ実行します。

```
reboot
```

```
svcs -a
```

最初のノードの準備ができました。

- 20 パッチを適用する 2 番目のノードを退避させます。

```
clresourcegroup evacuate -n node2
```

- 21 2 番目のノードに対して手順 8 から 13 を繰り返します。

- 22 パッチを適用したゾーンを切り離します。すでにパッチを適用したゾーンを切り離さないと、パッチプロセスが失敗します。

```
zoneadm -z zonename detach
```

- 23 パッシブノードでシングルユーザーモードに切り替えます。

```
init s
```

- 24 Solaris コンテナ用の Oracle Solaris Cluster データサービスの制御下でない可能性があるブート済みゾーンをすべて停止します。

```
zoneadm -z zonename halt
```

- 25 (省略可能) 複数のパッチをインストールする場合は、パフォーマンス上の理由で、シングルユーザーモードの構成済みゾーンをすべてブートすることを選択できます。

```
zoneadm -z zonename boot -s
```

- 26 パッチを適用します。

- 27 切り離れたゾーンを接続します。
- ```
# zoneadm -z zonename attach -F
```
- 28 ノードをリブートしてクラスタモードにします。
- ```
reboot
```
- 29 1つまたは複数のデバイスグループをオンラインにします。
- 30 リソースグループを起動します。
- 31 **scversions** コマンドを使用して、パッチソフトウェアをコミットする必要があるかどうかを確認します。
- ```
# /usr/cluster/bin/scversions
```
- 次のいずれかの結果が表示されます。
- ```
Upgrade commit is needed.
```
- ```
Upgrade commit is NOT needed. All versions match.
```
- 32 コミットが必要な場合は、パッチソフトウェアをコミットします。
- ```
scversions -c
```

---

注 - **scversions** を実行すると、状況に応じて1つ以上のCMM再構成が発生します。

---

## Oracle Solaris Cluster パッチの変更

クラスタに適用した Oracle Solaris Cluster パッチを削除するには、最初に新しい Oracle Solaris Cluster パッチを削除してから、以前のパッチまたはアップデートリリースを再適用します。新しい Oracle Solaris Cluster パッチを削除する場合は次の手順を参照してください。以前の Oracle Solaris Cluster パッチを再適用する場合は、次のいずれかの手順を参照してください。

- [296 ページの「リブートするパッチを適用する方法\(ノード\)」](#)
- [300 ページの「リブートするパッチを適用する方法\(クラスタ\)」](#)
- [304 ページの「リブートしない Oracle Solaris Cluster パッチを適用する方法」](#)

---

注 - Oracle Solaris Cluster パッチを適用する前に、パッチの README ファイルを確認します。

---

## ▼ リブートしない **Oracle Solaris Cluster** パッチを削除する方法

- 1 クラスタ内にある任意のノード上でスーパーユーザーになります。
- 2 リブートしないパッチを削除します。

```
patchrm patchid
```

## ▼ リブートする **Oracle Solaris Cluster** パッチを削除する方法

- 1 クラスタ内にある任意のノード上でスーパーユーザーになります。
- 2 クラスタノードを非クラスタモードでブートします。ノードを非クラスタモードでブートする方法については、[82 ページの「非クラスタモードでノードをブートする方法」](#)を参照してください。

- 3 リブートするパッチを削除します。

```
patchrm patchid
```

- 4 クラスタノードをリブートしてクラスタモードに戻します。

```
reboot
```

- 5 クラスタノードごとに手順 2 から 4 を繰り返します。

## クラスタのバックアップと復元

---

この章は次のセクションから構成されています。

- [311 ページの「クラスタのバックアップ」](#)
- [319 ページの「クラスタファイルの復元」](#)

### クラスタのバックアップ

表 12-1 タスクリスト: クラスタファイルのバックアップ

タスク	手順
バックアップするファイルシステムの名前の検索	<a href="#">312 ページの「バックアップするファイルシステム名を検索する方法」</a>
フルバックアップを格納するのに必要なテープ数の計算	<a href="#">312 ページの「完全バックアップに必要なテープ数を調べる方法」</a>
ルートファイルシステムのバックアップ	<a href="#">313 ページの「ルート (/) ファイルシステムをバックアップする方法」</a>
ミラーファイルシステムのオンラインバックアップの実行	<a href="#">316 ページの「ミラーのオンラインバックアップを実行する方法 (Solaris Volume Manager)」</a>
クラスタ構成のバックアップ	<a href="#">319 ページの「クラスタ構成をバックアップする方法」</a>
ストレージディスクのディスクパーティション分割構成のバックアップ	ストレージディスクのドキュメントを参照

## ▼ バックアップするファイルシステム名を検索する方法

バックアップするファイルシステムの名前を調べるには、この手順を使用します。

- 1 **/etc/vfstab** ファイルの内容を表示します。  
このコマンドを実行するために、スーパーユーザーまたは同等の役割になる必要はありません。

```
more /etc/vfstab
```

- 2 バックアップするファイルシステムの名前の、マウントポイントを表す列を調べます。  
ファイルシステムのバックアップを作成する際にはこの名前を使用します。

```
more /etc/vfstab
```

### 例 12-1 バックアップするファイルシステム名の確認

次の例は、**/etc/vfstab** ファイルに記述されている使用可能なファイルシステムの名前を示しています。

```
more /etc/vfstab
#device device mount FS fsck mount mount
#to mount to fsck point type pass at boot options
#
#/dev/dsk/cl1d0s2 /dev/rdisk/cl1d0s2 /usr ufs 1 yes -
f - /dev/fd fd - no -
/proc - /proc proc - no -
/dev/dsk/cl1t6d0s1 - - swap - no -
/dev/dsk/cl1t6d0s0 /dev/rdisk/cl1t6d0s0 / ufs 1 no -
/dev/dsk/cl1t6d0s3 /dev/rdisk/cl1t6d0s3 /cache ufs 2 yes -
swap - /tmp tmpfs - yes -
```

## ▼ 完全バックアップに必要なテープ数を調べる方法

ファイルシステムのバックアップに必要なテープ数を計算するには、この手順を使用します。

- 1 バックアップするクラスタノード上で、スーパーユーザーまたは同等の役割になります。
- 2 バックアップのサイズをバイト単位で推定します。

```
ufsdump S filesystem
```

S                      バックアップの実行に必要なだと推定されるバイト数を表示します。



*filesystem* バックアップするファイルシステムの名前を指定します。

- 3 推定のサイズをテープの容量で割り、必要なテープの数を確認します。

### 例 12-2 必要なテープの数の特定

次の例では、ファイルシステムのサイズが 905,881,620 バイトなので、4G バイトのテープに容易に収まります (905,881,620 ÷ 4,000,000,000)。

```
ufsdump S /global/phys-schost-1
905881620
```

## ▼ ルート (/) ファイルシステムをバックアップする方法

クラスタノードのルート (/) ファイルシステムをバックアップするには、この手順を使用します。バックアップ手順を実行する前に、クラスタが問題なく動作していることを確認してください。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 バックアップするクラスタノードで、スーパーユーザーになるか、RBAC の承認 **solaris.cluster.modify** を提供する役割になります。
- 2 実行中の各データサービスを、バックアップ対象のノードから、クラスタ内の別のノードに切り替えます。

```
clnode evacuate node
```

*node* リソースグループとデバイスグループを切り替えるノードを指定します。

- 3 ノードを停止します。

```
shutdown -g0 -y -i0
```

- 4 ノードを非クラスタモードでリブートします。

- SPARC ベースのシステム上で、次のコマンドを実行します。

```
ok boot -xs
```

- x86 ベースのシステム上で、次のコマンドを実行します。

```
phys-schost# shutdown -g -y -i0
```

Press any key to continue

- a. GRUB メニューで矢印キーを使用して該当する **Oracle Solaris** エントリを選択し、**e** と入力してコマンドを編集します。

次のような GRUB メニューが表示されます。

```
GNU GRUB version 0.95 (631K lower / 2095488K upper memory)
+-----+
| Solaris 10 /sol_10_x86 |
| Solaris failsafe |
| |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.
```

GRUB ベースのブートの詳細については、『[Oracle Solaris の管理: 基本管理](#)』の「[GRUB を使用して x86 システムをブートする \(タスクマップ\)](#)」を参照してください。

- b. ブートパラメータの画面で、矢印キーを使用してカーネルエントリを選択し、**e** を入力してエントリを編集します。

次のような GRUB ブートパラメータの画面が表示されます。

```
GNU GRUB version 0.95 (615K lower / 2095552K upper memory)
+-----+
| root (hd0,0,a) |
| kernel /platform/i86pc/multiboot |
| module /platform/i86pc/boot_archive |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.
```

- c. コマンドに **-x** を追加して、システムが非クラスタモードでブートするように指定します。

```
[Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time exits.]
```

```
grub edit> kernel /platform/i86pc/multiboot -x
```

- d. **Enter** キーを押して変更を受け入れ、ブートパラメータの画面に戻ります。  
画面には編集されたコマンドが表示されます。

```
GNU GRUB version 0.95 (615K lower / 2095552K upper memory)
+-----+
| root (hd0,0,a) |
| kernel /platform/i86pc/multiboot -x |
+-----+
```

```
| module /platform/i86pc/boot_archive |
+-----+
Use the ^ and v keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.-
```

- e. **b**と入力して、ノードを非クラスタモードでブートします。

---

注-カーネルブートパラメータコマンドへのこの変更は、システムをブートする  
と無効になります。次にノードをリブートする際には、ノードはクラスタモード  
でブートします。そうではなく、非クラスタモードでブートするには、これらの  
手順を再度実行して、カーネルブートパラメータコマンドに **-x** オプションを追加  
します。

---

- 5 **UFS** スナップショットを作成することで、ルート (**/**) ファイルシステムをバックアップします。

- a. ファイルシステムに、バックキングストアファイル用の十分なディスク容量があることを確認します。

```
df -k
```

- b. 名前と場所が同じバックキングストアファイルがすでに存在していないことを確認します。

```
ls /backing-store-file
```

- c. **UFS** スナップショットを作成します。

```
fssnap -F ufs -o bs=/backing-store-file /file-system
```

- d. スナップショットが作成されたことを確認します。

```
/usr/lib/fs/ufs/fssnap -i /file-system
```

- 6 ファイルシステムのスナップショットをバックアップします。

```
ufsdump 0ucf /dev/rmt/0 snapshot-name
```

例:

```
ufsdump 0ucf /dev/rmt/0 /dev/rfssnap/1
```

- 7 スナップショットがバックアップされたことを確認します。

```
ufsrestore ta /dev/rmt/0
```

- 8 ノードをクラスタモードでリブートします。

```
init 6
```

### 例 12-3 ルート (/) ファイルシステムのバックアップ

次の例では、ルート (/) ファイルシステムのスナップショットは /usr ディレクトリの /scratch/usr.back.file に保存されます。

```
fssnap -F ufs -o bs=/scratch/usr.back.file /usr
/dev/fssnap/1
```

## ▼ ミラーのオンラインバックアップを実行する方法 (Solaris Volume Manager)

Solaris Volume Manager のミラー化ボリュームは、マウント解除したりミラー全体をオフラインにしたりしなくても、バックアップできます。サブミラーの1つを一時的にオフラインにする必要があるのですが、ミラー化の状態ではなくなります。バックアップ完了後ただちにオンラインに戻し、再度同期をとることができます。システムを停止したり、データへのユーザーアクセスを拒否する必要はありません。ミラーを使用してオンラインバックアップを実行すると、アクティブなファイルシステムの「スナップショット」であるバックアップが作成されます。

lockfs コマンドを実行する直前にプログラムがボリュームにデータを書き込むと、問題が生じることがあります。この問題を防ぐには、このノードで実行中のすべてのサービスを一時的に停止します。また、バックアップ手順を実行する前に、クラスタが正常に動作していることを確認してください。

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 バックアップするクラスタノード上で、スーパーユーザーまたは同等の役割になります。
- 2 **metaset(1M)** コマンドを使用して、バックアップするボリュームの所有権を持っているノードを確認します。

```
metaset -s setname
-s setname ディスクセット名を指定します。
```

- 3 **lockfs(1M)** コマンドに **-w** オプションを付けて使用し、ファイルシステムへの書き込みをロックします。

```
lockfs -w mountpoint
```

---

注- ファイルシステムをロックする必要があるのは、UFS ファイルシステムがミラー上にある場合のみです。たとえば、データベース管理ソフトウェアまたはほかの特定アプリケーションのために、Solaris Volume Manager ボリュームが raw デバイスとして設定されている場合は、`lockfs` コマンドを使用する必要はありません。ただし、ベンダーに依存する適切なユーティリティを実行し、任意のバッファをフラッシュしてアクセスをロックすることもできます。

---

- 4 **metastat(1M)** コマンドを使用し、サブミラーの名前を調べます。

```
metastat -s setname -p
```

-p                      md.tab ファイルと同様の形式でステータスを表示します。

- 5 **metadetach(1M)** コマンドを使用し、ミラーから 1 つのサブミラーをオフラインにします。

```
metadetach -s setname mirror submirror
```

---

注- 読み取り操作は引き続きそのほかのサブミラーから行われます。読み取り操作は引き続きそのほかのサブミラーから実行できますが、オフラインのサブミラーは、ミラーに最初書き込んだ直後から同期がとれなくなります。この不一致は、オフラインのサブミラーをオンラインに戻したときに修正されます。`fsck` を実行する必要はありません。

---

- 6 **-u** オプションを指定して **lockfs** コマンドを使用し、ファイルシステムのロックを解除して書き込みを続行できるようにします。

```
lockfs -u mountpoint
```

- 7 ファイルシステムを検査します。

```
fsck /dev/md/diskset/rdsk/submirror
```

- 8 オフラインのサブミラーをテープなどのメディアにバックアップします。

**ufsdump(1M)** コマンド、または通常使用しているバックアップユーティリティを使用します。

```
ufsdump 0ucf dump-device submirror
```

---

注- ブロックデバイス (/dsk) 名ではなく、サブミラーの raw デバイス (/rdsk) 名を使用してください。

---

- 9 **metattach(1M)** コマンドを使用し、メタデバイスまたはボリュームをオンラインに戻します。

```
metattach -s setname mirror submirror
```

メタデバイスまたはボリュームをオンラインに戻すと、自動的にミラーとの再同期が行われます。

- 10** **metastat** コマンドを使用し、サブミラーが再同期されていることを確認します。

```
metastat -s setname mirror
```

#### 例 12-4 ミラーのオンラインバックアップの実行 (Solaris Volume Manager)

次の例では、クラスタノード phys-schost-1 はメタセット schost-1 の所有者なので、バックアップ手順は phys-schost-1 から実行します。ミラー /dev/md/schost-1/dsk/d0 は、サブミラーの d10、d20、および d30 で構成されています。

```
[Determine the owner of the metaset:]
metastat -s schost-1
Set name = schost-1, Set number = 1
Host Owner
phys-schost-1 Yes
...
[Lock the file system from writes:]
lockfs -w /global/schost-1
[List the submirrors:]
metastat -s schost-1 -p
schost-1/d0 -m schost-1/d10 schost-1/d20 schost-1/d30 1
schost-1/d10 1 1 d4s0
schost-1/d20 1 1 d6s0
schost-1/d30 1 1 d8s0
[Take a submirror offline:]
metadetach -s schost-1 d0 d30
[Unlock the file system:]
lockfs -u /
[Check the file system:]
fsck /dev/md/schost-1/rdisk/d30
[Copy the submirror to the backup device:]
ufsdump 0ucf /dev/rmt/0 /dev/md/schost-1/rdisk/d30
DUMP: Writing 63 Kilobyte records
DUMP: Date of this level 0 dump: Tue Apr 25 16:15:51 2000
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /dev/md/schost-1/rdisk/d30 to /dev/rdisk/c1t9d0s0.
...
DUMP: DUMP IS DONE
[Bring the submirror back online:]
metattach -s schost-1 d0 d30
schost-1/d0: submirror schost-1/d30 is attached
[Resynchronize the submirror:]
metastat -s schost-1 d0
schost-1/d0: Mirror
Submirror 0: schost-0/d10
State: Okay
Submirror 1: schost-0/d20
State: Okay
Submirror 2: schost-0/d30
State: Resyncing
```

```
Resync in progress: 42% done
Pass: 1
Read option: roundrobin (default)
...
```

## ▼ クラスタ構成をバックアップする方法

クラスタ構成をアーカイブし、クラスタ構成の簡単な復元を実現するため、定期的にクラスタ構成をバックアップします。Oracle Solaris Cluster には、クラスタ構成を XML (eXtensible Markup Language) ファイルにエクスポートする機能があります。

- 1 クラスタ内の任意のノードにログオンし、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.read` を提供する役割になります。

- 2 クラスタ構成情報をファイルにエクスポートします。

```
/usr/cluster/bin/cluster export -o configfile
```

*configfile* クラスタコマンドのクラスタ構成情報のエクスポート先である XML 構成ファイルの名前。XML 構成ファイルについては、[clconfiguration\(5CL\)](#) を参照してください。

- 3 クラスタ構成情報が正常に XML ファイルにエクスポートされたことを確認します。

```
vi configfile
```

## クラスタファイルの復元

[ufsrestore\(1M\)](#) コマンドは、現在の作業用ディレクトリを基準に、[ufsdump\(1M\)](#) コマンドを使用して作成されたバックアップからファイルをディスクにコピーします。`ufsrestore` を使用すると、レベル 0 のダンプおよびそれに続く増分ダンプから、ファイルシステムの階層全体を再読み込みしたり、任意のダンプテープから 1 つまたは複数の単独ファイルを復元したりできます。スーパーユーザーとして、または同等の役割になって `ufsrestore` を実行すると、元の所有者、最終変更時間、およびモード (アクセス権) の情報とともにファイルが復元されます。

ファイルまたはファイルシステムの復元を開始する前に、次の点を確認してください。

- 必要なテープ
- ファイルシステムを復元する raw デバイス名
- 使用するテープドライブの種類
- テープドライブのデバイス名 (ローカルまたはリモート)

- 障害が発生したディスクのパーティション分割方式。これは、パーティションとファイルシステムを交換用ディスクに正確に複製しなければならないためです。

表 12-2 タスクリスト:クラスタファイルの復元

タスク	手順
Solaris Volume Manager の場合にファイルを対話式に復元	320 ページの「個々のファイルを対話式に復元する方法 (Solaris Volume Manager)」
Solaris Volume Manager の場合にルート (/) ファイルシステムを復元	320 ページの「ルート (/) ファイルシステムを復元する方法 (Solaris Volume Manager)」  323 ページの「Solaris Volume Manager ボリューム上にあったルート (/) ファイルシステムを復元する方法」

▼ 個々のファイルを対話式に復元する方法 (Solaris Volume Manager)

1 つまたは複数の個々のファイルを復元するには、この手順を使用します。復元手順を実行する前に、クラスタが正常に動作していることを確認してください。

- 1 復元するクラスタノードで、スーパーユーザーになるか、RBAC の承認 `solaris.cluster.admin` を提供する役割になります。
- 2 復元するファイルを使用中のデータサービスをすべて停止します。  
`# clresourcegroup offline resource-group`
- 3 ファイルを復元します。  
`# ufsrestore`

▼ ルート (/) ファイルシステムを復元する方法 (Solaris Volume Manager)

障害が発生したルートディスクの交換後などに、ルート (/) ファイルシステムを新しいディスクに復元するには、この手順を使用します。復元中のノードはブートしないでおさないでください。復元手順を実行する前に、クラスタが正常に動作していることを確認してください。



---

注- 新しいディスクは、障害が発生したディスクと同じ形式でパーティション分割する必要があります。このため、この手順を始める前に、パーティションの分割方式を確認し、ファイルシステムを適切に再作成しておいてください。

---

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 復元するノードの添付先であるディスクセットへのアクセス権があるクラスタノード上で、スーパーユーザーになるか、**solaris.cluster.modify RBAC** 権限を提供する役割になります。  
復元する以外のノードを使用します。

- 2 すべてのメタセットから、復元するノードのホスト名を削除します。  
このコマンドは、削除するノード以外のメタセットのノードから実行します。復元を行なっているノードはオフラインであるため、システムは「RPC: Rpcbind failure - RPC: Timed out」というエラーを表示します。このエラーを無視し、次のステップを続けます。

```
metaset -s setname -f -d -h nodelist
```

```
-s setname ディスクセット名を指定します。
-f ディスクセットから最後のホストを削除します。
-d ディスクセットから削除します。
-h nodelist ディスクセットから削除するノードの名前を指定します。
```

- 3 ルート (/) および /usr ファイルシステムを復元します。  
ルートおよび /usr ファイルシステムを復元するには、『[Oracle Solaris の管理: デバイスとファイルシステム](#)』の「[UFS ファイルとファイルシステムの復元](#)」の手順に従います。Oracle Solaris OS の場合のやり方の手順は省略してシステムをリブートします。

---

注 - /global/.devices/ node@nodeid ファイルシステムが作成されていることを確認します。

---

- 4 ノードをマルチユーザーモードでリブートします。

```
reboot
```

- 5 デバイスIDを交換します。

```
cldevice repair rootdisk
```

- 6 **metadb(1M)** コマンドを使用して、状態データベースの複製を再作成します。

```
metadb -c copies -af raw-disk-device
```

-c copies                      作成する複製の数を指定します。

-f raw-disk-device          複製の作成先の raw ディスクデバイス名を指定します。

-a                              複製を追加します。

- 7 復元するノード以外のクラスタノードから、復元するノードをすべてのディスクセットに追加します。

```
phys-schost-2# metaset -s setname -a -h nodelist
```

-a                              ホストを作成してディスクセットに追加します。

ノードがクラスタモードでリブートします。これでクラスタを使用できるようになります。

## 例 12-5 ルート (/) ファイルシステムの復元 (Solaris Volume Manager)

次の例では、テープデバイス /dev/rmt/0 からノード phys-schost-1 に復元されたルート (/) ファイルシステムを示しています。metaset コマンドは、クラスタの別のノード phys-schost-2 から実行し、ノード phys-schost-1 を削除し、後でディスクセット schost-1 に追加します。そのコマンドはすべて phys-schost-1 から実行します。新しいブートブロックが /dev/rdsk/c0t0d0s0 に作成され、3 つの状態データベースの複製が /dev/rdsk/c0t0d0s4 に再作成されます。

```
[Become superuser or assume a role that provides solaris.cluster.modify RBAC authorization on a cluster node
other than the node to be restored.]
[Remove the node from the metaset:]
phys-schost-2# metaset -s schost-1 -f -d -h phys-schost-1
[Replace the failed disk and boot the node:]
Restore the root (/) and /usr file system using the procedure in the Solaris system
administration documentation
[Reboot:]
reboot
[Replace the disk ID:]
cldevice repair /dev/dsk/c0t0d0
[Re-create state database replicas:]
metadb -c 3 -af /dev/rdsk/c0t0d0s4
[Add the node back to the metaset:]
phys-schost-2# metaset -s schost-1 -a -h phys-schost-1
```

## ▼ Solaris Volume Manager ボリューム上にあったルート (/) ファイルシステムを復元する方法

バックアップが実行されたときに Solaris Volume Manager ボリューム上にあったルート (/) ファイルシステムを復元するには、この手順を使用します。この手順は、ルートディスクが破損し、新しいディスクに交換する場合などの状況で実行します。復元中のノードはブートしなさいでください。復元手順を実行する前に、クラスタが正常に動作していることを確認してください。

---

注-新しいディスクは、障害が発生したディスクと同じ形式でパーティション分割する必要があります。このため、この手順を始める前に、パーティションの分割方式を確認し、ファイルシステムを適切に再作成しておいてください。

---

phys-schost# プロンプトは、グローバルクラスタのプロンプトを表します。この手順は、グローバルクラスタ上で実行します。

この手順では、長形式の Oracle Solaris Cluster コマンドを使用して説明します。多くのコマンドには短縮形もあります。コマンド名の形式の長短を除き、コマンドは同一です。

- 1 ディスクセットにアクセスできる、復元するノード以外のクラスタノード上で、スーパーユーザーになるか、RBAC 権限 **solaris.cluster.modify** を提供する役割になります。  
復元する以外のノードを使用します。
- 2 そのノードが接続されているすべてのディスクセットから、復元しようとしているノードのホスト名を削除します。次のコマンドをディスクセットごとに1回実行します。

```
metaset -s setname -d -h hostname
```

-s setname	メタセット名を指定します。
-f	ディスクセットから最後のホストを削除します。
-d	メタセットから削除します。
-h nodelist	メタセットから削除するノードの名前を指定します。
-h hostname	ホストの名前を指定します。
-m mediator_host_list	ディスクセットに対して追加または削除するメディアータホストの名前を指定しま

す。

- 3 ノードがデュアルストリングメディアータホストの場合は、メディアータを削除します。次のコマンドを、ノードの接続先になっているディスクセットごとに1回実行します。

```
metaset -ssetname-d -m hostname
```

- 4 ルート(/)ファイルシステムが復元されるノードで、障害が発生したディスクを交換します。

ディスクの交換手順については、サーバーに付属するドキュメントを参照してください。

- 5 復元するノードをブートします。修復されたノードは、**CD-ROM**からシングルユーザーモードでブートされるので、このノードでは**Solaris Volume Manager**は実行されていません。

- Oracle Solaris OS の CD を使用する予定の場合は、次のことに注意してください。

- SPARC: 次のように入力します。

```
ok boot cdrom -s
```

- x86: CD をシステムの CD ドライブに挿入し、システムをシャットダウンしてからオフおよびオンにすることでシステムをブートします。「Current Boot Parameters」画面で、b または i と入力します。

```
<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci8086,2545@3/pci8086,1460@1d/pci8086,341a@
7,1/sd@0,0:a
Boot args:

Type b [file-name] [boot-flags] <ENTER> to boot with options
or i <ENTER> to enter boot interpreter
or <ENTER> to boot with defaults
```

```
<<< timeout in 5 seconds >>>
Select (b)oot or (i)nterpreter: b -s
```

- Solaris JumpStart サーバーを使用する場合は、次のことに注意してください。

- SPARC: 次のように入力します。

```
ok boot net -s
```

- x86: CD をシステムの CD ドライブに挿入し、システムをシャットダウンしてからオフおよびオンにすることでシステムをブートします。「Current Boot Parameters」画面で、b または i と入力します。

```
<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci8086,2545@3/pci8086,1460@1d/pci8086,341a@
7,1/sd@0,0:a
Boot args:

Type b [file-name] [boot-flags] <ENTER> to boot with options
```

```

or i <ENTER> to enter boot interpreter
or <ENTER> to boot with defaults

<<< timeout in 5 seconds >>>
Select (b)oot or (i)nterpreter: b -s

```

- 6 **format** コマンドを使用して、ルートディスク上にすべてのパーティションとスワップ空間を作成します。  
障害が発生したディスクでの元のパーティション分割スキームを再作成します。
- 7 **newfs** コマンドを使用して、ルート(/)ファイルシステムと、必要に応じてほかのファイルシステムを作成します。  
障害が発生したディスクでの元のファイルシステムを再作成します。

---

注 - /global/.devices/ node@nodeid ファイルシステムが作成されていることを確認します。

---

- 8 ルート(/)ファイルシステムを一時マウントポイントにマウントします。  
**# mount device temp-mountpoint**
- 9 次のコマンドを使用して、ルート(/)ファイルシステムを復元します。  
**# cd temp-mountpoint**  
**# ufsrestore rvf dump-device**  
**# rm restoresymtable**
- 10 新しいブートブロックを新しいディスクにインストールします。  
**# /usr/sbin/installboot /usr/platform/'uname -i'/lib/fs/ufs/bootblk raw-disk-device**
- 11 **/temp-mountpoint/etc/system** ファイルで、MDD ルート情報の行を削除します。  
\* Begin MDD root info (do not edit)  
forceload: misc/md\_trans  
forceload: misc/md\_raid  
forceload: misc/md\_mirror  
forceload: misc/md\_hotspares  
forceload: misc/md\_stripe  
forceload: drv/pcipsy  
forceload: drv/glm  
forceload: drv/sd  
rootdev:/pseudo/md@0:0,10,blk  
\* End MDD root info (do not edit)
- 12 **/temp-mountpoint/etc/vfstab** ファイルを編集し、ルートエントリを **Solaris Volume Manager** ボリュームから、メタデバイスまたはボリュームの一部であるルートディスク上の各ファイルシステムの対応する通常のスライスに変更します。  
Example:  
Change from-

```
/dev/md/dsk/d10 /dev/md/rdsk/d10 / ufs 1 no -
```

```
Change to-
/dev/dsk/c0t0d0s0 /dev/rdsk/c0t0d0s0 / ufs 1 no -
```

- 13 一時ファイルシステムをアンマウントし、**raw** ディスクデバイスを確認します。

```
cd /
umount temp-mountpoint
fsck raw-disk-device
```

- 14 ノードをマルチユーザーモードでリブートします。

```
reboot
```

- 15 デバイス ID を交換します。

```
cldevice repair rootdisk
```

- 16 **metadb** コマンドを使用し、状態データベースの複製を再作成します。

```
metadb -c copies -af raw-disk-device
```

-c copies 作成する複製の数を指定します。

-af raw-disk-device 指定した raw ディスクデバイス上に、初期状態のデータベース複製を作成します。

- 17 復元するノード以外のクラスタノードから、復元するノードをすべてのディスクセットに追加します。

```
phys-schost-2# metaset -s setname -a -h nodelist
```

-a メタセットを追加 (作成) します。

ドキュメントに従って、ルート (/) のボリュームやミラーを設定します。

ノードがクラスタモードでリブートします。

- 18 ノードがデュアルストリングメディアータホストであった場合は、メディアータを再度追加します。

```
phys-schost-2# metaset -s setname -a -m hostname
```

## 例 12-6 Solaris Volume Manager ボリューム上にあったルート (/) ファイルシステムの復元

次の例では、テープデバイス /dev/rmt/0 からノード phys-schost-1 に復元されたルート (/) ファイルシステムを示しています。metaset コマンドは、クラスタの別のノード phys-schost-2 から実行し、ノード phys-schost-1 を削除し、あとでメタセット schost-1 に追加します。そのコマンドはすべて phys-schost-1 から実行します。新しいブートブロックが /dev/rdsk/c0t0d0s0 に作成され、3 つの状態データベースの複製が /dev/rdsk/c0t0d0s4 に再作成されます。

```
[Become superuser or assume a role that provides solaris.cluster.modify RBAC
 authorization on a cluster node with access to the metaset, other than the node to be restored.]
[Remove the node from the metaset:]
phys-schost-2# metaset -s schost-1 -d -h phys-schost-1
[Replace the failed disk and boot the node:]
```

Oracle Solaris OS の CD からノードをブートします。

- SPARC: 次のように入力します。

```
ok boot cdrom -s
```

- x86: CD をシステムの CD ドライブに挿入し、システムをシャットダウンしてからオフおよびオンにすることでシステムをブートします。「Current Boot Parameters」画面で、**b** または **i** と入力します。

```
<<< Current Boot Parameters >>>
Boot path: /pci@0,0/pci8086,2545@3/pci8086,1460@1d/pci8086,341a@7,1/
sd@0,0:a
Boot args:

Type b [file-name] [boot-flags] <ENTER> to boot with options
or i <ENTER> to enter boot interpreter
or <ENTER> to boot with defaults

<<< timeout in 5 seconds >>>
Select (b)oot or (i)nterpreter: b -s
```

```
[Use format and newfs to recreate partitions and file systems
.]
[Mount the root file system on a temporary mount point:]
mount /dev/dsk/c0t0d0s0 /a
[Restore the root file system:]
cd /a
ufsrestore rvf /dev/rmt/0
rm restoresymtable
[Install a new boot block:]
/usr/sbin/installboot /usr/platform/'uname \
-i' /lib/fs/ufs/bootblk /dev/rdisk/c0t0d0s0

[Remove the lines in / temp-mountpoint/etc/system file for MDD root information:
]
* Begin MDD root info (do not edit)
forceload: misc/md_trans
forceload: misc/md_raid
forceload: misc/md_mirror
forceload: misc/md_hotspares
forceload: misc/md_stripe
forceload: drv/pcipsy
forceload: drv/glm
forceload: drv/sd
rootdev:/pseudo/md@0:0,10,blk
* End MDD root info (do not edit)
[Edit the /temp-mountpoint/etc/vfstab file]
Example:
Change from-
/dev/md/dsk/d10 /dev/md/rdsk/d10 / ufs 1 no -
```

```
Change to-
/dev/dsk/c0t0d0s0 /dev/rdsk/c0t0d0s0 /usr ufs 1 no -
[Unmount the temporary file system and check the raw disk device:]
cd /
umount /a
fsck /dev/rdsk/c0t0d0s0
[Reboot:]
reboot
[Replace the disk ID:]
cldevice repair /dev/rdsk/c0t0d0
[Re-create state database replicas:]
metadb -c 3 -af /dev/rdsk/c0t0d0s4
[Add the node back to the metaset:]
phys-schost-2# metaset -s schost-1 -a -h phys-schost-1
```



# グラフィカルユーザーインターフェースによる Oracle Solaris Cluster の管理

---

この章では、Oracle Solaris Cluster Manager グラフィカルユーザーインターフェース (GUI) ツールについて説明します。これを使用すると、クラスタをさまざまな面から管理できます。また、Oracle Solaris Cluster Manager を構成および起動する手順も含まれています。Oracle Solaris Cluster Manager GUI に含まれるオンラインヘルプでは、さまざまな Oracle Solaris Cluster 管理タスクの手順を説明しています。

この章の内容は、次のとおりです。

- 329 ページの「Oracle Solaris Cluster Manager の概要」
- 330 ページの「Oracle Solaris Cluster Manager の構成」
- 333 ページの「Oracle Solaris Cluster Manager ソフトウェアの起動」

## Oracle Solaris Cluster Manager の概要

Oracle Solaris Cluster Manager は、クラスタ情報のグラフィカルな表示、クラスタコンポーネントのステータスの確認、および構成変更の監視が可能な GUI です。Oracle Solaris Cluster Manager では、以下の Oracle Solaris Cluster コンポーネントを対象としたさまざまな管理タスクも行えます。

- アダプタ
- ケーブル
- データサービス
- グローバルデバイス
- 相互接続
- 接続点
- ノードの負荷制限
- NAS デバイス
- ノード
- 定足数デバイス
- リソースグループ
- リソース

Oracle Solaris Cluster Manager をインストールおよび使用する方法については、次の文書を参照してください。

- **Oracle Solaris Cluster Manager** のインストール: 『[Oracle Solaris Cluster ソフトウェアのインストール](#)』を参照してください。
- **Oracle Solaris Cluster Manager** の起動: [333 ページの「Oracle Solaris Cluster Manager ソフトウェアの起動」](#)を参照してください。
- ポート番号、サーバーアドレス、セキュリティ証明書、およびユーザーの構成: [330 ページの「Oracle Solaris Cluster Manager の構成」](#)を参照してください。
- **Oracle Solaris Cluster Manager** によるクラスタのインストールと管理: Oracle Solaris Cluster Manager に付属のオンラインヘルプを参照してください。
- **Oracle Solaris Cluster Manager** セキュリティー鍵の再生成: [332 ページの「共通エージェントコンテナのセキュリティ鍵を再生成する方法」](#)を参照してください。

---

注 - ただし、Oracle Solaris Cluster Manager は現在、Oracle Solaris Cluster のすべての管理タスクを実行できるわけではありません。一部の作業には、コマンド行インタフェースを使用する必要があります。

---

## Oracle Solaris Cluster Manager の構成

Oracle Solaris Cluster Manager は、定足数デバイス、IPMP グループ、インターコネクトコンポーネント、グローバルデバイスなどのあらゆる局面のステータスを管理、表示できる GUI です。この GUI は、多くの Oracle Solaris Cluster CLI コマンドの代わりに使用できます。

Oracle Solaris Cluster Manager をクラスタにインストールする手順については、『[Oracle Solaris Cluster ソフトウェアのインストール](#)』を参照してください。GUI を使用してさまざまなタスクを行う方法については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

このセクションでは、初期インストール後、Oracle Solaris Cluster Manager を再構成するための次のような手順について説明します。

- [331 ページの「RBAC の役割の設定」](#)
- [332 ページの「Oracle Solaris Cluster Manager のサーバーアドレスを変更する方法」](#)
- [332 ページの「共通エージェントコンテナのセキュリティ鍵を再生成する方法」](#)

## RBAC の役割の設定

Oracle Solaris Cluster Manager では、RBAC を使用してクラスタを管理する権限を持つユーザーが決定されます。Oracle Solaris Cluster ソフトウェアには、いくつかの RBAC 権利プロファイルが含まれています。これらの権利プロファイルをユーザーまたは役割に割り当てることで、Oracle Solaris Cluster に対するさまざまなレベルのアクセス権をユーザーに与えることができます。Oracle Solaris Cluster ソフトウェアの RBAC を設定および管理する方法については、第 2 章「Oracle Solaris Cluster と RBAC」を参照してください。

### ▼ 共通エージェントコンテナを使用して、サービスまたは管理エージェントのポート番号を変更する

共通エージェントコンテナサービスのデフォルトのポート番号が実行中の別のプロセスと競合する場合、`cacaoadm` コマンドを使用し、クラスタの各ノード上で、競合しているサービスまたは管理エージェントのポート番号を変更できます。

- 1 すべてのクラスタ上で共通エージェントコンテナ管理デモンを停止します。

```
/opt/bin/cacaoadm stop
```

- 2 Sun Java Web コンソールを停止します。

```
/usr/sbin/smcwebserver stop
```

- 3 `get-param` サブコマンドを使用して、共通エージェントコンテナサービスによって現在使用されているポート番号を取得します。

```
/opt/bin/cacaoadm get-param parameterName
```

`cacaoadm` コマンドを使用して、次の共通エージェントコンテナサービスのポート番号を変更できます。次のリストは、共通エージェントコンテナで管理できるサービスとエージェント、および対応するパラメータ名の例を示しています。

JMX コネクタポート	<code>jmxmp-connector-port</code>
SNMP ポート	<code>snmp-adapter-port</code>
SNMP トラップポート	<code>snmp-adapter-trap-port</code>
コマンドストリームポート	<code>commandstream-adapter-port</code>

- 4 ポート番号を変更します。

```
/opt/bin/cacaoadm set-param parameterName=parameterValue
```

- 5 クラスタの各ノードで、手順 4 を繰り返します。

- 6 **Sun Java Web** コンソールを再起動します。

```
/usr/sbin/smcwebserver start
```

- 7 すべてのクラスタノードで 共通エージェントコンテナ 管理デーモンを再起動します。

```
/opt/bin/cacaoadm start
```

## ▼ Oracle Solaris Cluster Manager のサーバーアドレスを変更する方法

クラスタノードのホスト名を変更する場合は、Oracle Solaris Cluster Manager が実行されるアドレスを変更する必要があります。デフォルトのセキュリティー証明書は、Oracle Solaris Cluster Manager がインストールされる時点でノードのホスト名に基づいて生成されます。ノードのホスト名をリセットするには、証明書ファイル `keystore` を削除し、Oracle Solaris Cluster Manager を再起動します。Oracle Solaris Cluster Manager は、新しいホスト名を使用して新しい証明書ファイルを自動的に作成します。この手順は、ホスト名を変更したすべてのノード上で行う必要があります。

- 1 `/etc/opt/webconsole` にある証明書ファイル `keystore` を削除します。

```
cd /etc/opt/webconsole
pkgrm keystore
```

- 2 Oracle Solaris Cluster Manager を再起動します。

```
/usr/sbin/smcwebserver restart
```

## ▼ 共通エージェントコンテナのセキュリティー鍵を再生成する方法

Oracle Solaris Cluster Manager は、強力な暗号化技術を使用して、Oracle Solaris Cluster Manager Web サーバーと各クラスタノード間の安全な通信を確保しています。

Oracle Solaris Cluster Manager が使用する鍵は、各ノードの `/etc/opt/SUNWcacao/security` ディレクトリに格納されています。これらの鍵は、すべてのクラスタノードで同一でなければなりません。

通常の動作では、これらのキーはデフォルトの構成のままとなります。クラスタノードのホスト名を変更する場合は、共通エージェントコンテナのセキュリティー鍵を再生成する必要があります。また、鍵が攻撃の対象となる恐れがある場合 (マシンのルート侵入など) にも鍵の再生成が必要となります。セキュリティー鍵を再生成するには、次の手順を実行します。

- 1 すべてのクラスタ上で 共通エージェントコンテナ 管理デーモンを停止します。  
`# /opt/bin/cacaoadm stop`
- 2 クラスタの1つのノード上で、セキュリティー鍵を再生成します。  
`phys-schost-1# /opt/bin/cacaoadm create-keys --force`
- 3 セキュリティー鍵を再生成したノード上で 共通エージェントコンテナ 管理デーモンを再起動します。  
`phys-schost-1# /opt/bin/cacaoadm start`
- 4 `/etc/cacao/instances/default` ディレクトリの `tar` ファイルを作成します。  
`phys-schost-1# cd /etc/cacao/instances/default`  
`phys-schost-1# tar cf /tmp/SECURITY.tar security`
- 5 `/tmp/Security.tar` ファイルを各クラスタノードにコピーします。
- 6 `/tmp/SECURITY.tar` ファイルをコピーした各ノード上で、セキュリティーファイルを解凍します。  
`/etc/opt/SUNWcacao/` ディレクトリに既にセキュリティーファイルがある場合は、すべて上書きされます。  
`phys-schost-2# cd /etc/cacao/instances/default`  
`phys-schost-2# tar xf /tmp/SECURITY.tar`
- 7 クラスタの各ノードから `/tmp/SECURITY.tar` ファイルを削除します。  
 セキュリティーのリスクを避けるために `tar` ファイルの各コピーを削除する必要があります。  
`phys-schost-1# rm /tmp/SECURITY.tar`  
`phys-schost-2# rm /tmp/SECURITY.tar`
- 8 すべてのノード上で 共通エージェントコンテナ 管理デーモンを再起動します。  
`phys-schost-1# /opt/bin/cacaoadm start`
- 9 **Oracle Solaris Cluster Manager** を再起動します。  
`# /usr/sbin/smcwebserver restart`

## Oracle Solaris Cluster Manager ソフトウェアの起動

Oracle Solaris Cluster Manager グラフィカルユーザーインターフェース (Graphical User Interface、GUI) は、Oracle Solaris Cluster ソフトウェアをさまざまな面から簡単に管理する方法を提供します。詳細については、Oracle Solaris Cluster Manager のオンラインヘルプを参照してください。

クラスタをブートすると、Sun Java Web コンソールと共通エージェントコンテナの両方が自動的に起動します。Sun Java Web コンソールと共通エージェントコンテナが実行されていることを確認する必要がある場合は、この手順のすぐあとの「トラブルシューティング」セクションを参照してください。

## ▼ Oracle Solaris Cluster Manager を起動する方法

この手順では、クラスタ上で Oracle Solaris Cluster Manager を起動する方法を示します。

- 1 **Oracle Solaris Cluster Manager** にアクセスするときに、クラスタノードの **root** のユーザー名とパスワードを使用するか、異なるユーザー名とパスワードを設定するかを決定します。
  - クラスタノードのルートのユーザー名を使用して Oracle Solaris Cluster Manager にアクセスする場合は、[手順 5](#)に進みます。
  - 別のユーザー名とパスワードを設定する場合は、[手順 3](#)に進んで Oracle Solaris Cluster Manager ユーザーアカウントを設定します。
- 2 クラスタノード上にインストールするクラスタノード上でスーパーユーザーになります。
- 3 **Oracle Solaris Cluster Manager** 経由でクラスタにアクセスするためのユーザーアカウントを作成します。

[useradd\(1M\)](#) コマンドを使用して、ユーザーアカウントをシステムに追加します。root システムアカウントを使用しない場合、Oracle Solaris Cluster Manager にアクセスするには、少なくとも1つのユーザーアカウントを設定する必要があります。Oracle Solaris Cluster Manager のユーザーアカウントは、Oracle Solaris Cluster Manager だけで使用されます。これらのアカウントは、Oracle Solaris OS システムのユーザーアカウントとの関連はありません。RBAC の役割を作成し、それをユーザーアカウントに割り当てる方法については、[55 ページの「Oracle Solaris Cluster 管理権利プロファイルによる RBAC 役割の作成と割り当て」](#)を参照してください。

---

注- ノードにユーザーアカウントが設定されていない場合、そのユーザーはそのノードからは Oracle Solaris Cluster Manager 経由でクラスタにアクセスできません。また、アクセス権を持っている別のクラスタノードからも、そのノードを管理することはできません。

---

- 4 (省略可能) 追加するユーザーアカウントごとに [手順 3](#)を繰り返します。
- 5 管理コンソール、またはクラスタの外部に存在する他のマシンから、ブラウザを起動します。

- 6 ブラウザのディスクとメモリーキャッシュのサイズが、0より大きな値に設定されていることを確認します。
- 7 ブラウザで **Java** および **Javascript** が有効になっていることを確認します。
- 8 ブラウザから、クラスタ内の任意のノード上にある **Oracle Solaris Cluster Manager** のポートに接続します。  
デフォルトのポート番号は 6789 です。  
`https://node:6789/`
- 9 **Web** ブラウザにより提示されたすべての証明書を受け入れます。  
Java Web Console ログインページが表示されます。
- 10 **Oracle Solaris Cluster Manager** にアクセスするユーザーのユーザー名とパスワードを入力します。
- 11 「**Log In**」 ボタンをクリックします。  
Java Web Console のアプリケーション起動ページが表示されます。
- 12 **Systems** カテゴリの下に **Oracle Solaris Cluster Manager** リンクをクリックします。
- 13 **Web** ブラウザにより提示されたすべての追加の証明書を受け入れます。
- 14 **Oracle Solaris Cluster Manager** に接続できない場合は、次のサブステップを実行して、**Solaris** のインストール中に制限されたネットワークプロファイルが選択されたかどうかを判別し、**Java Web** コンソールサービスへの外部アクセスを復元します。  
Oracle Solaris のインストール中に制限されたネットワークプロファイルを選択すると、Sun Java Web コンソールサービスの外部アクセスは制限されます。このネットワークは、Oracle Solaris Cluster Manager の GUI を使用するために必要です。
  - a. **Java Web** コンソールサービスが制限されているかどうかを調べます。  

```
svcprop /system/webconsole:console | grep tcp_listen
```

 tcp\_listen プロパティの値が true でない場合、Web コンソールサービスは制限されます。
  - b. **Java Web Console** サービスへの外部アクセスを復元します。  

```
svccfg
svc:> select system/webconsole
svc:/system/webconsole> setprop options/tcp_listen=true
svc:/system/webconsole> quit
/usr/sbin/smcwebserver restart
```

- c. サービスが使用可能になっていることを確認します。

```
netstat -a | grep 6789
```

サービスが使用可能な場合、コマンド出力により 6789 のエントリが返されます。これは Java Web コンソールへの接続に使用されるポート番号です。

- 注意事項 ■ この手順の実行後に Oracle Solaris Cluster Manager に接続できない場合、`/usr/sbin/smcwebserver status` を入力して、Sun Java Web コンソールが実行されているかどうかを調べます。Sun Java Web コンソールが実行されていない場合、`/usr/sbin/smcwebserver start` を入力して手動で起動します。それでも Oracle Solaris Cluster Manager に接続できない場合は、`usr/sbin/cacaoadm status` を入力して、共通エージェントコンテナが実行されているかどうかを調べます。共通エージェントコンテナが実行されていない場合は、`/usr/sbin/cacaoadm start` を入力して手動で起動します。
- GUI を実行しているノード以外のノードに関する情報を表示しようとしたときにシステムエラーメッセージが表示された場合は、共通エージェントコンテナのネットワークバインドアドレスパラメータが正しい値である `0.0.0.0` に設定されていることを確認してください。

クラスタの各ノードで、次の手順を実行します。

1. `network-bind-address` パラメータの値を表示します。

```
cacaoadm get-param network-bind-address
network-bind-address=0.0.0.0
```

2. パラメータ値が `0.0.0.0` に設定されていない場合は、この値に変更します。

```
cacaoadm stop
cacaoadm set-param network-bind-address=0.0.0.0
cacaoadm start
```



## 例

## Availability Suite ソフトウェアを使用したホストベースのデータ複製の構成

この付録では、Oracle Solaris Cluster Geographic Edition を使用しない、ホストベースの複製の代替方法を説明します。クラスタ間のホストベースの複製の構成と操作を簡素化するには、Oracle Solaris Cluster Geographic Edition をホストベースの複製に使用します。[88 ページの「データ複製についての理解」](#)を参照してください。

この付録の例は、Sun StorageTek Availability Suite 4.0 ソフトウェアを使用してクラスタ間のホストベースのデータ複製を構成する方法を示しています。この例では、NFS アプリケーション用の完全なクラスタ構成を示し、個別のタスクの実行方法に関する詳細情報を提供します。すべてのタスクはグローバルクラスタの投票ノードで行われます。例には、ほかのアプリケーションやクラスタ構成で必要な手順がすべて含まれているわけではありません。

スーパーユーザーの代わりに役割に基づくアクセス制御 (RBAC) を使用してクラスタノードにアクセスする場合は、すべての Oracle Solaris Cluster コマンドの承認を提供する RBAC の役割になることができますようにします。ユーザーがスーパーユーザーでない場合、一連のデータ複製手順には、次の Oracle Solaris Cluster RBAC の承認が必要です。

- `solaris.cluster.modify`
- `solaris.cluster.admin`
- `solaris.cluster.read`

RBAC の役割の使用の詳細については、『[Solaris のシステム管理: セキュリティサービス](#)』を参照してください。各 Oracle Solaris Cluster サブコマンドで必要となる RBAC の承認については、Oracle Solaris Cluster のマニュアルページを参照してください。

## クラスタにおける **Availability Suite** ソフトウェアの理解

このセクションでは、耐障害性について紹介し、Availability Suite ソフトウェアが使用するデータ複製方式について説明します。

耐障害性は、プライマリクラスタに障害が発生した場合に、アプリケーションを代替クラスタに復元する機能です。災害耐性のベースは、データ複製とテイクオーバーです。テイクオーバーは、1つ以上のリソースグループおよびデバイスグループをオンラインにすることにより、アプリケーションサービスをセカンダリクラスタに再配置します。

プライマリクラスタおよびセカンダリクラスタ間でデータが同期して複製されている場合、プライマリサイトで障害が発生してもコミットされたデータは失われません。ただし、データが非同期で複製されていた場合、プライマリサイトで障害が発生する前にセカンダリクラスタに複製されていなかったデータがある可能性があります、それらのデータは失われます。

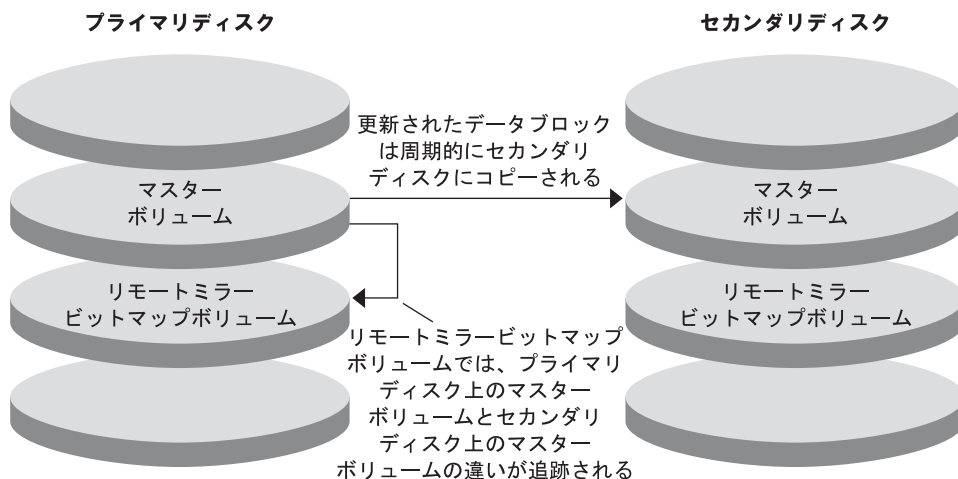
### Availability Suite ソフトウェアが使用するデータ複製方式

このセクションでは、Availability Suite ソフトウェアが使用するリモートミラー複製方式とポイントインタイムスナップショット方式について説明します。このソフトウェアは、`sndradm(1RPC)` コマンドと `iiadm(1II)` コマンドを使用してデータを複製します。これらのコマンドの詳細は、[Availability Suite のドキュメント](#) を参照してください。

### リモートミラー複製

**図 A-1** はリモートミラー複製を示しています。プライマリディスクのマスターボリュームのデータは、TCP/IP 接続を経由してセカンダリディスクのマスターボリュームに複製されます。リモートミラービットマップは、プライマリディスク上のマスターボリュームと、セカンダリディスク上のマスターボリュームの差分を追跡します。

図 A-1 リモートミラー複製



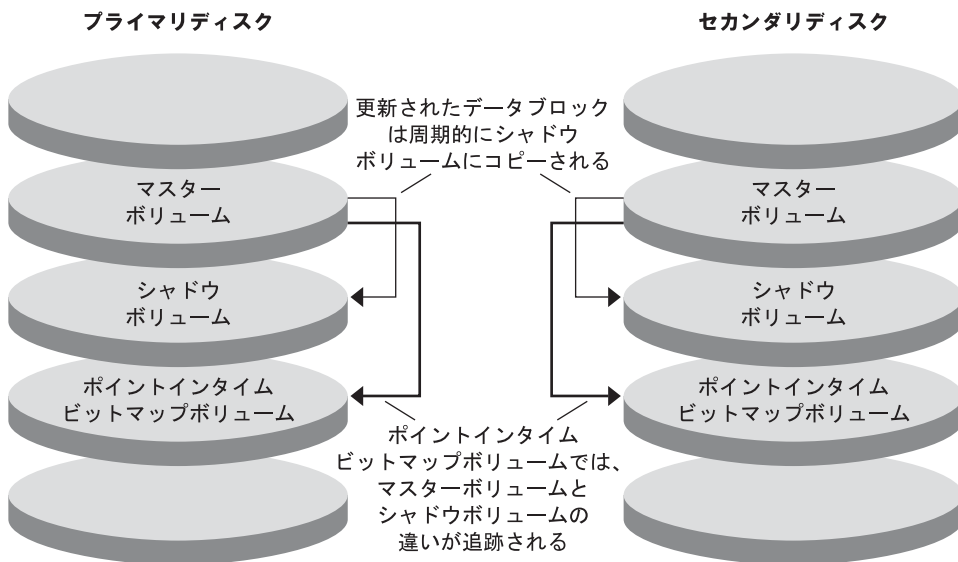
リモートミラー複製は、リアルタイムに同期で実行することも非同期で実行することもできます。各クラスタの各ボリュームセットはそれぞれ、同期複製または非同期複製に構成できます。

- 同期データ複製では、リモートボリュームが更新されるまで、書き込み操作は完了したとは確認されません。
- 非同期データ複製では、リモートボリュームが更新される前に書き込み操作が完了したと確認されます。非同期データ複製は、長い距離や低い帯域幅で大きな柔軟性を発揮します。

## ポイントインタイムスナップショット

図 A-2 は、ポイントインタイムスナップショットを示しています。各ディスクのマスターボリュームのデータは、同じディスクのシャドウボリュームにコピーされます。ポイントインタイムビットマップは、マスターボリュームとシャドウボリューム間の違いを追跡調査します。データがシャドウボリュームにコピーされると、ポイントインタイムビットマップはリセットされます。

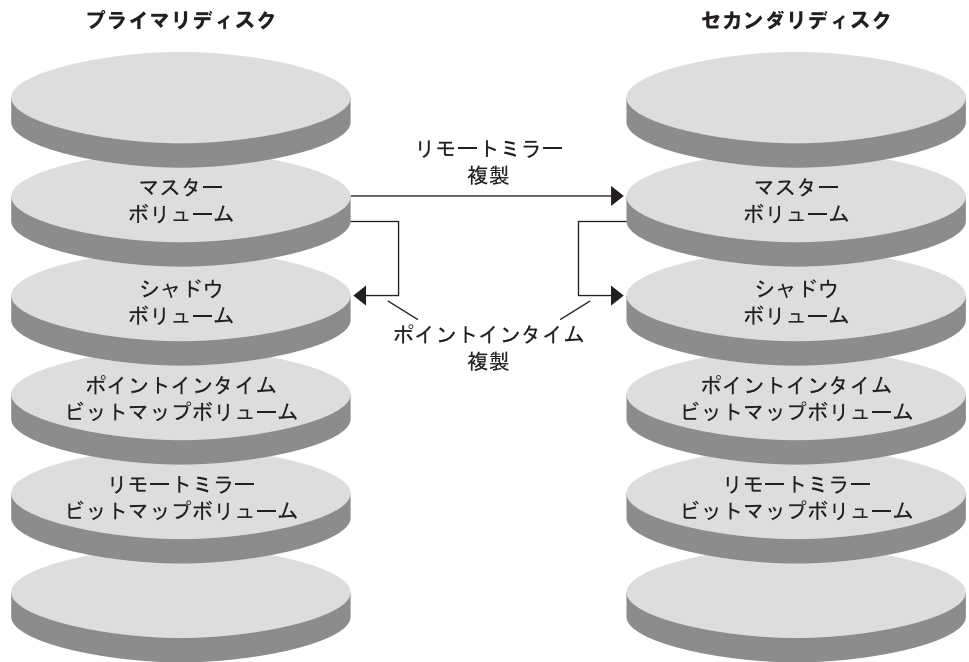
図 A-2 ポイントインタイムスナップショット



### 構成例での複製

図 A-3 に、この構成例でミラー複製とポイントインタイムスナップショットがどのように使用されているかを示します。

図 A-3 構成例での複製



## クラスタ間でホストベースのデータ複製を構成するためのガイドライン

このセクションでは、クラスタ間のデータ複製の構成ガイドラインを提供します。また、複製リソースグループとアプリケーションリソースグループの構成のコツも紹介します。これらのガイドラインは、クラスタのデータ複製を構成する際に使用してください。

このセクションでは、次の項目について説明します。

- [342 ページの「複製リソースグループの構成」](#)
- [343 ページの「アプリケーションリソースグループの構成」](#)
  - [343 ページの「フェイルオーバーアプリケーション向けのリソースグループの構成」](#)
  - [344 ページの「スケーラブルアプリケーション向けのリソースグループの構成」](#)
- [346 ページの「テイクオーバーの管理のガイドライン」](#)

## 複製リソースグループの構成

複製リソースグループは、Availability Suite ソフトウェアの制御下にあるデバイスグループを論理ホスト名リソースと結び付けます。複製リソースグループには、次の特徴があります。

- フェイルオーバーリソースグループである  
フェイルオーバーリソースは、常に単一のノード上で実行されます。フェイルオーバーが発生すると、フェイルオーバーリソースがフェイルオーバーに加わります。
- 論理ホスト名リソースを持つ  
論理ホスト名は、プライマリクラスタによってホストされている必要があります。フェイルオーバー後、論理ホスト名はセカンダリクラスタによってホストされる必要があります。論理ホスト名をクラスタに関連付けるために、ドメインネームシステム (DNS) が使用されます。
- HAStoragePlus リソースを持つ  
HAStoragePlus リソースは、複製リソースグループがスイッチオーバーまたはフェイルオーバーしたときに、デバイスグループをフェイルオーバーします。Oracle Solaris Cluster ソフトウェアはまた、デバイスグループがスイッチオーバーしたときに、複製リソースグループをフェイルオーバーします。このように複製リソースグループとデバイスグループは常に結び付き、同じノードから制御されます。

HAStoragePlus リソース内に次の拡張プロパティを定義する必要があります。

- *GlobalDevicePaths* この拡張プロパティは、ボリュームが属するデバイスグループを定義します。
- *AffinityOn property = True*。この拡張プロパティは、複製リソースグループがスイッチオーバーまたはフェイルオーバーしたときに、デバイスグループをスイッチオーバーまたはフェイルオーバーします。この機能はアフィニティスイッチオーバーと呼ばれます。
- *ZPoolsSearchDir*。この拡張プロパティは、ZFS ファイルシステムを使用するために必要です。

HAStoragePlus についての詳細は、[SUNW.HAStoragePlus\(5\)](#) のマニュアルページを参照してください。

- 結び付いているデバイスグループに `-stor-rg` を付けた名前になる  
たとえば、`devgrp-stor-rg` などです。
- プライマリクラスタとセカンダリクラスタでオンラインになる

## アプリケーションリソースグループの構成

高可用性を実現するためには、アプリケーションはアプリケーションリソースグループのリソースとして管理される必要があります。アプリケーションリソースグループは、フェイルオーバーアプリケーションまたはスケーラブルアプリケーション向けに構成できます。

プライマリクラスタ上に構成したアプリケーションリソースとアプリケーションリソースグループは、セカンダリクラスタ上でも構成される必要があります。また、アプリケーションリソースがアクセスするデータは、セカンダリクラスタに複製する必要があります。

この節では、次のアプリケーションリソースグループを構成するためのガイドラインを紹介します。

- [343 ページの「フェイルオーバーアプリケーション向けのリソースグループの構成」](#)
- [344 ページの「スケーラブルアプリケーション向けのリソースグループの構成」](#)

## フェイルオーバーアプリケーション向けのリソースグループの構成

フェイルオーバーアプリケーションでは、1つのアプリケーションが1度に1ノード上で動作します。ノードで障害が発生すると、アプリケーションは同じクラスタ内の別のノードにフェイルオーバーします。フェイルオーバーアプリケーション向けリソースグループは、以下の特徴を持っていない限りなりません。

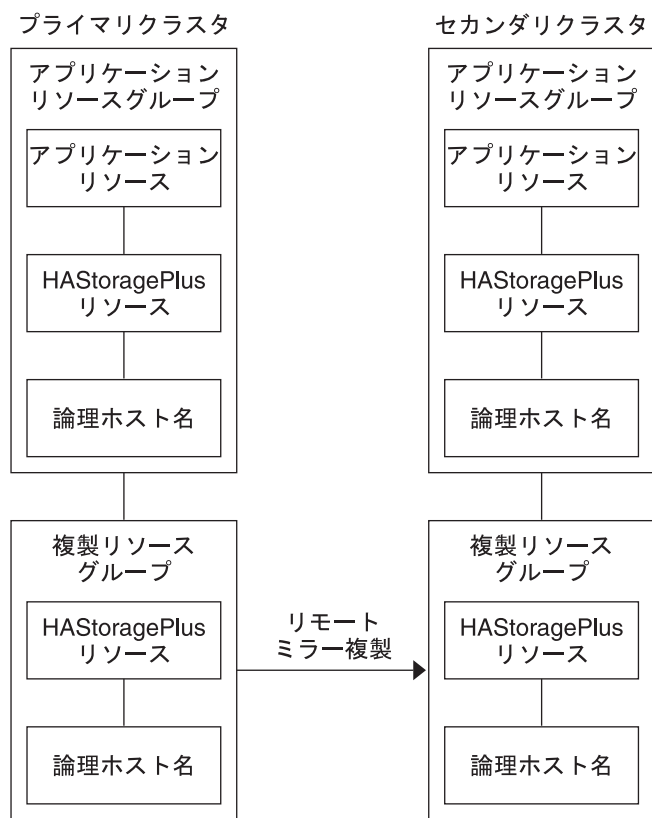
- アプリケーションリソースグループがスイッチオーバーまたはフェイルオーバーされた場合、HAStoragePlus リソースにファイルシステムまたは zpool をフェイルオーバーさせる  
デバイスグループは、複製リソースグループとアプリケーションリソースグループに結び付けられています。したがって、アプリケーションリソースグループがフェイルオーバーすると、デバイスグループと複製リソースグループもフェイルオーバーします。アプリケーションリソースグループ、複製リソースグループおよびデバイスグループは、同じノードによって制御されます。  
ただし、デバイスグループや複製リソースグループがフェイルオーバーしても、アプリケーションリソースグループはフェイルオーバーを行いません。
- アプリケーションデータがグローバルマウントされている場合は、アプリケーションリソースグループに HAStoragePlus リソースを必ず入れなければならないわけではありませんが、入れることをお勧めします。
- アプリケーションデータがローカルマウントされている場合は、アプリケーションリソースグループに HAStoragePlus リソースを必ず入れなければならないません。

HAStoragePlus についての詳細は、[SUNW.HAStoragePlus\(5\)](#) のマニュアルページを参照してください。

- プライマリクラスタでオンライン、セカンダリクラスタでオフラインとなる  
セカンダリクラスタがプライマリクラスタをテイクオーバーした場合は、セカンダリクラスタ上のアプリケーションリソースグループをオンラインにします。

図 A-4 に、フェイルオーバーアプリケーションでのアプリケーションリソースグループと複製リソースグループの構成を示します。

図 A-4 フェイルオーバーアプリケーションでのリソースグループの構成



## スケーラブルアプリケーション向けのリソースグループの構成

スケーラブルアプリケーションでは、アプリケーションは複数のノードで実行されて、1つの論理サービスを作成します。スケーラブルアプリケーションを実行しているノードで障害が発生しても、フェイルオーバーは起こりません。アプリケーションは別のノードで引き続き実行されます。



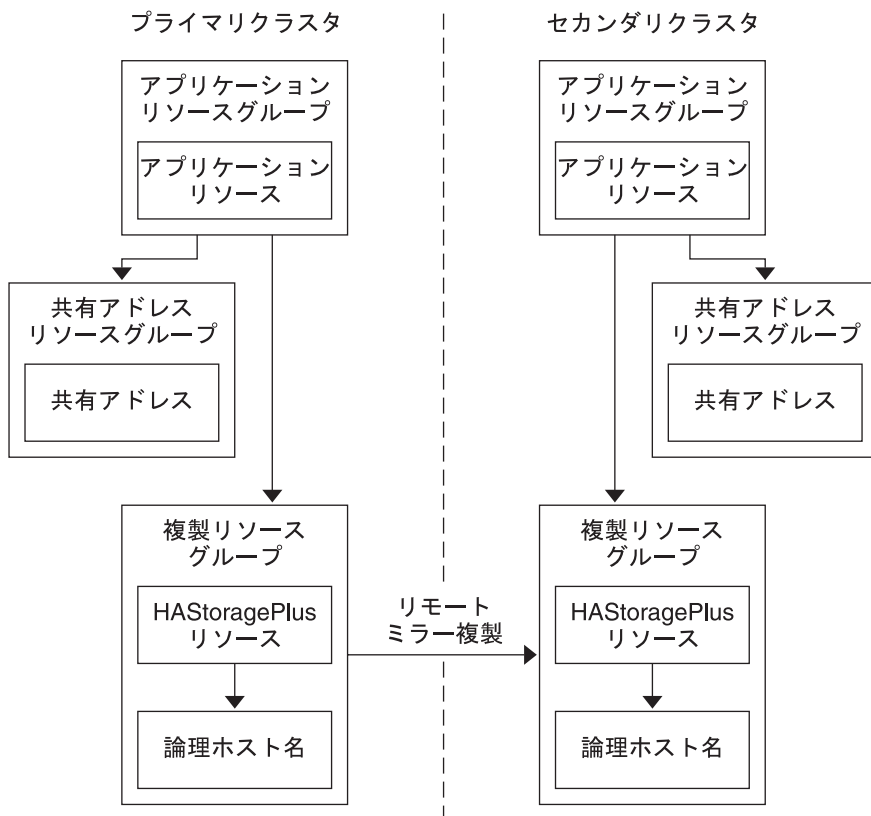
スケーラブルアプリケーションをアプリケーションリソースグループのリソースとして管理している場合は、アプリケーションリソースグループをデバイスグループと結び付ける必要はありません。したがって、アプリケーションリソースグループ向けに HAStoragePlus リソースを作成する必要はありません。

スケーラブルアプリケーション向けリソースグループは、以下の特徴を持っていないければなりません。

- 共有アドレスのリソースグループに依存する  
共有アドレスは、受信データを配信するためにスケーラブルアプリケーションを実行するノードで使用されます。
- プライマリクラスタでオンライン、セカンダリクラスタでオフラインとなる

図 A-5 に、スケーラブルアプリケーションでのリソースグループの構成を示します。

図 A-5 スケーラブルアプリケーションでのリソースグループの構成

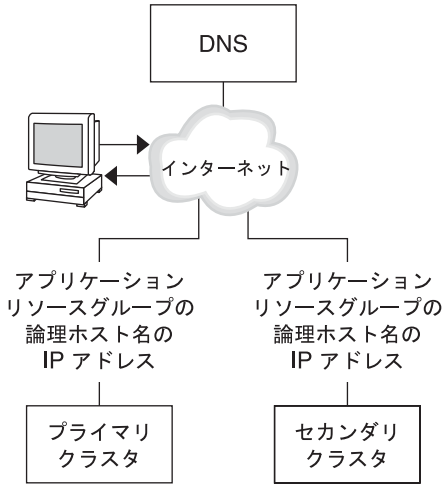


## テイクオーバーの管理のガイドライン

プライマリクラスタに障害が発生した場合、アプリケーションをできるだけ早くセカンダリクラスタにスイッチオーバーする必要があります。セカンダリクラスタがテイクオーバーできるようにするには、DNSを更新する必要があります。

クライアントはDNSを使用して、アプリケーションの論理ホスト名をIPアドレスにマップします。アプリケーションをセカンダリクラスタに移動することによりテイクオーバーを行ったあとに、アプリケーションの論理ホスト名と新しいIPアドレス間のマッピングが反映されるようにDNS情報を更新する必要があります。[図 A-6](#)は、DNSがどのようにクライアントをクラスタにマップするかを示しています。

図 A-6 クライアントからクラスタへの DNS マッピング



DNS を更新するには、`nsupdate` コマンドを使用します。詳細は、[nsupdate\(1M\)](#) のマニュアルページを参照してください。テイクオーバーの管理の例については、[373 ページの「テイクオーバーの管理の例」](#)を参照してください。

プライマリクラスタが修復されたら、オンラインに戻すことができます。元のプライマリクラスタにスイッチバックするには、次の手順を実行します。

1. プライマリクラスタとセカンダリクラスタを同期させ、プライマリボリュームが最新のものであることを確認します。これを行うには、複製データストリームがなくなるように、セカンダリノードのリソースグループを停止します。
2. データ複製の方向を逆にして、元のプライマリクラスタが元のセカンダリクラスタにふたたびデータを複製するようにします
3. プライマリクラスタでリソースグループを起動します。
4. クライアントがプライマリクラスタのアプリケーションにアクセスできるように、DNS を更新します。

## タスクマップ:データ複製の構成例

表 A-1 に、Availability Suite ソフトウェアを使用して NFS アプリケーション向けにどのようにデータ複製を構成したかを示すこの例でのタスクを記載します。

表 A-1 タスクマップ:データ複製の構成例

タスク	手順
1. クラスタを接続およびインストールする	<a href="#">348 ページの「クラスタの接続とインストール」</a>

表 A-1 タスクマップ:データ複製の構成例 (続き)

タスク	手順
2. プライマリクラスタとセカンダリクラスタで、デバイスグループ、NFS アプリケーション用のファイルシステム、およびリソースグループを構成する	<a href="#">350 ページの「デバイスグループとリソースグループの構成例」</a>
3. プライマリクラスタとセカンダリクラスタでデータ複製を有効にする	<a href="#">364 ページの「プライマリクラスタで複製を有効にする方法」</a> <a href="#">366 ページの「セカンダリクラスタで複製を有効にする方法」</a>
4. データ複製を実行する	<a href="#">368 ページの「リモートミラー複製を実行する方法」</a> <a href="#">369 ページの「ポイントインタイムスナップショットを実行する方法」</a>
5. データ複製の構成を確認する	<a href="#">370 ページの「複製が正しく構成されていることを確認する方法」</a>

## クラスタの接続とインストール

図 A-7 に、構成例で使用するクラスタ構成を示します。構成例のセカンダリクラスタにはノードが1つ含まれていますが、これ以外のクラスタ構成も使用できます。

図 A-7 クラスタ構成例

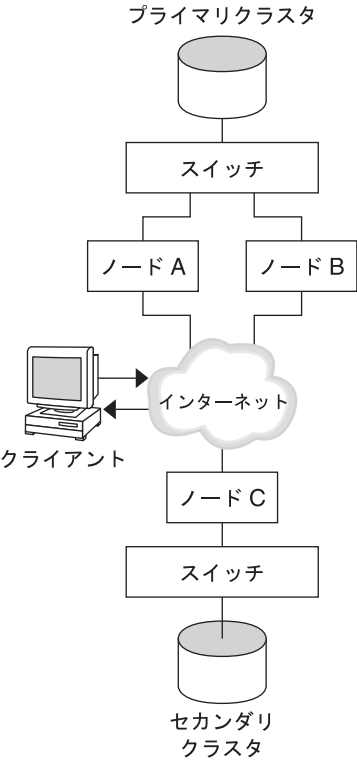


表 A-2 に、構成例で必要となるハードウェアとソフトウェアをまとめました。Oracle Solaris OS、Oracle Solaris Cluster ソフトウェア、およびボリューム管理ソフトウェアは、Availability Suite ソフトウェアとパッチをインストールする前にクラスタノードにインストールする必要があります。

表 A-2 必要なハードウェアとソフトウェア

ハードウェアまたはソフトウェア	要件
ノードハードウェア	Availability Suite ソフトウェアは、Oracle Solaris OS を使用するすべてのサーバー上でサポートされます。  使用するハードウェアについては、 <a href="#">『Oracle Solaris Cluster 3.3 3/13 Hardware Administration Manual』</a> を参照してください。
ディスク容量	約 15M バイト

表 A-2 必要なハードウェアとソフトウェア (続き)

ハードウェアまたはソフトウェア	要件
Oracle Solaris OS	Oracle Solaris Cluster ソフトウェアがサポートする Oracle Solaris OS のリリース。  すべてのノードが同じバージョンの Oracle Solaris OS を使用する必要があります。  インストールについては、『Oracle Solaris Cluster ソフトウェアのインストール』を参照してください。
Oracle Solaris Cluster ソフトウェア	Oracle Solaris Cluster 3.3 ソフトウェア  インストールについては、『Oracle Solaris Cluster ソフトウェアのインストール』を参照してください。
ボリューム管理ソフトウェア	Solaris Volume Manager ソフトウェア。  すべてのノードで、同じバージョンのボリューム管理ソフトウェアを使用する。  インストールについては、『Oracle Solaris Cluster ソフトウェアのインストール』の第4章「Solaris Volume Manager ソフトウェアの構成」を参照してください。
Availability Suite ソフトウェア	ソフトウェアのインストール方法については、使用しているリリースの Availability Suite ソフトウェアのインストールマニュアルを参照してください。 <ul style="list-style-type: none"><li>■ Sun StorageTek Availability Suite 4.0 – Sun StorageTek Availability のドキュメント</li></ul>
Availability Suite のソフトウェアパッチ	最新のパッチについては、 <a href="#">My Oracle Support</a> にログインしてください。

## デバイスグループとリソースグループの構成例

このセクションでは、NFS アプリケーション向けにディスクデバイスグループとリソースグループをどのように構成するかを説明します。追加情報については、[342 ページの「複製リソースグループの構成」](#) および [343 ページの「アプリケーションリソースグループの構成」](#) を参照してください。

ここでは、次の手順について説明します。

- [352 ページの「プライマリクラスタでデバイスグループを構成する方法」](#)
- [353 ページの「セカンダリクラスタでデバイスグループを構成する方法」](#)
- [354 ページの「プライマリクラスタのファイルシステムを NFS アプリケーション向けに構成する方法」](#)
- [355 ページの「セカンダリクラスタのファイルシステムを NFS アプリケーション向けに構成する方法」](#)
- [356 ページの「プライマリクラスタで複製リソースグループを作成する方法」](#)

- 358 ページの「セカンダリクラスタで複製リソースグループを作成する方法」
- 360 ページの「プライマリクラスタで NFS アプリケーションリソースグループを作成する方法」
- 362 ページの「セカンダリクラスタで NFS アプリケーションリソースグループを作成する方法」
- 370 ページの「複製が正しく構成されていることを確認する方法」

構成例のために作成されたグループとリソースの名前を次の表に示します。

表 A-3 構成例内のグループとリソースのサマリー

グループまたはリソース	名前	説明
デバイスグループ	devgrp	デバイスグループ
複製リソースグループ とリソース	devgrp-stor-rg	複製リソースグループ
	lhost-reprg-prim、 lhost-reprg-sec	プライマリクラスタとセカンダリクラスタの複製リソースグループの論理ホスト名
	devgrp-stor	複製リソースグループの HAStoragePlus リソース
アプリケーションリ ソースグループとリ ソース	nfs-rg	アプリケーションリソースグループ
	lhost-nfsrg-prim、 lhost-nfsrg-sec	プライマリクラスタとセカンダリクラスタのアプリケーションリソースグループの論理ホスト名
	nfs-dg-rs	アプリケーションの HAStoragePlus リソース
	nfs-rs	NFS リソース

devgrp-stor-rg 以外のグループとリソースの名前は一例で、必要に応じて変更可能です。複製リソースグループは、*devicegroupname-stor-rg* というフォーマットでなければなりません。

この構成例では、SVM ソフトウェアを使用します。Solaris Volume Manager ソフトウェアの詳細については、『Oracle Solaris Cluster ソフトウェアのインストール』の第 4 章「Solaris Volume Manager ソフトウェアの構成」を参照してください。

次の図は、デバイスグループ内に作成されたボリュームを示しています。

図 A-8 デバイスグループのボリューム



---

注- この手順で定義されたボリュームに、シリンダ 0 などのディスクラベルのプライベート領域を含めてはいけません。この制約は SVM ソフトウェアが自動的に管理します。

---

## ▼ プライマリクラスタでデバイスグループを構成する方法

始める前に 次のタスクを完成していることを確認してください。

- 次のセクションのガイドラインと要件を確認します。
  - 338 ページの「クラスタにおける [Availability Suite ソフトウェアの理解](#)」
  - 341 ページの「クラスタ間でホストベースのデータ複製を構成するためのガイドライン」
- 348 ページの「クラスタの接続とインストール」で説明されているように、プライマリクラスタおよびセカンダリクラスタを設定します。



- 1 **nodeA** にスーパーユーザー、または RBAC の承認 `solaris.cluster.modify` を提供する役割になりますしてアクセスします。  
nodeA は、プライマリクラスタの最初のノードです。どのノードが nodeA であるかを確認するには、[図 A-7](#) を参照してください。
- 2 **nodeA** 上に、ボリューム 1 の **vol01** からボリューム 4 の **vol04** までを含むディスクグループを作成します。
- 3 ディスクグループを構成してデバイスグループを作成します。  
nodeA# `cldevicegroup create -t svm -n nodeA nodeB devgrp`  
デバイスグループの名前は devgrp です。
- 4 デバイスグループのファイルシステムを作成します。  
nodeA# `newfs /dev/vx/rdisk/devgrp/vol01 < /dev/null`  
nodeA# `newfs /dev/vx/rdisk/devgrp/vol02 < /dev/null`  
vol03 や vol04 にファイルシステムは必要ありません。これらはその代わり、raw ボリュームとして使用されます。

次の手順 [353 ページ](#) の「セカンダリクラスタでデバイスグループを構成する方法」に進みます。

## ▼ セカンダリクラスタでデバイスグループを構成する方法

始める前に 手順 [352 ページ](#) の「プライマリクラスタでデバイスグループを構成する方法」を完了します。

- 1 **nodeC** にスーパーユーザー、または RBAC の承認 `solaris.cluster.modify` を提供する役割になりますしてアクセスします。
- 2 **nodeC** 上に、ボリューム 1 の **vol01** からボリューム 4 の **vol04** までを含むディスクグループを作成します。
- 3 ディスクグループを構成してデバイスグループを作成します。  
nodeC# `cldevicegroup create -t svm -n nodeC devgrp`  
デバイスグループの名前は devgrp です。
- 4 デバイスグループのファイルシステムを作成します。  
nodeC# `newfs /dev/vx/rdisk/devgrp/vol01 < /dev/null`  
nodeC# `newfs /dev/vx/rdisk/devgrp/vol02 < /dev/null`  
vol03 や vol04 にファイルシステムは必要ありません。これらはその代わり、raw ボリュームとして使用されます。

次の手順 354 ページの「プライマリクラスタのファイルシステムを NFS アプリケーション向けに構成する方法」に進みます。

## ▼ プライマリクラスタのファイルシステムを NFS アプリケーション向けに構成する方法

始める前に 手順353 ページの「セカンダリクラスタでデバイスグループを構成する方法」を完了します。

- 1 **nodeA** および **nodeB** で、スーパーユーザー、または **RBAC** の承認 **solaris.cluster.admin** を提供する役割になります。
- 2 **nodeA** と **nodeB** で、NFS ファイルシステム向けのマウントポイントディレクトリを作成します。

例:

```
nodeA# mkdir /global/mountpoint
```

- 3 **nodeA** と **nodeB** で、マウントポイントに自動でマウントされるようにマスターボリュームを構成します。

**nodeA** と **nodeB** の `/etc/vfstab` ファイルに以下のテキストを追加するか、既存のテキストと置き換えます。テキストは 1 行で記述してください。

```
/dev/vx/dsk/devgrp/vol01 /dev/vx/rdisk/devgrp/vol01 \
/global/mountpoint ufs 3 no global,logging
```

デバイスグループで使用されているボリューム名とボリューム番号を確認するには、[図 A-8](#) を参照してください。

- 4 **nodeA** で、Oracle Solaris Cluster HA for NFS データサービスによって使用されるファイルシステム情報のボリュームを作成します。

```
nodeA# vxassist -g devgrp make vol05 120m disk1
```

ボリューム 5 の **vol05** は、Oracle Solaris Cluster HA for NFS データサービスによって使用されるファイルシステム情報を含みます。

- 5 **nodeA** で、デバイスグループと Oracle Solaris Cluster ソフトウェアを再同期します。

```
nodeA# cldevicegroup sync devgrp
```

- 6 **nodeA** で、**vol05** のファイルシステムを作成します。

```
nodeA# newfs /dev/vx/rdisk/devgrp/vol05
```

- 7 **nodeA** と **nodeB** で、**vol05** のマウントポイントを作成します。  
次の例では、マウントポイント `/global/etc` を作成しています。

```
nodeA# mkdir /global/etc
```

- 8 **nodeA** と **nodeB** で、マウントポイントに自動でマウントされるように **vol05** を構成します。

**nodeA** と **nodeB** の `/etc/vfstab` ファイルに以下のテキストを追加するか、既存のテキストと置き換えます。テキストは1行で記述してください。

```
/dev/vx/dsk/devgrp/vol05 /dev/vx/rdisk/devgrp/vol05 \
/global/etc ufs 3 yes global,logging
```

- 9 **nodeA** 上の **vol05** をマウントします。

```
nodeA# mount /global/etc
```

- 10 **vol05** をリモートシステムからアクセスできるようにします。

- a. **nodeA** に `/global/etc/SUNW.nfs` というディレクトリを作成します。

```
nodeA# mkdir -p /global/etc/SUNW.nfs
```

- b. **nodeA** に `/global/etc/SUNW.nfs/dfstab.nfs-rs` ファイルを作成します。

```
nodeA# touch /global/etc/SUNW.nfs/dfstab.nfs-rs
```

- c. **nodeA** の `/global/etc/SUNW.nfs/dfstab.nfs-rs` ファイルに次の行を追加します。

```
share -F nfs -o rw -d "HA NFS" /global/mountpoint
```

次の手順 355 ページの「セカンダリクラスタのファイルシステムを NFS アプリケーション向けに構成する方法」に進みます。

## ▼ セカンダリクラスタのファイルシステムを NFS アプリケーション向けに構成する方法

始める前に 手順354 ページの「プライマリクラスタのファイルシステムを NFS アプリケーション向けに構成する方法」を完了します。

- 1 **nodeC** で、スーパーユーザー、または RBAC の承認 `solaris.cluster.admin` を提供する役割になります。

- 2 **nodeC** で、NFS ファイルシステム向けのマウントポイントディレクトリを作成します。

例:

```
nodeC# mkdir /global/mountpoint
```

- 3 **nodeC** で、マウントポイントに自動でマウントされるようにマスターボリュームを構成します。

**nodeC** の `/etc/vfstab` ファイルに以下のテキストを追加するか、既存のテキストと置き換えます。テキストは1行で記述してください。

```
/dev/vx/dsk/devgrp/vol01 /dev/vx/rdisk/devgrp/vol01 \
/global/mountpoint ufs 3 no global,logging
```

- 4 **nodeC** で、**Oracle Solaris Cluster HA for NFS** データサービスによって使用されるファイルシステム情報のボリュームを作成します。

```
nodeC# vxassist -g devgrp make vol05 120m disk1
```

ボリューム 5 の **vol05** は、**Oracle Solaris Cluster HA for NFS** データサービスによって使用されるファイルシステム情報を含みます。

- 5 **nodeC** で、デバイスグループと **Oracle Solaris Cluster** ソフトウェアを再同期します。

```
nodeC# cldevicegroup sync devgrp
```

- 6 **nodeC** で、**vol05** のファイルシステムを作成します。

```
nodeC# newfs /dev/vx/rdisk/devgrp/vol05
```

- 7 **nodeC** で、**vol05** のマウントポイントを作成します。

次の例では、マウントポイント **/global/etc** を作成しています。

```
nodeC# mkdir /global/etc
```

- 8 **nodeC** で、マウントポイントに自動でマウントされるように **vol05** を構成します。

**nodeC** の **/etc/vfstab** ファイルに以下のテキストを追加するか、既存のテキストと置き換えます。テキストは 1 行で記述してください。

```
/dev/vx/dsk/devgrp/vol05 /dev/vx/rdisk/devgrp/vol05 \
/global/etc ufs 3 yes global,logging
```

- 9 **nodeC** 上の **vol05** をマウントします。

```
nodeC# mount /global/etc
```

- 10 **vol05** をリモートシステムからアクセスできるようにします。

- a. **nodeC** に、**/global/etc/SUNW.nfs** というディレクトリを作成します。

```
nodeC# mkdir -p /global/etc/SUNW.nfs
```

- b. **nodeC** に **/global/etc/SUNW.nfs/dfstab.nfs-rs** ファイルを作成します。

```
nodeC# touch /global/etc/SUNW.nfs/dfstab.nfs-rs
```

- c. **nodeC** の **/global/etc/SUNW.nfs/dfstab.nfs-rs** ファイルに次の行を追加します。

```
share -F nfs -o rw -d "HA NFS" /global/mountpoint
```

次の手順 [356 ページの「プライマリクラスタで複製リソースグループを作成する方法」](#)に進みます。

## ▼ プライマリクラスタで複製リソースグループを作成する方法

始める前に 手順 [355 ページの「セカンダリクラスタのファイルシステムを NFS アプリケーション向けに構成する方法」](#) を完了します。

- 1 **nodeA** にスーパーユーザー、または RBAC の承認  
**solaris.cluster.modify**、**solaris.cluster.admin**、および **solaris.cluster.read** を提供する役割になりますしてアクセスします。

- 2 **SUNW.HAStoragePlus** というリソースタイプを登録します。

```
nodeA# clresourcetype register SUNW.HAStoragePlus
```

- 3 デバイスグループの複製リソースグループを作成します。

```
nodeA# clresourcegroup create -n nodeA,nodeB devgrp-stor-rg
```

-n nodeA,nodeB      クラスタノード nodeA および nodeB が複製リソースグループをマスターできることを指定します。

devgrp-stor-rg      複製リソースグループの名前。この名前で、devgrp はデバイスグループの名前を指定します。

- 4 複製リソースグループに **SUNW.HAStoragePlus** リソースを追加します。

```
nodeA# clresource create -g devgrp-stor-rg -t SUNW.HAStoragePlus \
-p GlobalDevicePaths=devgrp \
-p AffinityOn=True \
devgrp-stor
```

-g                      リソースを追加するリソースグループを指定します。

-p GlobalDevicePaths=      Availability Suite ソフトウェアが依存する拡張プロパティを指定します。

-p AffinityOn=True      SUNW.HAStoragePlus リソースが、-x GlobalDevicePaths=で定義されたグローバルデバイスおよびクラスタファイルシステムに対して、アフィニティスウィッチオーバーを実行する必要があることを指定します。したがって、複製リソースグループがフェイルオーバーまたはスイッチオーバーすると、関連デバイスグループがスイッチオーバーします。

これらの拡張プロパティについての詳細は、[SUNW.HAStoragePlus\(5\)](#) のマニュアルページを参照してください。

- 5 論理ホスト名リソースを複製リソースグループに追加します。

```
nodeA# clreslogicalhostname create -g devgrp-stor-rg lhost-reprg-prim
```

プライマリクラスタ上の複製リソースグループの論理ホスト名は lhost-reprg-prim です。

- 6 リソースを有効にし、リソースグループを管理し、リソースグループをオンラインにします。

```
nodeA# clresourcegroup online -e -M -n nodeA devgrp-stor-rg
```

- e 関連付けられたリソースを有効にします。
- M リソースグループを管理状態にします。
- n リソースグループをオンラインにするノードを指定します。

## 7 リソースグループがオンラインであることを確認します。

```
nodeA# clresourcegroup status devgrp-stor-rg
```

リソースグループの状態フィールドを調べ、複製リソースグループが **nodeA** でオンラインとなっていることを確認します。

次の手順 [358 ページの「セカンダリクラスタで複製リソースグループを作成する方法」](#)に進みます。

## ▼ セカンダリクラスタで複製リソースグループを作成する方法

始める前に 手順 [356 ページの「プライマリクラスタで複製リソースグループを作成する方法」](#)を完了します。

### 1 **nodeC** にスーパーユーザー、または RBAC の承認 **solaris.cluster.modify**、**solaris.cluster.admin**、および **solaris.cluster.read** を提供する役割になりますしてアクセスします。

### 2 **SUNW.HAStoragePlus** というリソースタイプを登録します。

```
nodeC# clresourcetype register SUNW.HAStoragePlus
```

### 3 デバイスグループの複製リソースグループを作成します。

```
nodeC# clresourcegroup create -n nodeC devgrp-stor-rg
```

create リソースグループを作成します。

-n リソースグループのノードリストを指定します。

devgrp デバイスグループの名前。

devgrp-stor-rg 複製リソースグループの名前。

### 4 複製リソースグループに **SUNW.HAStoragePlus** リソースを追加します。

```
nodeC# clresource create \
-t SUNW.HAStoragePlus \
-p GlobalDevicePaths=devgrp \
-p AffinityOn=True \
devgrp-stor
```

create リソースを作成します。

-t リソースタイプを指定します。

- p GlobalDevicePaths=** Availability Suite ソフトウェアが依存する拡張プロパティを指定します。
- p AffinityOn=True** SUNW.HAStoragePlus リソースが、**-x GlobalDevicePaths=** で定義されたグローバルデバイスおよびクラスタファイルシステムに対して、アフィニティスイッチオーバーを実行する必要があることを指定します。したがって、複製リソースグループがフェイルオーバーまたはスイッチオーバーすると、関連デバイスグループがスイッチオーバーします。

**devgrp-stor** 複製リソースグループの HAStoragePlus リソース

これらの拡張プロパティについての詳細は、[SUNW.HAStoragePlus\(5\)](#) のマニュアルページを参照してください。

## 5 論理ホスト名リソースを複製リソースグループに追加します。

```
nodeC# clreslogicalhostname create -g devgrp-stor-rg lhost-reprg-sec
```

プライマリクラスタ上の複製リソースグループの論理ホスト名は `lhost-reprg-sec` です。

## 6 リソースを有効にし、リソースグループを管理し、リソースグループをオンラインにします。

```
nodeC# clresourcegroup online -e -M -n nodeC devgrp-stor-rg
```

**online** オンラインにします。

**-e** 関連付けられたリソースを有効にします。

**-M** リソースグループを管理状態にします。

**-n** リソースグループをオンラインにするノードを指定します。

## 7 リソースグループがオンラインであることを確認します。

```
nodeC# clresourcegroup status devgrp-stor-rg
```

リソースグループの状態フィールドを調べ、複製リソースグループが `nodeC` でオンラインとなっていることを確認します。

次の手順 [360 ページの「プライマリクラスタで NFS アプリケーションリソースグループを作成する方法」](#)に進みます。

## ▼ プライマリクラスタで **NFS** アプリケーションリソースグループを作成する方法

この手順では、アプリケーションリソースグループを NFS に対して作成する方法を説明します。この手順はこのアプリケーションに固有で、別の種類のアプリケーションには使用できません。

始める前に 手順358 ページの「セカンダリクラスタで複製リソースグループを作成する方法」を完了します。

- 1 **nodeA** にスーパーユーザー、または RBAC の承認 **solaris.cluster.modify**、**solaris.cluster.admin**、および **solaris.cluster.read** を提供する役割になりますしてアクセスします。

- 2 **SUNW.nfs** をリソースタイプとして登録します。

```
nodeA# clresourcetype register SUNW.nfs
```

- 3 **SUNW.HASStoragePlus** をリソースタイプとして登録していない場合は、登録します。

```
nodeA# clresourcetype register SUNW.HASStoragePlus
```

- 4 デバイスグループ **devgrp** のアプリケーションリソースグループを作成します。

```
nodeA# clresourcegroup create \
-p Pathprefix=/global/etc \
-p Auto_start_on_new_cluster=False \
-p RG_dependencies=devgrp-stor-rg \
nfs-rg
```

```
Pathprefix=/global/etc
```

グループのリソースが管理ファイルを書き込むディレクトリを指定します。

```
Auto_start_on_new_cluster=False
```

アプリケーションリソースグループが自動的に起動しないように指定します。

```
RG_dependencies=devgrp-stor-rg
```

アプリケーションリソースグループが依存するリソースグループを指定します。この例では、アプリケーションリソースグループは複製リソースグループ **devgrp-stor-rg** に依存しています。

アプリケーションリソースグループが新しいプライマリノードにスイッチオーバーすると、複製リソースグループが自動的にスイッチオーバーします。ただし、複製リソースグループが新しいプライマリノードにスイッチオーバーする場合、アプリケーションリソースグループは手動でスイッチオーバーする必要があります。

```
nfs-rg
```

アプリケーションリソースグループの名前。



## 5 アプリケーションリソースグループに **SUNW.HAStoragePlus** リソースを追加します。

```
nodeA# clresource create -g nfs-rg \
-t SUNW.HAStoragePlus \
-p FileSystemMountPoints=/global/mountpoint \
-p AffinityOn=True \
nfs-dg-rs
```

create

リソースを作成します。

-g

リソースを追加するリソースグループを指定します。

-t SUNW.HAStoragePlus

リソースのタイプに **SUNW.HAStoragePlus** を指定します。

-p FileSystemMountPoints=/global/

ファイルシステムのマウントポイントがグローバルであることを指定します。

-p AffinityOn=True

アプリケーションリソースが -p GlobalDevicePaths= で定義されたグローバルデバイスとクラスタファイルシステム向けにアフィニティスイッチオーバーを実行する必要があることを指定します。したがって、アプリケーションリソースグループがフェイルオーバーまたはスイッチオーバーすると、関連デバイスグループがスイッチオーバーします。

nfs-dg-rs

NFS アプリケーション向けの **HAStoragePlus** リソースの名前。

これらの拡張プロパティについての詳細は、[SUNW.HAStoragePlus\(5\)](#) のマニュアルページを参照してください。

## 6 論理ホスト名リソースをアプリケーションリソースグループに追加します。

```
nodeA# clreslogicalhostname create -g nfs-rg \
lhost-nfsrg-prim
```

プライマリクラスタ上のアプリケーションリソースグループの論理ホスト名は **lhost-nfsrg-prim** です。

## 7 リソースを有効にし、アプリケーションリソースグループを管理し、アプリケーションリソースグループをオンラインにします。

### a. NFS アプリケーションの **HAStoragePlus** リソースを有効にします。

```
nodeA# clresource enable nfs-rs
```

### b. **nodeA** でアプリケーションリソースグループをオンラインにします。

```
nodeA# clresourcegroup online -e -M -n nodeA nfs-rg
```

online      リソースグループをオンラインにします。

-e          関連付けられたリソースを有効にします。

- M            リソースグループを管理状態にします。
- n            リソースグループをオンラインにするノードを指定します。
- nfs-rg       リソースグループの名前。

## 8 アプリケーションリソースグループがオンラインであることを確認します。

```
nodeA# clresourcegroup status
```

アプリケーションリソースグループの状態フィールドを調べ、複製リソースグループが nodeA と nodeB でオンラインとなっているかどうかを調べます。

次の手順 [362 ページの「セカンダリクラスタで NFS アプリケーションリソースグループを作成する方法」](#)に進みます。

## ▼ セカンダリクラスタで **NFS** アプリケーションリソースグループを作成する方法

始める前に 手順 [360 ページの「プライマリクラスタで NFS アプリケーションリソースグループを作成する方法」](#)を完了します。

- 1 nodeC にスーパーユーザー、または RBAC の承認 `solaris.cluster.modify`、`solaris.cluster.admin`、および `solaris.cluster.read` を提供する役割になりますしてアクセスします。

- 2 **SUNW.nfs** をリソースタイプとして登録します。

```
nodeC# clresourcetype register SUNW.nfs
```

- 3 **SUNW.HAStoragePlus** をリソースタイプとして登録していない場合は、登録します。

```
nodeC# clresourcetype register SUNW.HAStoragePlus
```

- 4 デバイスグループのアプリケーションリソースグループを作成します。

```
nodeC# clresourcegroup create \
-p Pathprefix=/global/etc \
-p Auto_start_on_new_cluster=False \
-p RG_dependencies=devgrp-stor-rg \
nfs-rg
```

```
create
```

リソースグループを作成します。

```
-p
```

リソースグループのプロパティを指定します。

```
Pathprefix=/global/etc
```

グループのリソースが管理ファイルを書き込むディレクトリを指定します。

`Auto_start_on_new_cluster=False`

アプリケーションリソースグループが自動的に起動しないように指定します。

`RG_dependencies=devgrp-stor-rg`

アプリケーションリソースグループが依存するリソースグループを指定します。この例では、アプリケーションリソースグループは複製リソースグループに依存しています。

アプリケーションリソースグループが新しいプライマリノードにスイッチオーバーすると、複製リソースグループが自動的にスイッチオーバーします。ただし、複製リソースグループが新しいプライマリノードにスイッチオーバーする場合、アプリケーションリソースグループは手動でスイッチオーバーする必要があります。

`nfs-rg`

アプリケーションリソースグループの名前。

## 5 アプリケーションリソースグループに **SUNW.HAStoragePlus** リソースを追加します。

```
nodeC# clresource create -g nfs-rg \
-t SUNW.HAStoragePlus \
-p FileSystemMountPoints=/global/mountpoint \
-p AffinityOn=True \
nfs-dg-rs
```

`create`

リソースを作成します。

`-g`

リソースを追加するリソースグループを指定します。

`-t SUNW.HAStoragePlus`

リソースのタイプに **SUNW.HAStoragePlus** を指定します。

`-p`

リソースのプロパティを指定します。

`FileSystemMountPoints=/global/`

ファイルシステムのマウントポイントがグローバルであることを指定します。

`AffinityOn=True`

アプリケーションリソースが `-x GlobalDevicePaths=` で定義されたグローバルデバイスとクラスタファイルシステム向けにアフィニティスイッチオーバーを実行する必要があることを指定します。したがって、アプリケーションリソースグループがフェイルオーバーまたはスイッチオーバーすると、関連デバイスグループがスイッチオーバーします。

`nfs-dg-rs`

NFS アプリケーション向けの **HAStoragePlus** リソースの名前。

これらの拡張プロパティについての詳細は、[SUNW.HAStoragePlus\(5\)](#) のマニュアルページを参照してください。

- 6 論理ホスト名リソースをアプリケーションリソースグループに追加します。

```
nodeC# clreslogicalhostname create -g nfs-rg \
lhost-nfsrg-sec
```

セカンダリクラスタ上のアプリケーションリソースグループの論理ホスト名は `lhost-nfsrg-sec` です。

- 7 NFS リソースをアプリケーションリソースグループに追加します。

```
nodeC# clresource create -g nfs-rg \
-t SUNW.nfs -p Resource_dependencies=nfs-dg-rs nfs-rg
```

- 8 アプリケーションリソースグループが `nodeC` でオンラインになっていないことを確認します。

```
nodeC# clresource disable -n nodeC nfs-rs
nodeC# clresource disable -n nodeC nfs-dg-rs
nodeC# clresource disable -n nodeC lhost-nfsrg-sec
nodeC# clresourcegroup online -n "" nfs-rg
```

`Auto_start_on_new_cluster=False` のため、リソースグループはリブート後もオフラインのままになります。

- 9 グローバルボリュームがプライマリクラスタにマウントされている場合は、セカンダリクラスタのグローバルボリュームのマウントを解除します。

```
nodeC# umount /global/mountpoint
```

ボリュームがセカンダリクラスタにマウントされていると、同期が失敗します。

次の手順 [364 ページの「データ複製の有効化例」](#)に進みます。

## データ複製の有効化例

このセクションでは、構成例のデータ複製をどのように有効にするかを説明します。このセクションでは、Availability Suite ソフトウェアコマンドの `sndradm` と `iiadm` を使用します。これらのコマンドの詳細は、Availability Suite のドキュメントを参照してください。

ここでは、次の手順について説明します。

- [364 ページの「プライマリクラスタで複製を有効にする方法」](#)
- [366 ページの「セカンダリクラスタで複製を有効にする方法」](#)

### ▼ プライマリクラスタで複製を有効にする方法

- 1 `nodeA` にスーパーユーザー、または RBAC の承認 `solaris.cluster.read` を提供する役割になりますしてアクセスします。

- 2 すべてのトランザクションをフラッシュします。

```
nodeA# lockfs -a -f
```

- 3 論理ホスト名 **lhost-reprg-prim** と **lhost-reprg-sec** がオンラインであることを確認します。

```
nodeA# clresourcegroup status
nodeC# clresourcegroup status
```

リソースグループの状態フィールドを調べます。

- 4 プライマリクラスタからセカンダリクラスタへのリモートミラー複製を有効にします。

この手順によって、プライマリクラスタのマスターボリュームからセカンダリクラスタのマスターボリュームへの複製が有効になります。さらに、この手順で **vol04** のリモートミラービットマップへの複製も有効になります。

- プライマリクラスタとセカンダリクラスタが同期されていない場合、Sun StorageTek Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeA# /usr/sbin/sndradm -n -e lhost-reprg-prim \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 lhost-reprg-sec \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 ip sync
```

- プライマリクラスタとセカンダリクラスタが同期されている場合、Sun StorageTek Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeA# /usr/sbin/sndradm -n -E lhost-reprg-prim \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 lhost-reprg-sec \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 ip sync
```

- 5 自動同期機能を有効にします。

Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeA# /usr/sbin/sndradm -n -a on lhost-reprg-prim \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 lhost-reprg-sec \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 ip sync
```

この手順で自動同期が有効になります。自動同期のアクティブ状態が **on** に設定されている場合、システムがリブートされたり障害が発生すると、ボリュームセットは再度同期化されます。

- 6 クラスタがロギングモードであることを確認します。

Availability Suite ソフトウェアに対して次のコマンドを使用します。

```
nodeA# /usr/sbin/sndradm -P
```

次のような出力が表示されます。

```
/dev/vx/rdsk/devgrp/vol01 ->
lhost-reprg-sec:/dev/vx/rdsk/devgrp/vol01
autosync: off, max q writes:4194304, max q fbas:16384, mode:sync,ctag:
devgrp, state: logging
```

ロギングモードでは、状態は `logging` で、自動同期のアクティブ状態は `off` です。ディスクのデータボリュームに書き込みが行われると、同じディスクのビットマップファイルが更新されます。

## 7 ポイントインタイムスナップショットを有効にします。

Availability Suite ソフトウェアに対して次のコマンドを使用します。

```
nodeA# /usr/sbin/iidm -e ind \
/dev/vx/rdsk/devgrp/vol01 \
/dev/vx/rdsk/devgrp/vol02 \
/dev/vx/rdsk/devgrp/vol03
nodeA# /usr/sbin/iidm -w \
/dev/vx/rdsk/devgrp/vol02
```

この手順によって、プライマリクラスタのマスターボリュームが同じクラスタのシャドウボリュームにコピーされるようになります。マスターボリューム、シャドウボリューム、およびポイントインタイムビットマップボリュームは同じデバイスグループに存在する必要があります。この例では、マスターボリュームは `vol01`、シャドウボリュームは `vol02`、ポイントインタイムビットマップボリュームは `vol03` です。

## 8 ポイントインタイムスナップショットをリモートミラーセットに設定します。

Availability Suite ソフトウェアに対して次のコマンドを使用します。

```
nodeA# /usr/sbin/sndradm -I a \
/dev/vx/rdsk/devgrp/vol01 \
/dev/vx/rdsk/devgrp/vol02 \
/dev/vx/rdsk/devgrp/vol03
```

この手順によって、ポイントインタイムスナップショットがリモートミラーボリュームセットに関連付けられます。Availability Suite ソフトウェアは、リモートミラー複製の前にポイントインタイムスナップショットを必ず取ります。

次の手順 [366 ページの「セカンダリクラスタで複製を有効にする方法」](#)に進みます。

## ▼ セカンダリクラスタで複製を有効にする方法

始める前に [手順364 ページの「プライマリクラスタで複製を有効にする方法」](#)を完了します。

- 1 スーパーユーザーとして `nodeC` にアクセスします。
- 2 すべてのトランザクションをフラッシュします。

```
nodeC# lockfs -a -f
```

- 3 プライマリクラスタからセカンダリクラスタへのリモートミラー複製を有効にします。

Availability Suite ソフトウェアに対して次のコマンドを使用します。

```
nodeC# /usr/sbin/sndradm -n -e lhost-reprg-prim \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 lhost-reprg-sec \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 ip sync
```

プライマリクラスタがセカンダリクラスタの存在を認識し、同期を開始します。クラスタのステータスについては、Availability Suite のシステムログファイル `/var/adm` を参照してください。

- 4 それぞれのポイントインタイムスナップショットを有効にします。

Availability Suite ソフトウェアに対して次のコマンドを使用します。

```
nodeC# /usr/sbin/iidm -e ind \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol02 \
/dev/vx/rdisk/devgrp/vol03
nodeC# /usr/sbin/iidm -w \
/dev/vx/rdisk/devgrp/vol02
```

- 5 ポイントインタイムスナップショットをリモートミラーセットに設定します。

Availability Suite ソフトウェアに対して次のコマンドを使用します。

```
nodeC# /usr/sbin/sndradm -I a \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol02 \
/dev/vx/rdisk/devgrp/vol03
```

次の手順 [367 ページの「データ複製の実行例」](#)に進みます。

## データ複製の実行例

このセクションでは、構成例のデータ複製をどのように実行するかを説明します。このセクションでは、Availability Suite ソフトウェアコマンドの `sndradm` と `iidm` を使用します。これらのコマンドの詳細は、Availability Suite の [ドキュメント](#)を参照してください。

ここでは、次の手順について説明します。

- [368 ページの「リモートミラー複製を実行する方法」](#)
- [369 ページの「ポイントインタイムスナップショットを実行する方法」](#)
- [370 ページの「複製が正しく構成されていることを確認する方法」](#)

## ▼ リモートミラー複製を実行する方法

この手順では、プライマリディスクのマスターボリュームがセカンダリディスクのマスターボリュームに複製されます。マスターボリュームは `vol01` で、リモートミラービットマップボリュームは `vol04` です。

- 1 スーパーユーザーとして **nodeA** にアクセスします。

- 2 クラスタがロギングモードであることを確認します。

Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeA# /usr/sbin/sndradm -P
```

次のような出力が表示されます。

```
/dev/vx/rdsk/devgrp/vol01 ->
lhost-reprg-sec:/dev/vx/rdsk/devgrp/vol01
autosync: off, max q writes:4194304, max q fbas:16384, mode:sync,ctag:
devgrp, state: logging
```

ロギングモードでは、状態は `logging` で、自動同期のアクティブ状態は `off` です。ディスクのデータボリュームに書き込みが行われると、同じディスクのビットマップファイルが更新されます。

- 3 すべてのトランザクションをフラッシュします。

```
nodeA# lockfs -a -f
```

- 4 **nodeC** で [手順 1](#) から [手順 3](#) を繰り返します。

- 5 **nodeA** のマスターボリュームを **nodeC** のマスターボリュームにコピーします。

Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeA# /usr/sbin/sndradm -n -m lhost-reprg-prim \
/dev/vx/rdsk/devgrp/vol01 \
/dev/vx/rdsk/devgrp/vol04 lhost-reprg-sec \
/dev/vx/rdsk/devgrp/vol01 \
/dev/vx/rdsk/devgrp/vol04 ip sync
```

- 6 複製が完了し、ボリュームが同期化されるのを待ちます。

Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeA# /usr/sbin/sndradm -n -w lhost-reprg-prim \
/dev/vx/rdsk/devgrp/vol01 \
/dev/vx/rdsk/devgrp/vol04 lhost-reprg-sec \
/dev/vx/rdsk/devgrp/vol01 \
/dev/vx/rdsk/devgrp/vol04 ip sync
```

- 7 クラスタが複製モードであることを確認します。

Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeA# /usr/sbin/sndradm -P
```



次のような出力が表示されます。

```
/dev/vx/rdisk/devgrp/vol01 ->
lhost-reprg-sec:/dev/vx/rdisk/devgrp/vol01
autosync: on, max q writes:4194304, max q fbas:16384, mode:sync,ctag:
devgrp, state: replicating
```

複製モードでは、状態は `replicating` で、自動同期のアクティブ状態は `on` です。プライマリボリュームに書き込みが行われると、Availability Suite ソフトウェアがセカンダリボリュームを更新します。

次の手順 [369 ページの「ポイントインタイムスナップショットを実行する方法」](#)に進みます。

## ▼ ポイントインタイムスナップショットを実行する方法

この手順では、ポイントインタイムスナップショットを使用して、プライマリクラスタのシャドウボリュームをプライマリクラスタのマスターボリュームに同期させます。マスターボリュームは `vol01`、ビットマップボリュームは `vol04`、シャドウボリュームは `vol02` です。

始める前に 手順[368 ページの「リモートミラー複製を実行する方法」](#)を完了します。

- 1 **nodeA** にスーパーユーザー、または RBAC の承認 `solaris.cluster.modify` および `solaris.cluster.admin` を提供する役割になりますしてアクセスします。

- 2 **nodeA** で実行されているリソースを無効にします。

```
nodeA# clresource disable -n nodeA nfs-rs
```

- 3 プライマリクラスタをロギングモードに変更します。

Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeA# /usr/sbin/sndradm -n -l lhost-reprg-prim \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 lhost-reprg-sec \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 ip sync
```

ディスクのデータボリュームに書き込みが行われると、同じディスクのビットマップファイルが更新されます。複製は行われません。

- 4 プライマリクラスタのシャドウボリュームをプライマリクラスタのマスターボリュームに同期化させます。

Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeA# /usr/sbin/iiadm -u s /dev/vx/rdisk/devgrp/vol02
nodeA# /usr/sbin/iiadm -w /dev/vx/rdisk/devgrp/vol02
```

- 5 セカンダリクラスタのシャドウボリュームをセカンダリクラスタのマスターボリュームに同期化させます。

Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeC# /usr/sbin/iidm -u s /dev/vx/rdisk/devgrp/vol02
nodeC# /usr/sbin/iidm -w /dev/vx/rdisk/devgrp/vol02
```

- 6 **nodeA** でアプリケーションを再起動します。

```
nodeA# clresource enable -n nodeA nfs-rs
```

- 7 セカンダリボリュームをプライマリボリュームと再同期化させます。

Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeA# /usr/sbin/sndradm -n -u lhost-reprg-prim \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 lhost-reprg-sec \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 ip sync
```

次の手順 [370 ページの「複製が正しく構成されていることを確認する方法」](#)に進みます。

## ▼ 複製が正しく構成されていることを確認する方法

始める前に 手順[369 ページの「ポイントインタイムスナップショットを実行する方法」](#)を完了します。

- 1 **nodeA** および **nodeC** にスーパーユーザー、または RBAC の承認 **solaris.cluster.admin** を提供する役割になりますを使用してアクセスします。
- 2 プライマリクラスタが複製モードで、自動同期機能がオンになっていることを確認します。

Availability Suite ソフトウェアに対して次のコマンドを使用します。

```
nodeA# /usr/sbin/sndradm -P
```

次のような出力が表示されます。

```
/dev/vx/rdisk/devgrp/vol01 ->
lhost-reprg-sec:/dev/vx/rdisk/devgrp/vol01
autosync: on, max q writes:4194304, max q fbas:16384, mode:sync,ctag:
devgrp, state: replicating
```

複製モードでは、状態は **replicating** で、自動同期のアクティブ状態は **on** です。プライマリボリュームに書き込みが行われると、Availability Suite ソフトウェアがセカンダリボリュームを更新します。

- 3 プライマリクラスタが複製モードでない場合は、複製モードにします。

Availability Suite ソフトウェアに対して次のコマンドを使用します。

```
nodeA# /usr/sbin/sndradm -n -u lhost-reprg-prim \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 lhost-reprg-sec \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 ip sync
```

- 4 クライアントマシンにディレクトリを作成します。

- a. クライアントマシンにスーパーユーザーとしてログインします。

次のようなプロンプトが表示されます。

```
client-machine#
```

- b. クライアントマシンにディレクトリを作成します。

```
client-machine# mkdir /dir
```

- 5 ディレクトリをプライマリクラスタのアプリケーションにマウントし、マウントしたディレクトリを表示します。

- a. ディレクトリをプライマリクラスタのアプリケーションにマウントします。

```
client-machine# mount -o rw lhost-nfsrg-prim:/global/mountpoint /dir
```

- b. マウントしたディレクトリを表示します。

```
client-machine# ls /dir
```

- 6 ディレクトリをセカンダリクラスタのアプリケーションにマウントし、マウントしたディレクトリを表示します。

- a. ディレクトリをプライマリクラスタのアプリケーションからアンマウントします。

```
client-machine# umount /dir
```

- b. プライマリクラスタのアプリケーションリソースグループをオフラインにします。

```
nodeA# clresource disable -n nodeA nfs-rs
nodeA# clresource disable -n nodeA nfs-dg-rs
nodeA# clresource disable -n nodeA lhost-nfsrg-prim
nodeA# clresourcegroup online -n "" nfs-rg
```

- c. プライマリクラスタをロギングモードに変更します。

Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeA# /usr/sbin/sndradm -n -l lhost-reprg-prim \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 lhost-reprg-sec \
```

```
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 ip sync
```

ディスクのデータボリュームに書き込みが行われると、同じディスクのビットマップファイルが更新されます。複製は行われません。

- d. **PathPrefix** ディレクトリが使用可能であることを確認します。

```
nodeC# mount | grep /global/etc
```

- e. セカンダリクラスタのアプリケーションリソースグループをオンラインにします。

```
nodeC# clresourcegroup online -eM -n nodeC nfs-rg
```

- f. クライアントマシンにスーパーユーザーとしてアクセスします。

次のようなプロンプトが表示されます。

```
client-machine#
```

- g. [手順 4](#) で作成したディレクトリをセカンダリクラスタのアプリケーションにマウントします。

```
client-machine# mount -o rw lhost-nfsrg-sec:/global/mountpoint /dir
```

- h. マウントしたディレクトリを表示します。

```
client-machine# ls /dir
```

- 7 [手順 5](#) で表示されたディレクトリが [手順 6](#) で表示されたディレクトリと同じであることを確認します。

- 8 プライマリクラスタのアプリケーションを、マウントされたディレクトリに戻します。

- a. セカンダリクラスタのアプリケーションリソースグループをオフラインにします。

```
nodeC# clresource disable -n nodeC nfs-rs
nodeC# clresource disable -n nodeC nfs-dg-rs
nodeC# clresource disable -n nodeC lhost-nfsrg-sec
nodeC# clresourcegroup online -n "" nfs-rg
```

- b. グローバルボリュームは必ずセカンダリクラスタからアンマウントします。

```
nodeC# umount /global/mountpoint
```

- c. プライマリクラスタのアプリケーションリソースグループをオンラインにします。

```
nodeA# clresourcegroup online -eM -n nodeA nfs-rg
```

## d. プライマリクラスタを複製モードに変更します。

Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeA# /usr/sbin/sndradm -n -u lhost-reprg-prim \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 lhost-reprg-sec \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 ip sync
```

プライマリボリュームに書き込みが行われると、Availability Suite ソフトウェアがセカンダリボリュームを更新します。

参照 [373 ページの「テイクオーバーの管理の例」](#)

## テイクオーバーの管理の例

このセクションでは、フェイルオーバーを発生させる方法と、アプリケーションがどのようにセカンダリクラスタに転送されるかを説明します。フェイルオーバー後、DNS エントリを更新します。詳細は、[346 ページの「テイクオーバーの管理のガイドライン」](#)を参照してください。

ここでは、次の手順について説明します。

- [373 ページの「スイッチオーバーを発生させる方法」](#)
- [374 ページの「DNS エントリを更新する方法」](#)

### ▼ スwitchオーバーを発生させる方法

- 1 **nodeA** および **nodeC** にスーパーユーザー、または RBAC の承認 **solaris.cluster.admin** を提供する役割になりますを使用してアクセスします。
- 2 プライマリクラスタをロギングモードに変更します。

Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeA# /usr/sbin/sndradm -n -l lhost-reprg-prim \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 lhost-reprg-sec \
/dev/vx/rdisk/devgrp/vol01 \
/dev/vx/rdisk/devgrp/vol04 ip sync
```

ディスクのデータボリュームに書き込みが行われると、同じデバイスグループのビットマップボリュームが更新されます。複製は行われません。

- 3 プライマリクラスタとセカンダリクラスタが、自動同期がオフでロギングモードになっていることを確認します。

- a. **nodeA** で、モードと設定を確認します。

Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeA# /usr/sbin/sndradm -P
```

次のような出力が表示されます。

```
/dev/vx/rdsk/devgrp/vol01 ->
lhost-reprg-sec:/dev/vx/rdsk/devgrp/vol01
autosync:off, max q writes:4194304,max q fbas:16384,mode:sync,ctag:
devgrp, state: logging
```

- b. **nodeC** で、モードと設定を確認します。

Availability Suite ソフトウェアに対して次のコマンドを実行します。

```
nodeC# /usr/sbin/sndradm -P
```

次のような出力が表示されます。

```
/dev/vx/rdsk/devgrp/vol01 <-
lhost-reprg-prim:/dev/vx/rdsk/devgrp/vol01
autosync:off, max q writes:4194304,max q fbas:16384,mode:sync,ctag:
devgrp, state: logging
```

**nodeA** と **nodeC** については、状態が **logging** で、自動同期のアクティブ状態が **off** である必要があります。

- 4 セカンダリクラスタが、プライマリクラスタからテイクオーバーする準備ができていることを確認します。

```
nodeC# fsck -y /dev/vx/rdsk/devgrp/vol01
```

- 5 セカンダリクラスタにスイッチオーバーします。

```
nodeC# clresourcegroup switch -n nodeC nfs-rg
```

次の手順 [374 ページの「DNS エントリを更新する方法」](#)に進みます。

## ▼ DNS エントリを更新する方法

DNS がクライアントをクラスタにどのようにマッピングするかについては、[図 A-6](#)を参照してください。

始める前に 手順[373 ページの「スイッチオーバーを発生させる方法」](#)を完了します。

- 1 **nsupdate** コマンドを開始します。

詳細は、[nsupdate\(1M\)](#) のマニュアルページを参照してください。

- 2 両方のクラスタについて、アプリケーションリソースグループの論理ホスト名とクラスタ IP アドレス間の現在の DNS マッピングを削除します。

```
> update delete lhost-nfsrg-prim A
> update delete lhost-nfsrg-sec A
> update delete ipaddress1rev.in-addr.arpa ttl PTR lhost-nfsrg-prim
> update delete ipaddress2rev.in-addr.arpa ttl PTR lhost-nfsrg-sec
```

*ipaddress1rev*      プライマリクラスタの IP アドレス (逆順) です。

*ipaddress2rev*      セカンダリクラスタの IP アドレス (逆順) です。

*ttl*                  秒単位の有効時間です。一般的な値は 3600 になります。

- 3 両方のクラスタについて、アプリケーションリソースグループの論理ホスト名とクラスタ IP アドレス間の、新しい DNS マッピングを作成します。

プライマリ論理ホスト名をセカンダリクラスタの IP アドレスにマッピングし、セカンダリ論理ホスト名をプライマリクラスタの IP アドレスにマッピングします。

```
> update add lhost-nfsrg-prim ttl A ipaddress2fwd
> update add lhost-nfsrg-sec ttl A ipaddress1fwd
> update add ipaddress2rev.in-addr.arpa ttl PTR lhost-nfsrg-prim
> update add ipaddress1rev.in-addr.arpa ttl PTR lhost-nfsrg-sec
```

*ipaddress2fwd*      セカンダリクラスタの IP アドレス (正順) です。

*ipaddress1fwd*      プライマリクラスタの IP アドレス (正順) です。





# 索引

---

## A

autoboot プロパティ, 223  
Availability Suite, データ複製に使用, 337

## B

boot コマンド, 65–67

## C

cconsole コマンド, 20, 23  
ccp コマンド, 19, 23  
claccess コマンド, 19  
cldevicegroup コマンド, 19  
cldevice コマンド, 19  
clinterconnect コマンド, 19  
clnasdevice コマンド, 19  
clnode check コマンド, 19  
clnode コマンド, 265, 266–267  
clquorum コマンド, 19  
clreslogicalhostname コマンド, 19  
clresourcegroup コマンド, 19, 266–267  
clresourcetype コマンド, 19  
clresource コマンド, 19  
    リソースとリソースグループの削除, 270  
clressharedaddress コマンド, 19  
clsetup, ゾーンクラスタへのネットワークアドレスの追加, 269–270  
clsetup ユーティリティ, 18, 19, 25  
clsnmpghost コマンド, 19

clsnmpmib コマンド, 19  
clsnmpuser コマンド, 19  
cltelemattribute コマンド, 19  
cluster check  
    コマンド  
        への変更, 44  
cluster check コマンド, 19  
    vfstab ファイル確認, 156  
cluster shutdown コマンド, 61–71  
clzonecluster  
    boot, 65–67  
    説明, 25  
    停止, 61–71  
clzonecluster コマンド, 19  
CPU シェア  
    グローバルクラスタの投票ノード, 283  
    グローバルクラスタの非投票ノード, 285  
    グローバルクラスタの非投票ノード、専用プロセッサセット, 288  
    構成, 281  
    制御, 281  
crlogin コマンド, 23  
cssh コマンド, 23  
ctelnet コマンド, 23

## D

DID 情報, 手動更新, 164–165  
DID 情報の手動更新, 164–165  
DR, 「動的再構成」を参照

**E****EMC SRDF**

- DID デバイスの構成, 112–113
- 管理, 109–121
- キャンパスクラスタのプライマリルームの完全な失敗後に回復する, 119–121
- 構成の確認, 113–114
- 構成例, 114–121
- 制限, 92
- 適応型コピー, 91
- ドミノモード, 91
- 複製グループの構成, 109–111
- ベストプラクティス, 94
- 要件, 92
- /etc/inet/hosts ファイル, 排他的 IP ゾーン上の構成, 225
- /etc/nsswitch.conf ファイル, 非大域ゾーンの変更, 224
- /etc/vfstab ファイル, 49
  - 構成の確認, 156
  - マウントポイントの追加, 155

**F**

- failback プロパティ, 139
- fence\_level, 「複製中」を参照
- Oracle Solaris Cluster ソフトウェアのパッチ適用, 293–295

**G**

- GUI 管理ツール, 18, 329–336
  - Oracle Solaris Cluster Manager, 329

**H**

- Hitachi TrueCopy
  - DID デバイスの構成, 100–102
  - 管理, 98–109
  - 構成の確認, 102–103
  - 構成例, 103–109
  - 制限, 92

**Hitachi TrueCopy (続き)**

- データまたはステータスモード, 91
- 複製グループの構成, 98–100
- ベストプラクティス, 94
- 要件, 92
- Hitachi Universal Replicator, 91
  - 制限, 92
  - ベストプラクティス, 94
  - 要件, 92
- hosts ファイル, 排他的 IP ゾーン上の構成, 225

**I****IPMP**

- 管理, 215
- ステータス, 34
- 排他的 IP ゾーン上のグループ構成, 224
- IP アドレス, 排他的 IP ゾーンのネームサービスへの追加, 225

**K**

- /kernel/drv/md.conf ファイル, 129

**L**

- lofi ファイル, アンインストール, 257

**M**

- md.tab ファイル, 21
- MIB
  - SNMP イベントの有効化と無効化, 260, 261
  - SNMP イベントプロトコルの変更, 261

**N**

- netcon コマンド, 20

Network Appliance NAS, 定足数デバイスとしてサポート, 172  
 nsswitch.conf ファイル, 非大域ゾーンの変更, 224  
 ntp.conf.cluster ファイル, 245  
 numsecondaries プロパティ, 141

## O

OpenBoot PROM (OBP), 243  
 Oracle Solaris Cluster Manager, 18, 329  
   RBAC の役割、設定, 331  
   起動, 333  
   サーバーアドレスの変更, 332  
 Oracle Solaris OS  
   CPU 制御, 281  
   svcadm コマンド, 243  
   グローバルクラスタの管理タスク, 16  
   グローバルクラスタの定義, 15  
   ゾーンクラスタ定義, 15  
   ノードの再起動に関する特別な指示, 78–82  
   ノードのブートに関する特別な指示, 76–78  
   ホストベースの複製, 89  
 Oracle Solaris Cluster 定足数サーバー, 定足数デバイスとしてサポート, 172  
 Oracle Solaris ゾーン  
   autoboot プロパティ, 223  
   nsswitch.conf ファイルの変更, 224  
   共有 IP ゾーン, 223  
   排他的 IP ゾーン  
     hosts ファイルの構成, 225  
     IPMP グループの構成, 224  
 Oracle Solaris ソフトウェア, SMF, 223

## R

raw ディスクデバイス, 命名規則, 155  
 raw ディスクデバイスグループ, 追加, 132  
 RBAC, 53–60  
   Oracle Solaris Cluster Manager, 331  
   グローバルクラスタの投票ノードの場合, 54  
   権利プロファイル (説明), 54–55

RBAC (続き)  
   タスク  
     カスタム役割を追加, 58  
     使用, 53  
     設定, 53  
     役割の追加, 55  
     ユーザーの変更, 59  
     非投票ノードの場合, 54  
 Role-Based Access Control, 「RBAC」を参照

## S

SATA, 173  
 SATA ストレージ, 定足数デバイスとしてサポート, 172  
 Secure Shell, 24  
 showrev -p コマンド, 26, 27  
 SMF, オンラインサービスの確認, 223  
 SNMP  
   イベント MIB の有効化と無効化, 260, 261  
   プロトコルの変更, 261  
   ホストの無効化, 263  
   ホストの有効化, 262  
   ユーザーの削除, 265  
   ユーザーの追加, 264  
 SNMP イベント MIB の有効化と無効化, 260, 261  
 Solaris OS  
   「Oracle Solaris OS」も参照  
 Solaris Volume Manager, raw ディスクデバイス名, 155  
 SRDF  
   「EMC SRDF」を参照  
 ssh, 24  
 Sun NAS, 定足数デバイスとしてサポート, 172  
 Sun ZFS Storage Appliance, 定足数デバイスとして追加, 175  
 Sun NAS 定足数デバイス, 追加, 175  
 Sun ZFS ストレージアプライアンス, 定足数デバイスとしてサポート, 172

**T**

## TrueCopy

「Hitachi TrueCopy」を参照

**U**

Universal Replicator, Hitachi Universal Replicator, 91  
/usr/cluster/bin/clresource, リソースグループ  
の削除, 270

/usr/cluster/bin/cluster check コマンド, vfstab  
ファイル確認, 156

**V**

/var/adm/messages ファイル, 85

## vfstab ファイル

構成の確認, 156

マウントポイントの追加, 155

**Z**

## ZFS

デバイスグループ追加, 132

ファイルシステムの削除, 271-273

複製, 132

ルートファイルシステムの制限, 122

ZFS Storage Appliance, 「Sun ZFS Storage Appliance  
定足数デバイス」を参照

**あ**

アクセス権、グローバルデバイスの, 96

アダプタ、トランスポート, 205

アフィニティスイッチオーバー、データ複製の  
構成, 357

アフィニティフェイルオーバー、データ複製の  
拡張プロパティ, 342

アプリケーションリソースグループ

ガイドライン, 343

データ複製の構成, 360-362

## アンインストール

lofi デバイスファイル, 257

Oracle Solaris Cluster ソフトウェア, 255

**い**

移行, グローバル名前空間, 126

## 一覧表示

定足数構成, 190

デバイスグループ構成, 143

## イベント MIB

SNMP の有効化と無効化, 260, 261

SNMP プロトコルの変更, 261

**え**

## エラーメッセージ

/var/adm/messages ファイル, 85

ノードの削除, 233-234

**か**

## 開始

グローバルクラスタノード, 71-85

ゾーンクラスタノード, 71-85

ノード, 71-85

回復, ストレージベースのデータ複製を使用する  
クラスタ, 93

概要, 定足数, 169-192

## 確認

SMF, 223

vfstab 構成, 156

グローバルマウントポイント, 49, 159

データ複製構成, 370-373

監視, ディスクパス, 161-162

監視解除, ディスクパス, 162-163

完全に修復する /var/adm/messages ファイル, 85  
管理

EMC SRDF で複製されたデバイス, 109-121

Hitachi TrueCopy で複製されたデバイ  
ス, 98-109

IPMP, 199-218

## 管理 (続き)

- クラスタインターコネクトとパブリックネットワーク, 199-218
- クラスタファイルシステム, 121
- グラフィカルユーザーインタフェース (Graphical User Interface、GUI) ツールによるクラスタ, 329-336
- グローバルクラスタ, 16
- グローバルクラスタ設定, 235-280
- グローバルクラスタの非投票ノード, 16
- ストレージベースの複製されたデバイス, 97-121
- ゾーンクラスタ, 16, 267
- 定足数, 169-192
- 管理コンソール, 20
- 「管理役割を追加」ウィザード, 説明, 55

## き

## 起動

- Oracle Solaris Cluster Manager, 333
- グローバルクラスタ, 65-67
- グローバルクラスタノード, 71-85
- ゾーンクラスタ, 65-67
- ゾーンクラスタノード, 71-85
- ノード, 71-85
- キャンパスクラスタ
  - ストレージベースのデータ複製, 90-94
  - ストレージベースのデータ複製を使用した回復, 93
- 共通エージェントコンテナ
  - セキュリティ鍵の再生成, 332
  - ポート番号の変更, 331
- 共有 IP ゾーン, 「Oracle Solaris ゾーン」を参照
- 共有 SCSI ディスク, 定足数デバイスとしてサポ-ト, 172
- 共有ディスクパス
  - 監視, 160-168
  - 自動リブートの無効化, 168
  - 自動リブートの有効化, 167
- 切り換え, デバイスグループのプライマリノード, 145-146

## く

- クラスタ
  - 時間の設定, 240
  - スコープ, 26
  - 名前の変更, 236-237
  - ノード認証, 238
  - バックアップ, 21, 311-319
  - ファイルの復元, 319
  - リブートするパッチの適用, 300
- クラスタインターコネクト
  - 管理, 199-218
  - ステータスのチェック, 202
  - 動的再構成, 201
- クラスタコンソールへのセキュア接続, 24
- クラスタコントロールパネル (CCP), 20
- クラスタの時間の設定, 240
- クラスタファイルシステム, 95-168
  - 管理, 121
  - グローバルクラスタの投票ノード, 121
  - グローバルクラスタの非投票ノード, 121
  - 構成の確認, 156
  - 削除, 157-159
  - 追加, 154-157
  - マウントオプション, 155
- クラスタファイルシステムのマウントオプション, 要件, 155
- グローバル
  - デバイス, 95-168
    - アクセス権の設定, 96
    - 動的再構成, 96-97
  - 名前空間, 95-97, 124
  - マウントポイント、確認, 49, 159
- グローバルクラスタ
  - 管理, 235-280
  - 構成の検証, 44
  - 構成の表示, 35
  - コンポーネントのステータス, 31
  - 定義, 16
  - 停止, 61-86
  - ノードの削除, 227
  - ブート, 61-86
  - リブート, 67
- グローバルクラスタノード
  - 起動, 71-85

## グローバルクラスタノード (続き)

再起動, 78-82

シャットダウン, 71-85

## グローバルクラスタの投票ノード

CPU シェア, 283

クラスタファイルシステムの管理, 121

## グローバルクラスタの非投票ノード

CPU シェア, 285, 288

管理, 16

クラスタファイルシステムの管理, 121

シャットダウンとリブート, 71

パッチの適用, 299

プライベートホスト名、削除, 248

プライベートホスト名の追加, 246

プライベートホスト名の変更, 246

## グローバル名前空間、移行, 126

## グローバル名前空間の更新, 124

## グローバルファイルシステム、「クラスタファイルシステム」を参照

## け

ケーブル、トランスポート, 205

## 検索

グローバルクラスタ用ノード ID, 238

ゾーンクラスタ用ノード ID, 238

ファイルシステム名, 312

## 検証

グローバルクラスタ構成, 44

ゾーンクラスタ構成, 44

権利プロファイル, RBAC, 54-55

## こ

## 構成

Hitachi TrueCopy で複製されたデバイス, 98-100

データ複製, 337-375

構成済みリソースの表示, 30

構成例 (キャンパスのクラスタ化)

2 ルーム、ストレージベースのデータ複製, 90-94

公平配分スケジューラ, CPU シェアの構成, 282

## コマンド

boot, 65-67

cconsole, 20, 23

ccp, 19, 23

claccess, 19

cldevice, 19

cldevicegroup, 19

clinterconnect, 19

clnasdevice, 19

clnode check, 19

clquorum, 19

clreslogicalhostname, 19

clresource, 19

clresourcegroup, 19

clresourcetype, 19

clressharedaddress, 19

clsetup, 19

clsnmp host, 19

clsnmpmib, 19

clsnmpuser, 19

cltelemetryattribute, 19

cluster check, 19, 21, 44, 49

cluster shutdown, 61-71

clzonecluster, 19, 61-71

clzonecluster boot, 65-67

clzonecluster verify, 44

crlogin, 23

cssh, 23

ctelnet, 23

netcon, 20

メタセット, 95-97

コマンド行管理ツール, 18

コンソール

セキュア接続, 24

接続, 23

## さ

## 再起動

グローバルクラスタノード, 78-82

ゾーンクラスタノード, 78-82

最後の定足数デバイス、削除, 183

再生成、セキュリティー鍵, 332

## 削除

- SNMP ホスト, 263
- SNMP ユーザー, 265
- Solaris Volume Manager デバイスグループ, 134
- クラスタファイルシステム, 157-159
- グローバルクラスタ上の非投票
  - ノード, 230-231
- 最後の定足数デバイス, 183
- ストレージアレイ, 231
- すべてのデバイスグループからノードを, 134
- ゾーンクラスタから, 226
- 定足数デバイス, 171, 181
- トランスポートケーブル、アダプタ、およびス
  - イッチ, 205
- ノード, 225, 227
- リソースとリソースグループをゾーンクラスタ
  - から削除, 270
- サポートされている定足数デバイスタイプ, 172

## し

- シャットダウン
  - グローバルクラスタノード, 71-85
  - ゾーンクラスタノード, 71-85
  - ノード, 71-85
  - 非大域ゾーン, 71
- 修復, 定足数デバイス, 191
- 使用, 役割 (RBAC), 53

## す

- スイッチ、トランスポート, 205
- スイッチバック、データ複製で実行するためのガイ
  - ドライン, 347
- ステータス
  - グローバルクラスタのコンポーネント, 31
  - ゾーンクラスタのコンポーネント, 31
- ストレージアレイ, 削除, 231
- ストレージベースのデータ複製, 90-94
  - 回復, 93
  - 制限, 92
  - 定義, 88
  - 定足数デバイス, 93

## ストレージベースのデータ複製 (続き)

- ベストプラクティス, 94
- 要件, 92
- ストレージベースの複製されたデバイス、管
  - 理, 97-121
- スナップショット
  - 「ストレージベースの複製」を参照
  - ポイントインタイム, 339

## せ

- セカンダリノード
  - 希望数の設定, 141
  - デフォルト数, 139
- セキュリティー鍵、再生成, 332
- 設定、役割 (RBAC), 53
- 専用プロセッサセット、構成, 288

## そ

- ゾーンクラスタ
  - アプリケーション用に準備, 267
  - 管理, 235-280
  - 構成の検証, 44
  - 構成の表示, 35
  - コンポーネントのステータス, 31
  - サポートされている直接マウント, 271-273
  - ゾーンパスの移動, 267
  - 定義, 16
  - 停止, 61-86
  - ネットワークアドレスの追加, 269-270
  - ファイルシステムの削除, 267
  - ブート, 61-86
  - 複製, 267
  - リブート, 67
- ゾーンクラスタノード
  - IP アドレスおよび NIC の指定, 219-225
  - 起動, 71-85
  - 再起動, 78-82
  - シャットダウン, 71-85
- ゾーンパス、移動, 267
- 属性、「プロパティ」を参照

## た

耐障害性, 定義, 338

タイムアウト, 定足数デバイスのデフォルト値の  
変更, 192

## ち

チェック, クラスタインターコネクトのステータ  
ス, 202

直接接続共有ディスク定足数デバイス, 追加, 173

直接マウント, ファイルシステムをゾーンクラ  
スタにエクスポート, 271-273

## つ

## 追加

SNMP ホスト, 262

SNMP ユーザー, 264

Solaris Volume Manager デバイスグループ, 131

Sun NAS 定足数デバイス, 175

Sun ZFS Storage Appliance NAS 定足数デバイ  
ス, 175

ZFS デバイスグループ, 132

カスタム役割 (RBAC), 58

クラスタファイルシステム, 154-157

ゾーンクラスタへのネットワークアドレ  
ス, 269-270

直接接続共有ディスク定足数デバイス, 173

定足数サーバー定足数デバイス, 177

定足数デバイス, 173

デバイスグループ, 129, 132

トランスポートケーブル、アダプタ、およびス  
イッチ, 203

ノード, 219-225

ノードをグローバルクラスタへ, 220

ノードをゾーンクラスタへ, 220

役割 (RBAC), 55

## 停止 (続き)

グローバルクラスタノード, 71-85

ゾーンクラスタ, 61-86

ゾーンクラスタノード, 71-85

ノード, 71-85

## ディスクパス

監視, 95-168

障害のあるディスクパスを表示, 163-164

監視解除, 162-163

ディスクパスの、ステータスエラーの解  
決, 164-165

## 定足数

概要, 169-192

管理, 169-192

## 定足数サーバー定足数デバイス

インストールの要件, 177

削除のトラブルシューティング, 182

追加, 177

## 定足数デバイス

「定足数サーバー定足数デバイス」を参照

## adding

Sun ZFS Storage Appliance NAS 定足数デバイ  
ス, 175

交換, 184

構成の一覧表示, 190

最後の定足数デバイスの削除, 183

削除, 171, 181

修復, 191

ストレージベースのデータ複製, 93

追加, 173

Sun NAS 定足数デバイス, 175

直接接続共有ディスク定足数デバイス, 173

定足数サーバー定足数デバイス, 177

デバイスの動的再構成, 171

デフォルトのタイムアウトの変更, 192

ノードリストの変更, 185

保守状態、デバイス, 187

保守状態、デバイスを保守状態から戻す, 189

定足数デバイスタイプ、サポートされているタイ  
プのリスト, 172

定足数デバイスの交換, 184

データ複製, 87-94

DNS エントリの更新, 374-375

## て

## 停止

グローバルクラスタ, 61-86



## データ複製 (続き)

## ガイドライン

スイッチオーバーの管理, 346

テイクオーバーの管理, 346

リソースグループの構成, 341

概要, 338

## 構成

NFS アプリケーション用ファイルシステム, 354-355

NFS アプリケーションリソースグループ, 360-362

アフィニティスイッチオーバー, 342, 357

デバイスグループ, 351

構成の確認, 370-373

構成例, 347

ストレージベース, 88, 90-94

定義, 88-89

テイクオーバーの管理, 373-375

同期, 339

必要なハードウェアとソフトウェア, 349

非同期, 339

ポイントインタイムスナップショット, 339, 369-370

ホストベース, 88

有効化, 364-367

## リソースグループ

アプリケーション, 343

共有アドレス, 345

構成, 342

作成, 356-358

スケーラブルアプリケーション, 344-345

フェイルオーバーアプリ

ケーション, 343-344

命名規則, 342

リモートミラー, 338, 368-369

例, 367-373

## データ複製の拡張プロパティ

アプリケーションリソース, 361, 363

複製リソース, 357, 358

## データ複製のスイッチオーバー

アフィニティスイッチオーバー, 342

実行, 373-375

## データ複製のためのスケーラブルアプリ

ケーション, 344-345

## データ複製のフェイルオーバーアプリケーション

アフィニティスイッチオーバー, 342

## ガイドライン

テイクオーバーの管理, 346

リソースグループ, 343-344

管理, 373-375

データ複製用の共有アドレスリソースグループ, 345

## 適用

パッチ, 296

リブートしないパッチ, 304

リブートしないパッチのゾーンクラスタへの適用, 296

デバイス, グローバル, 95-168

## デバイスグループ

raw ディスク

追加, 132

SVM

追加, 129

管理の概要, 122

構成の表示, 143

削除と登録解除, 134

主所有者権, 139

追加, 131

データ複製の構成, 351

変更プロパティ, 139

保守状態, 146

デバイスグループの主所有者権, 139

デバイスグループのプライマリノードの切り替え, 145-146

電源管理, 235

## と

同期データ複製, 91, 339

動的再構成, 96-97

クラスタインターコネクト, 201

定足数デバイス, 171

パブリックネットワークインタフェース, 217

登録解除, Solaris Volume Manager デバイスグループ, 134

ドメインネームシステム (DNS)

更新のガイドライン, 346

データ複製での更新, 374-375

トランスポートアダプタ、追加、203,205  
トランスポートケーブル  
追加、203,205  
無効化、210  
有効化、208  
トランスポートケーブルの無効化、210  
トランスポートケーブルの有効化、208  
トランスポートスイッチ、追加、203,205

## な

## 名前空間

移行、126  
グローバル、95-97

## ね

ネームサービス、排他的 IP ゾーンの IP アドレス  
マッピングの追加、225  
ネットワークアドレス、ゾーンクラスタへの追  
加、269-270  
ネットワークファイルシステム (Network File  
System、NFS)、データ複製用アプリケーション  
ファイルシステムの構成、354-355

## の

## ノード

ID 検索、238  
起動、71-85  
グローバルクラスタからのノードの削除、227  
グローバルクラスタからの非投票ノードの削  
除、230-231  
グローバルクラスタでの名前変更、249  
削除  
エラーメッセージ、233-234  
シャットダウン、71-85  
セカンダリ、139  
接続、23  
ゾーンクラスタから削除、226  
追加、219-225  
デバイスグループから削除、134

## ノード (続き)

認証、238  
負荷制限の構成、266-267  
プライマリ、96-97,139  
保守状態にする、251  
リブートするパッチのグローバルクラスタへの  
適用、296  
ノード名の変更  
グローバルクラスタでの、249

## は

排他的 IP ゾーン、「Oracle Solaris ゾーン」を参照

## バックアップ

クラスタ、21,311-319  
ファイルシステム、312  
ミラーのオンライン、316  
ルートファイルシステム、313

## パッチ

クラスタおよびファームウェアへの適用、300  
グローバルクラスタの非投票ノード内、299  
ヒント、294  
リブートしないパッチのゾーンクラスタへの適  
用、296  
リブートしないパッチの適用、304  
リブートするパッチのグローバルクラスタへの  
適用、296

## パブリックネットワーク

管理、199-218  
動的再構成、217

## ひ

非クラスタモードでのブート、82  
ビットマップ  
ポイントインタイムスナップショット、339  
リモートミラー複製、338  
非同期データ複製、91,339  
表示  
グローバルクラスタの構成、35  
障害のあるディスクパス、163-164  
ゾーンクラスタの構成、35

## ふ

## ファイル

/etc/vfstab, 49  
md.conf, 129  
md.tab, 21  
ntp.conf.cluster, 245  
対話式に復元, 320

## ファイルシステム

NFS アプリケーション  
データ複製の構成, 354-355  
restoring root  
ボリュームから, 323  
ゾーンクラスタ内で削除, 267  
名前の検索, 312  
バックアップ, 312  
ルートの復元  
説明, 320  
メタデバイスから, 323

## ブート

グローバルクラスタ, 61-86  
ゾーンクラスタ, 61-86  
非クラスタモード, 82  
非大域ゾーン, 71

## 負荷制限

concentrate\_load プロパティ, 265  
preemption\_mode プロパティ, 265  
ノード上での構成, 265, 266-267

負荷制限の構成, ノード上, 266-267

## 復元

クラスタファイル, 319  
ファイルを対話式に, 320  
ルートファイルシステム, 320  
ボリュームから, 323  
メタデバイスから, 323

複製, 「データ複製」を参照

複製, ストレージベース, 90-94

## プライベートホスト名

グローバルクラスタの非投票ノード, 246  
グローバルクラスタの非投票ノードでの変更, 246  
グローバルクラスタの非投票ノードの削除, 248  
ゾーンへの割り当て, 223  
変更, 243

## プロパティ

failback, 139  
numsecondaries, 141  
preferenced, 139

プロファイル, RBAC 権限, 54-55

## へ

## ベストプラクティス

EMC SRDF, 94  
Hitachi TrueCopy, 94  
Hitachi Universal Replicator, 94  
ストレージベースのデータ複製, 94

## 変更

numsecondaries プロパティ, 141  
Oracle Solaris Cluster Manager のサーバーアドレス, 332  
SNMP イベント MIB プロトコル, 261  
クラスタ名, 236-237  
定足数デバイスのノードリスト, 185  
プライベートホスト名, 243  
プライマリノード, 145-146  
プロパティ, 139  
ポート番号, 共通エージェントコンテナの使用, 331  
ユーザー (RBAC), 59

## ほ

## ポイントインタイムスナップショット

実行, 369-370  
定義, 339

ポート番号, 共通エージェントコンテナを使用した変更, 331

保守, 定足数デバイス, 187

## 保守状態

定足数デバイス, 187  
定足数デバイスを保守状態から戻す, 189  
ノード, 251

## ホスト

SNMP の追加と削除, 262, 263

ホストベースのデータ複製

定義, 88

ホストベースのデータ複製 (続き)

例, 337-375

ボリューム, 「ストレージベースの複製」を参照

## ま

マウントポイント

/etc/vfstab ファイルの修正, 155

グローバル, 49

マルチユーザーサービス, 確認, 223

## み

ミラー, オンラインバックアップ, 316

## め

命名規則

raw ディスクデバイス, 155

複製リソースグループ, 342

メタセットコマンド, 95-97

## も

モニタリング, 共有ディスクバス, 167

## や

役割

カスタム役割を追加, 58

設定, 53

役割の追加, 55

## ゆ

ユーザー

SNMP の削除, 265

SNMP の追加, 264

プロパティの変更, 59

ユーザーアカウントツール, 説明, 59

## り

リソース

構成済みタイプの表示, 30

削除, 270

リソースグループ

データ複製

構成, 342

構成のガイドライン, 341

フェイルオーバーでの役割, 342

リブート

グローバルクラスタ, 67

ゾーンクラスタ, 67

リモートのミラー化, 「ストレージベースの複製」を参照

リモート複製, 「ストレージベースの複製」を参照

リモートミラー複製

実行, 368-369

定義, 338

リモートログイン, 23

リリース情報, 26, 27

## る

ループバックマウント, ファイルシステムを  
ゾーンクラスタにエクスポート, 271-273

## れ

例

インタラクティブな妥当性検査のリスト, 47

機能の妥当性検査の実行, 47-48

クラスタファイルシステムの作成, 157

## ろ

ローカルのミラー化, 「ストレージベースの複製」を参照

---

ログイン, リモート, 23  
論理ホスト名リソース, データ複製テイク  
オーバーでの役割, 342

